deep imaging

Lecture 4: Mathematical preliminaries for discrete functions

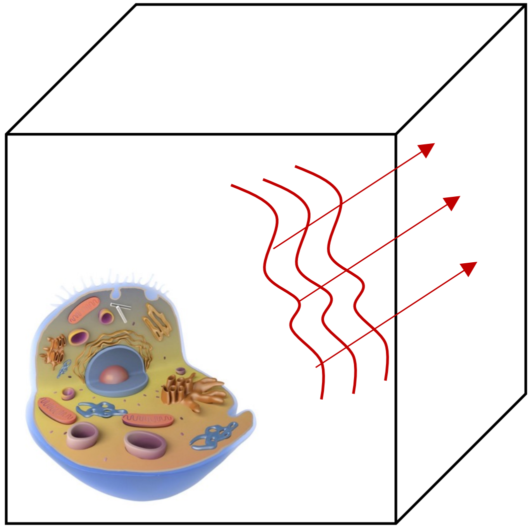
Machine Learning and Imaging

BME 548L

Roarke Horstmeyer

ML+Imaging pipeline introduction

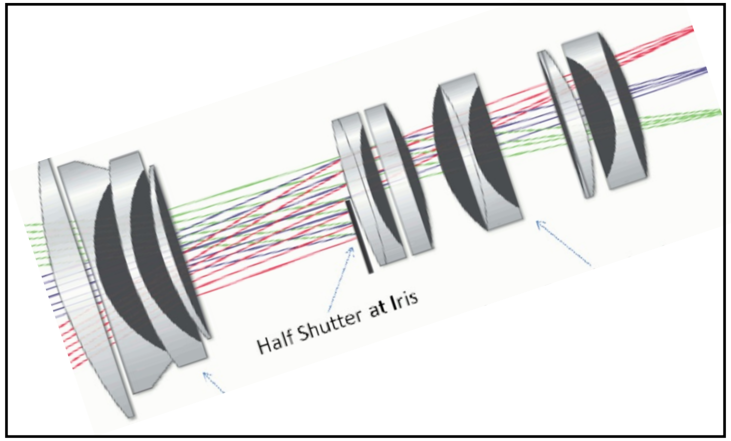
Real World



Continuous
complex fields

(last class)

Measurement device



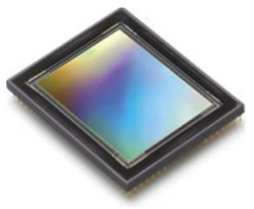
Black box transformations

- Convolution
- Fourier Transform

(last class)

Digitization

γ -> e^-

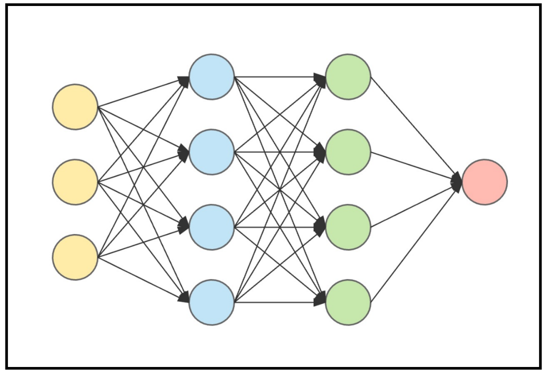


Sampling Theorem

Discrete math &
Linear algebra

(this class)

Machine Learning



Optimization

Linear classification

Logistic classifier

Neural networks

Convolutional NN's

(next few weeks)

Month 2

Month 3

Review - continuous Fourier transforms – for 2D images

Decomposition of a signal into elementary functions of form, $\exp\left(-2\pi i(f_x x + f_y y)\right)$:

$$\mathcal{F}\{U(x,y)\} = \hat{U}(f_x, f_y) = \iint_{-\infty}^{\infty} U(x,y)\exp\left(-2\pi i(f_x x + f_y y)\right) dx\, dy$$

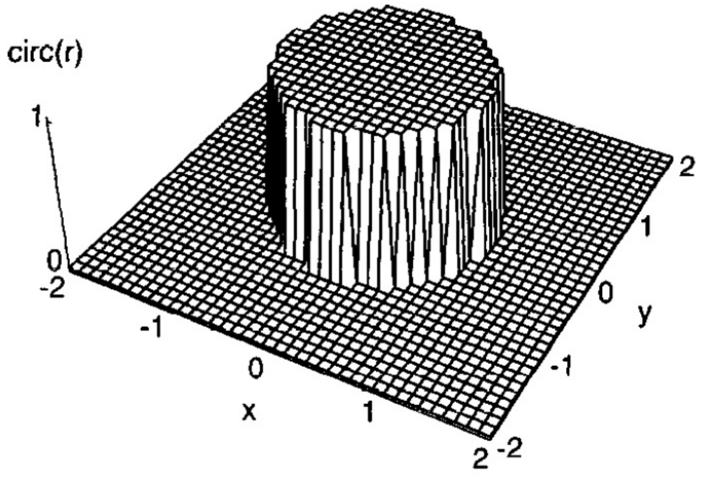
U is absolutely integrable & no infinite discontinuities. The inverse Fourier transform is,

$$\mathcal{F}^{-1}\{\hat{U}(f_x, f_y)\} = U(x,y) = \iint_{-\infty}^{\infty} \hat{U}(f_x, f_y)\exp\left(2\pi i(f_x x + f_y y)\right) df_x\, df_y$$

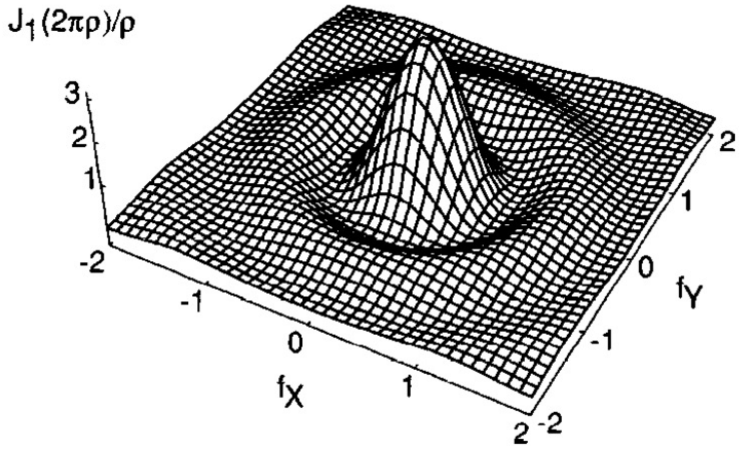
Additional Details:

- Goodman Chapter 2.1
- Mathworld/Wikipedia, Fourier Transform

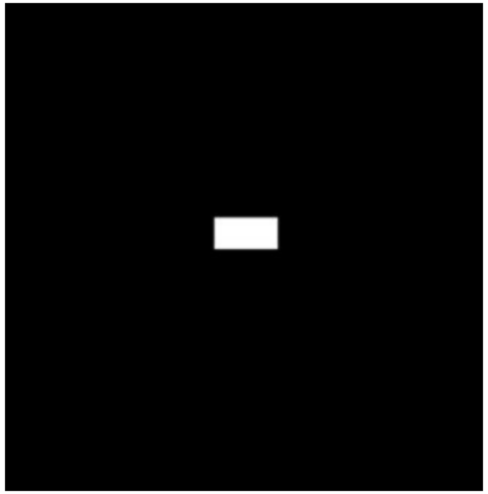
Examples of Fourier transform pairs, 2D

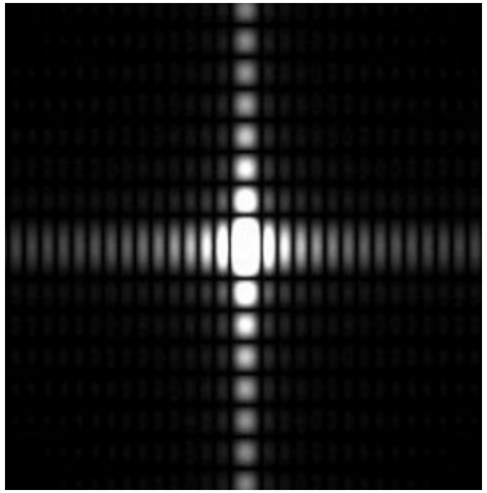


(a)



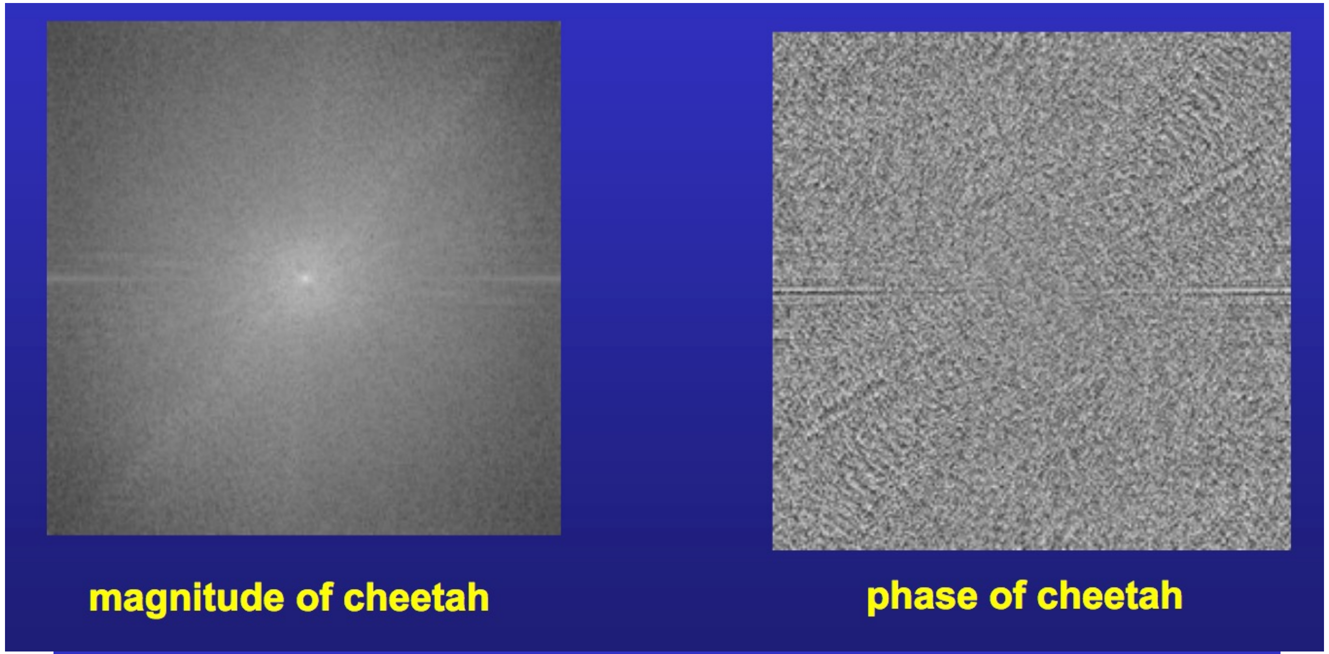
(b)



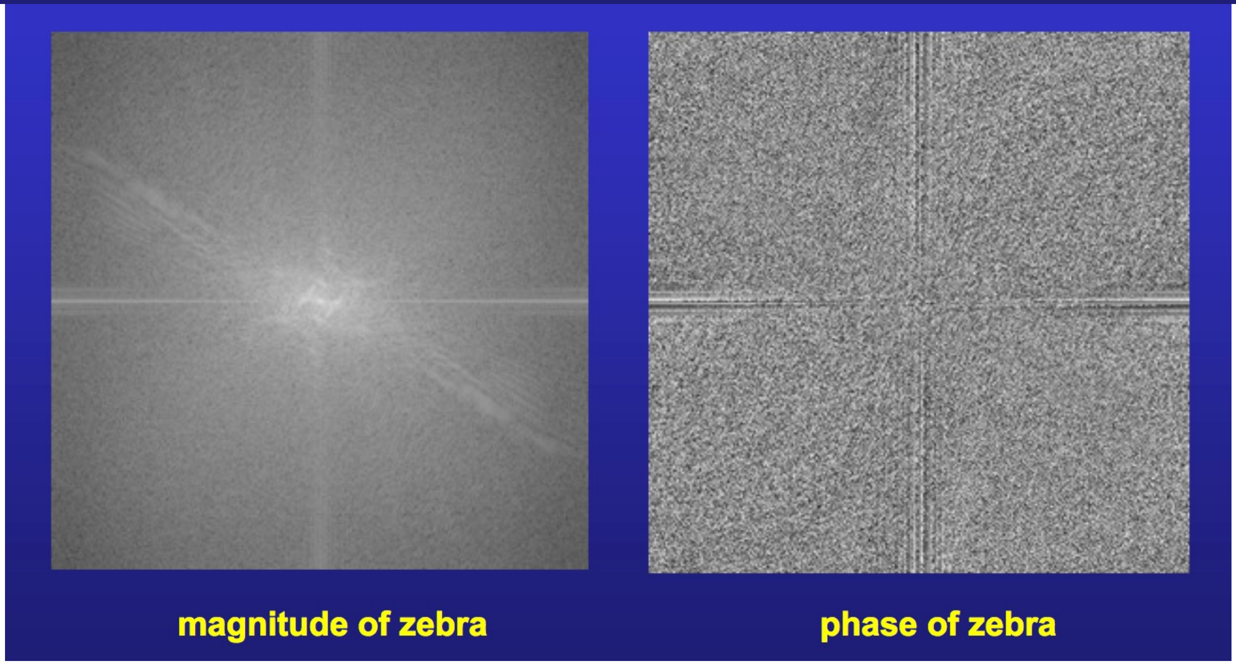


$U_1(x,y)$ $U_2(x,y)$

$\hat{U}_1(f_x, f_y)$



$\hat{U}_2(f_x, f_y)$



deep imaging

Convolution - Fourier Transform relationship: Convolution Theorem

Convolution theorem. If $\mathcal{F}\{g(x, y)) = G(f_X, f_Y)$ and $\mathcal{F}\{h(x, y)\} = H(f_X, f_Y)$, then

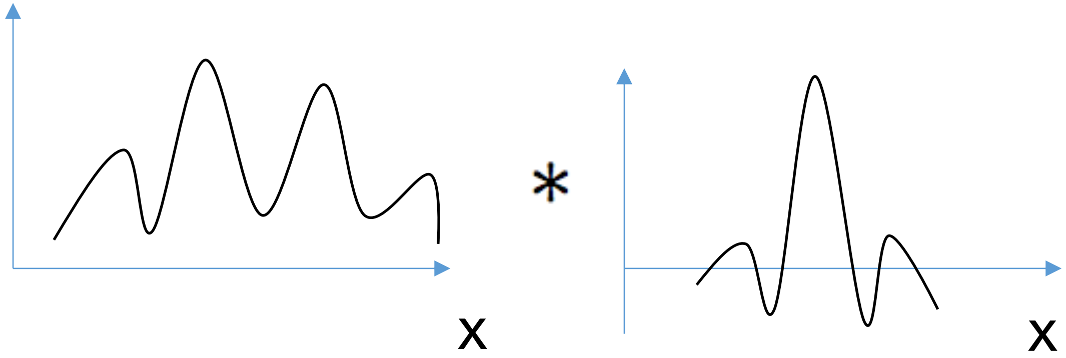
$$\mathcal{F}\left\{\iint\limits_{-\infty}^{\infty} g(\xi, \eta)\, h(x-\xi, y-\eta)\, d\xi\, d\eta\right\} = G(f_X, f_Y)\, H(f_X, f_Y).$$

"The convolution of two functions in space can be performed by a multiplication in the Fourier domain (spatial frequency domain)"

deep imaging

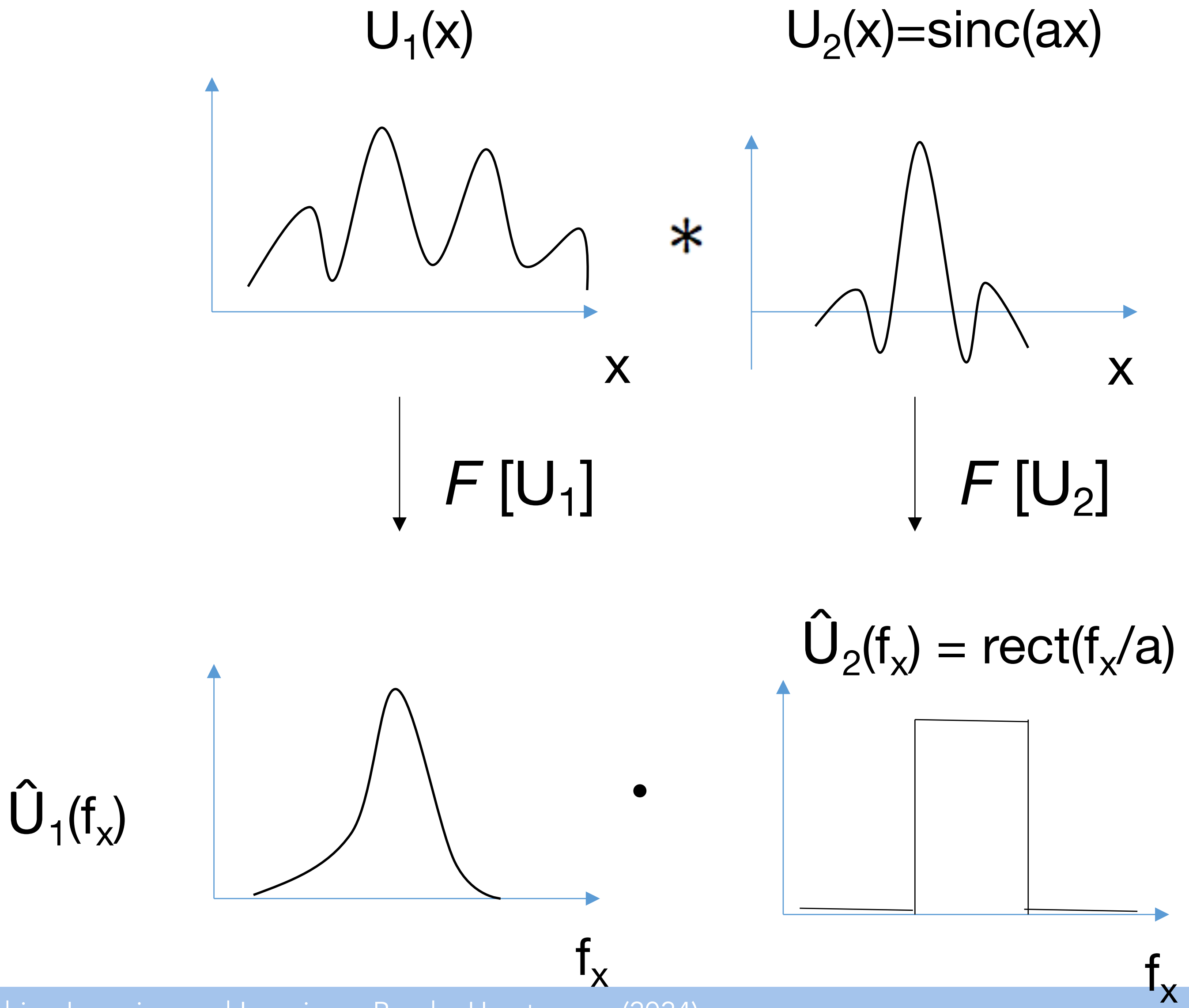
Example of convolution theorem, 1D

$U_1(x)$ $\qquad$ $U_2(x)=\mathrm{sinc}(ax)$

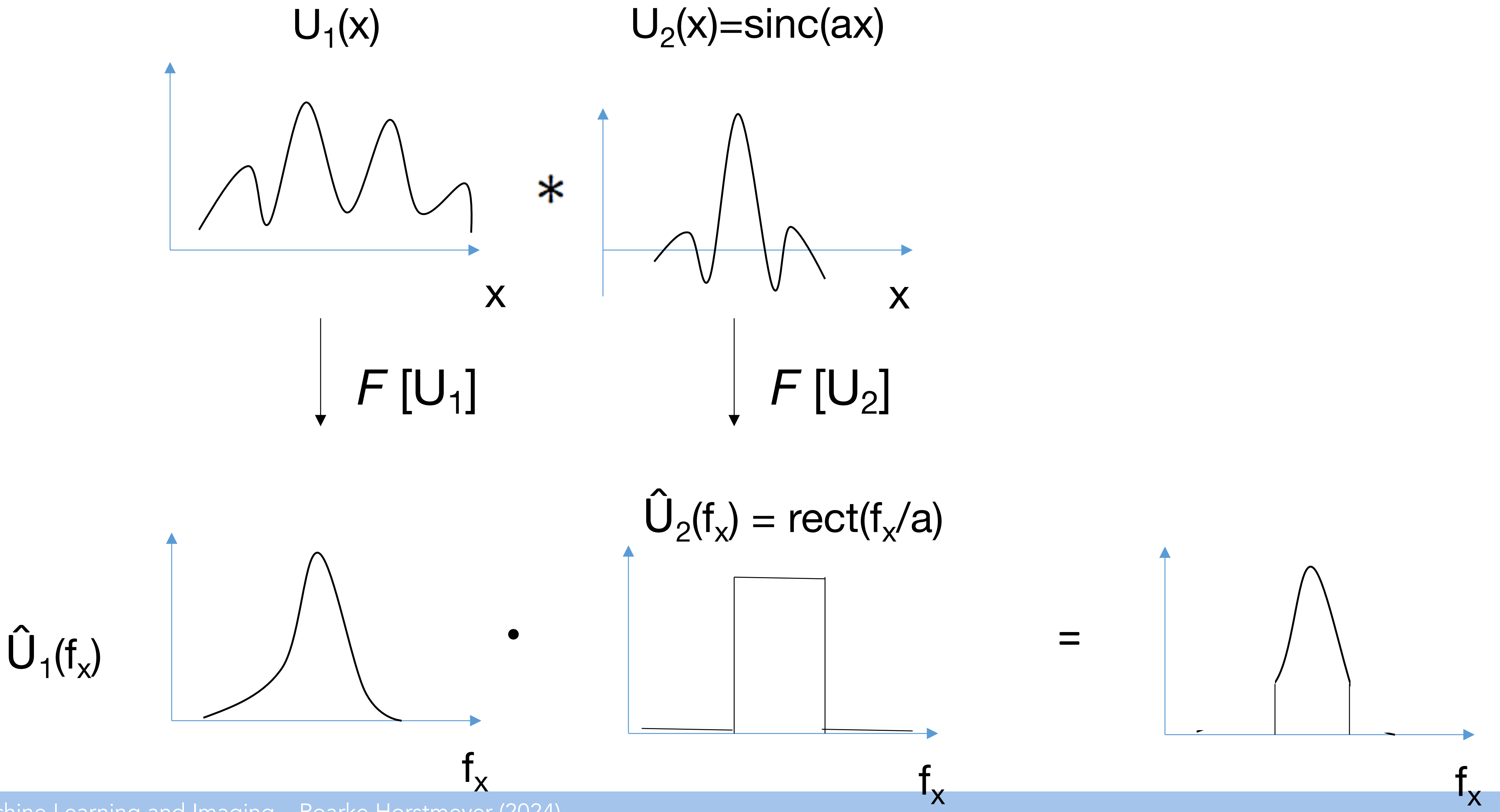


deep imaging

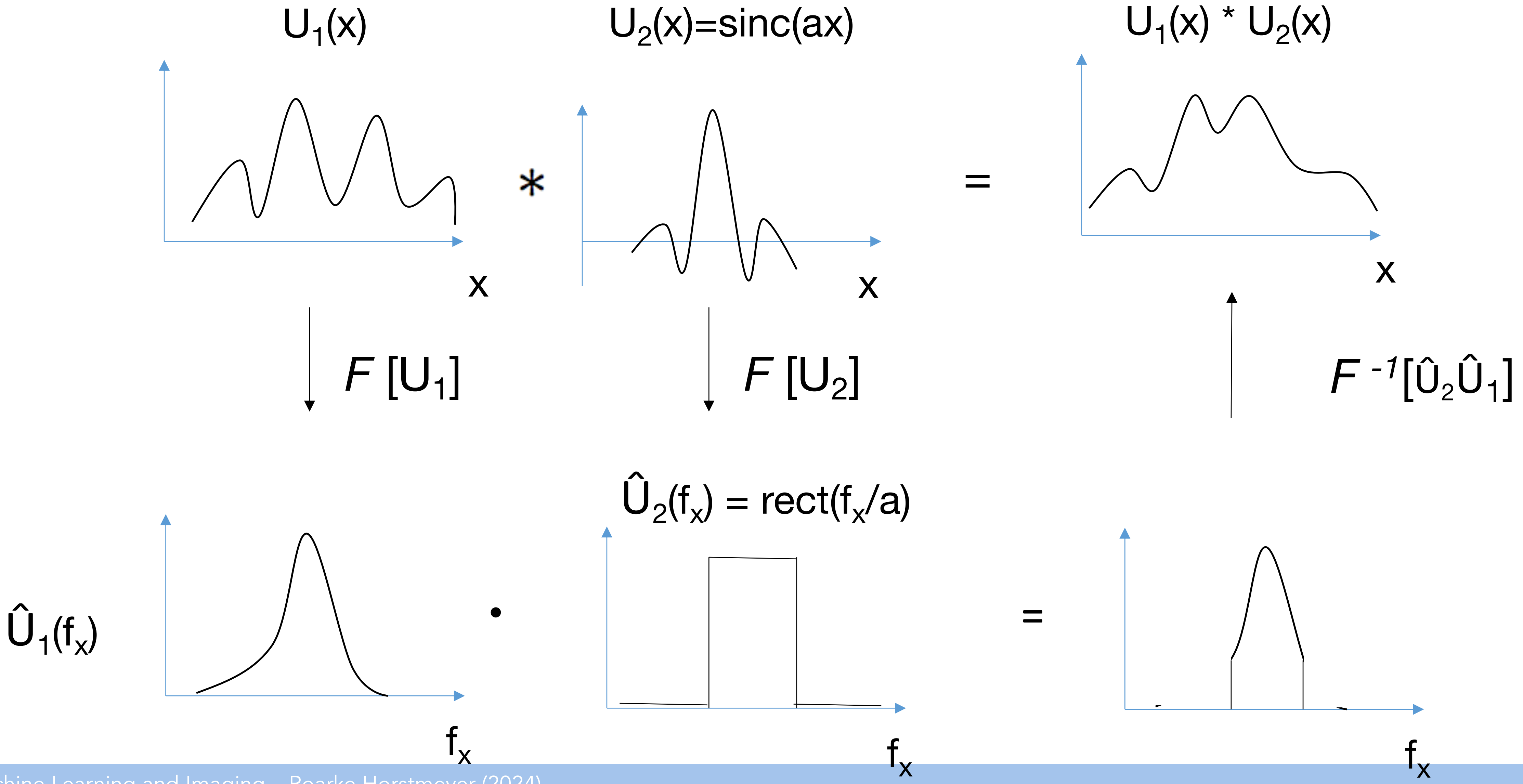
Example of convolution theorem, 1D



Example of convolution theorem, 1D

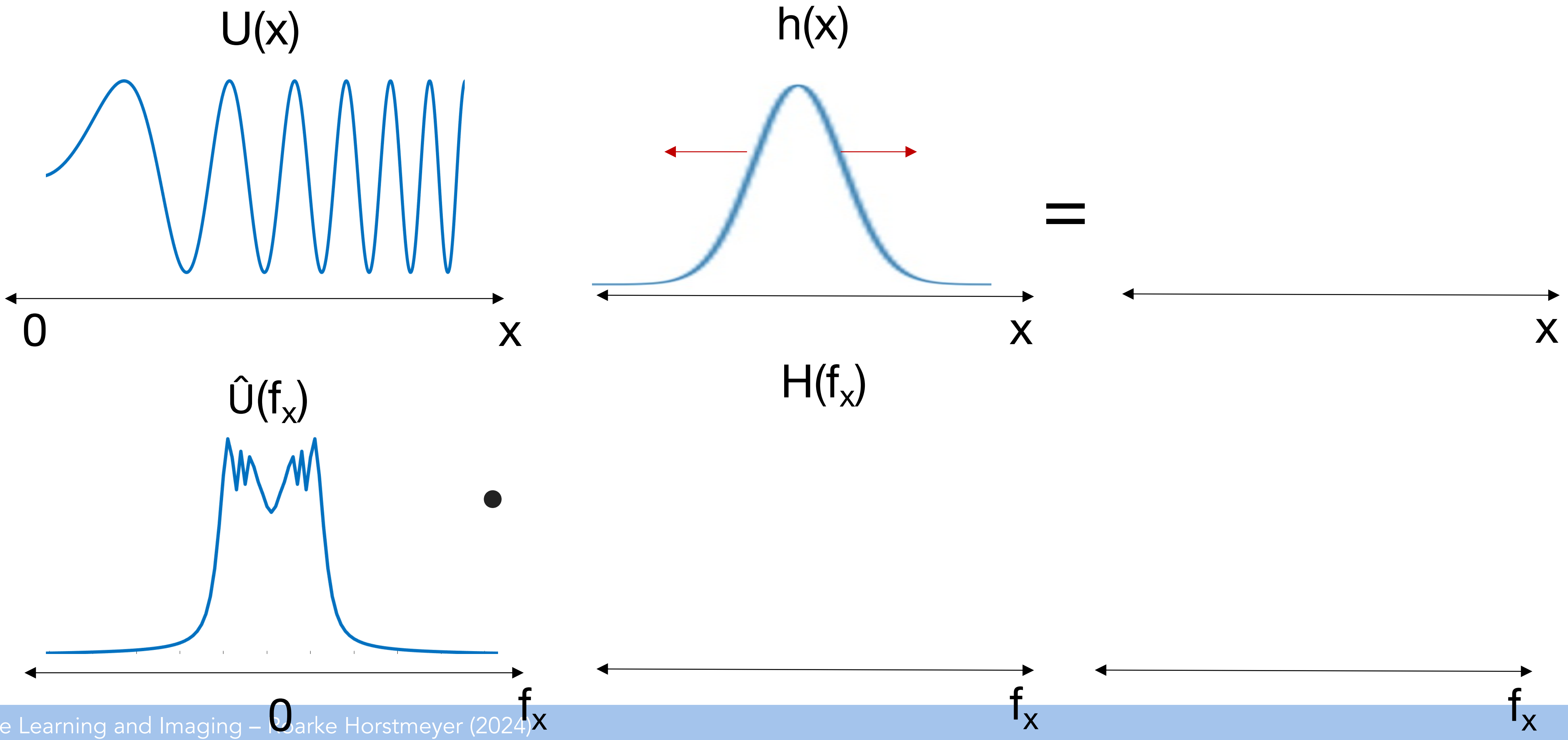


Example of convolution theorem, 1D



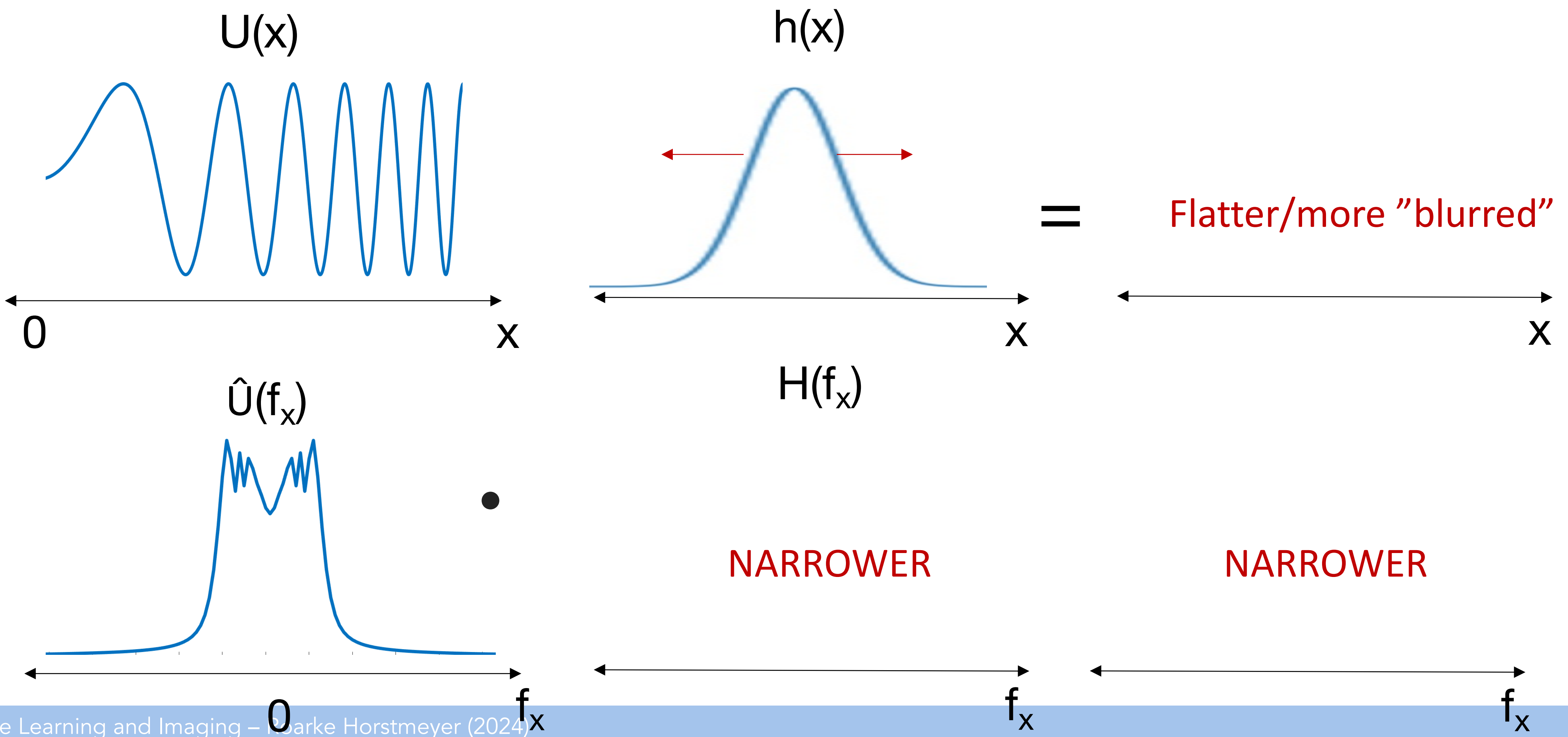
Conceptual questions:

2. Repeat with wider convolution filter:



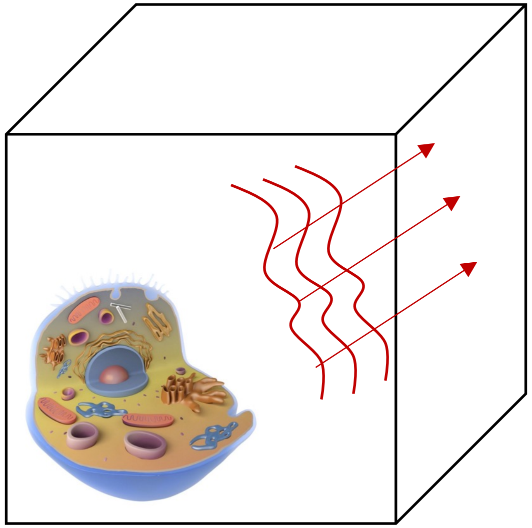
Conceptual questions:

2. Repeat with wider convolution filter:

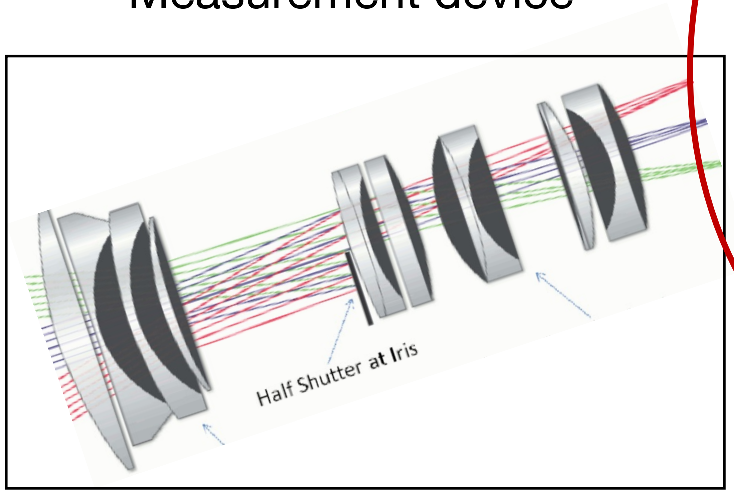


ML+Imaging pipeline introduction

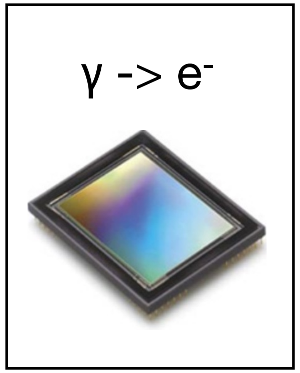
Real World



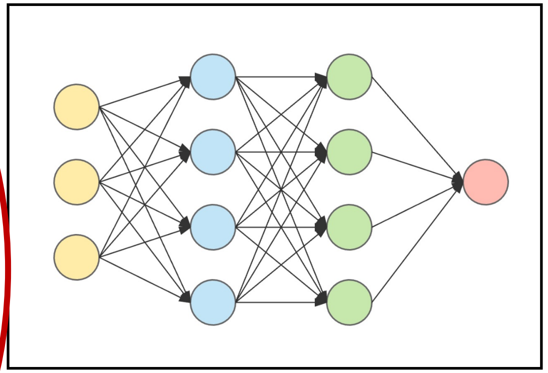
Measurement device



Digitization

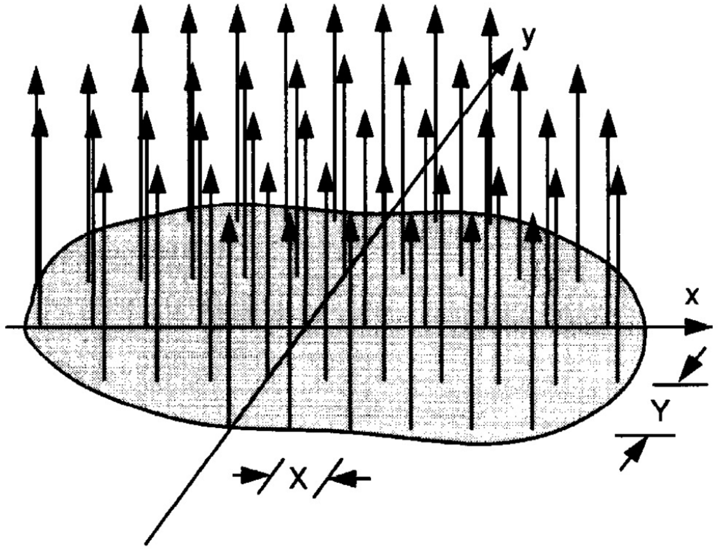


Machine Learning



The Sampling Theorem – from Goodman Section 2.4.1

$$U_s(x, y) = \mathrm{comb}(x/X)\mathrm{comb}(y/Y)U(x, y)$$



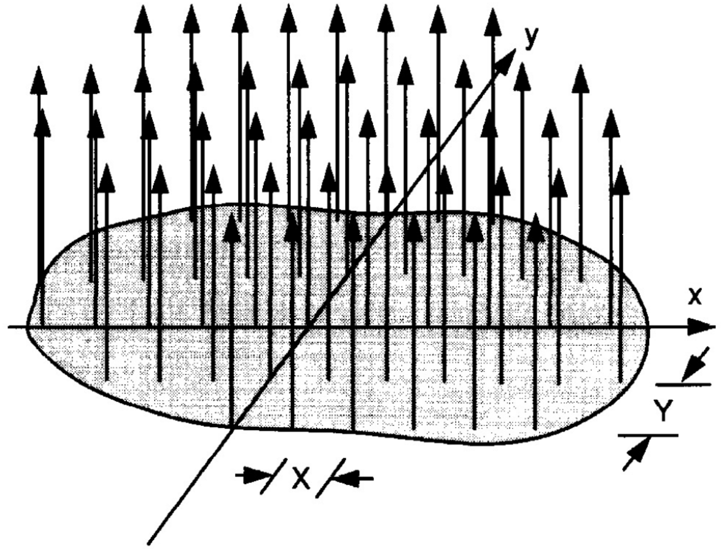
Signal sampling occurs with:

- CMOS (pixel) sensors, PMTs, SPADs
- A-to-D after antennas
- A-to-D after acoustic transducers

Sampling interval width X and Y

The Sampling Theorem – from Goodman Section 2.4.1

$$U_s(x,y) = \text{comb}(x/X)\text{comb}(y/Y)U(x,y)$$



Signal sampling occurs with:

- CMOS (pixel) sensors, PMTs, SPADs
- A-to-D after antennas
- A-to-D after acoustic transducers

Sampling interval width X and Y

$$\hat{U}_s(f_x, f_y) = \mathcal{F}\left[\text{comb}(x/X)\text{comb}(y/Y)\right] * \hat{U}(f_x, f_y)$$

The Sampling Theorem – from Goodman Section 2.4.1

$$\hat{U}_s(f_x, f_y) = \mathcal{F}\left[\text{comb}(x/X)\text{comb}(y/Y)\right] * \hat{U}(f_x, f_y)$$

The Sampling Theorem – from Goodman Section 2.4.1

$$\hat{U}_s(f_x, f_y) = \mathcal{F}\left[\mathrm{comb}(x/X)\mathrm{comb}(y/Y)\right] * \hat{U}(f_x, f_y)$$

$$\mathcal{F}\left[\mathrm{comb}(x/X)\mathrm{comb}(y/Y)\right] = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} \delta\left(f_x - \frac{n}{X}, f_y - \frac{m}{Y}\right)$$

The Sampling Theorem – from Goodman Section 2.4.1

$$\hat{U}_s(f_x, f_y) = \mathcal{F}\left[\mathrm{comb}(x/X)\mathrm{comb}(y/Y)\right] * \hat{U}(f_x, f_y)$$

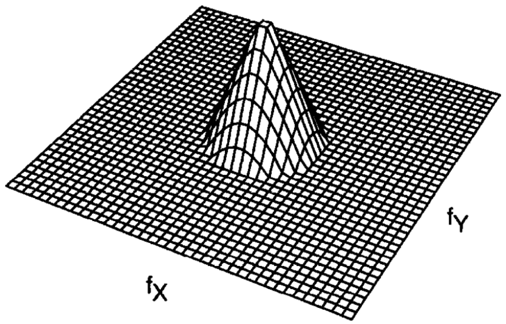
$$\mathcal{F}\left[\mathrm{comb}(x/X)\mathrm{comb}(y/Y)\right] = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \delta\left(f_x - \frac{n}{X}, f_y - \frac{m}{Y}\right)$$

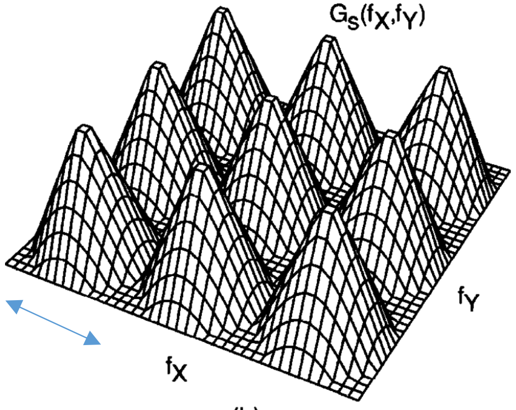
$$\hat{U}_s(f_x, f_y) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \hat{U}\left(f_x - \frac{n}{X}, f_y - \frac{m}{Y}\right)$$

The Sampling Theorem – from Goodman Section 2.4.1

$$\mathcal{F}\left[\text{comb}(x/X)\text{comb}(y/Y)\right] = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}\delta\left(f_x - \frac{n}{X}, f_y - \frac{m}{Y}\right)$$

$$\hat{U}_s(f_x, f_y) = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}\hat{U}\left(f_x - \frac{n}{X}, f_y - \frac{m}{Y}\right)$$



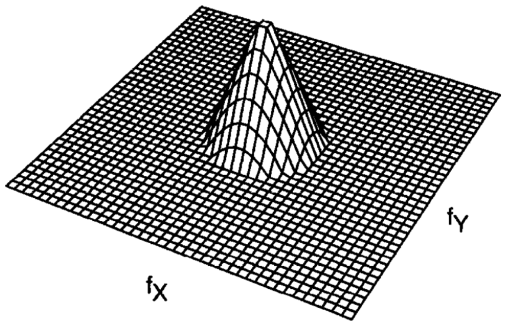


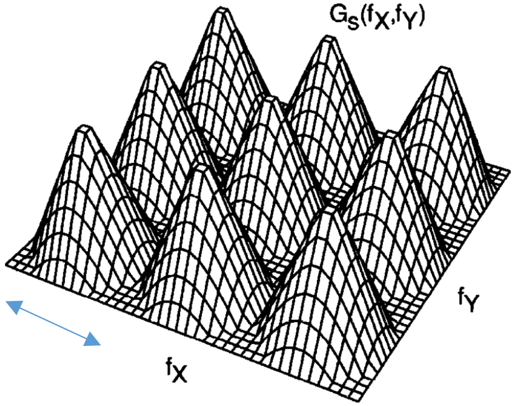
Signal extends from ($-B_x$, $-B_y$) to (B_x, B_y) in Fourier domain

The Sampling Theorem – from Goodman Section 2.4.1

$$\mathcal{F}\left[\mathrm{comb}(x/X)\mathrm{comb}(y/Y)\right] = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}\delta\left(f_x - \frac{n}{X}, f_y - \frac{m}{Y}\right)$$

$$\hat{U}_s(f_x, f_y) = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}\hat{U}\left(f_x - \frac{n}{X}, f_y - \frac{m}{Y}\right)$$





Signal extends from ($-B_x$, $-B_y$) to (B_x, B_y) in Fourier domain

Mask out copies with a rect function:

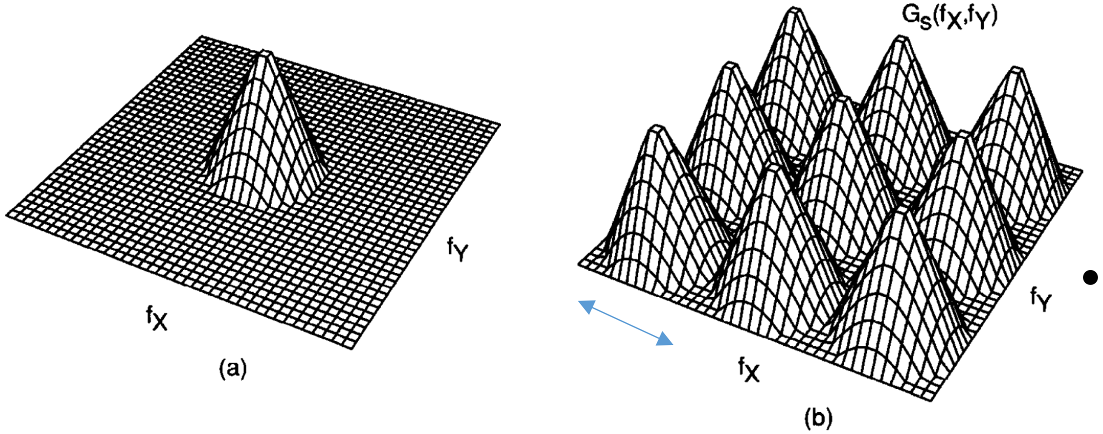
$$\mathrm{rect}\left(\frac{f_x}{2B_x}\right)\mathrm{rect}\left(\frac{f_y}{2B_y}\right)\hat{U}_s(f_x, f_y) = \hat{U}(f_x, f_y)$$

Bandwidth (B_x, B_y) of signal

The Sampling Theorem – from Goodman Section 2.4.1

$$\mathcal{F}\left[\text{comb}(x/X)\text{comb}(y/Y)\right] = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}\delta\left(f_x - \frac{n}{X}, f_y - \frac{m}{Y}\right)$$

$$\hat{U}_s(f_x, f_y) = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}\hat{U}\left(f_x - \frac{n}{X}, f_y - \frac{m}{Y}\right)$$



•

Signal extends from ($-B_x$, $-B_y$) to (B_x, B_y) in Fourier domain

Mask out copies with a rect function:

$$\text{rect}\left(\frac{f_x}{2B_x}\right)\text{rect}\left(\frac{f_y}{2B_y}\right)\hat{U}_s(f_x, f_y) = \hat{U}(f_x, f_y)$$

$$\mathrm{rect}\left(\frac{f_x}{2B_x}\right)\mathrm{rect}\left(\frac{f_y}{2B_y}\right)\hat{U}_s(f_x, f_y) = \hat{U}(f_x, f_y)$$

$F[\bullet]$ $F[\bullet]$

deep imaging

deep imaging

$$\mathrm{rect}\left(\frac{f_x}{2B_x}\right)\mathrm{rect}\left(\frac{f_y}{2B_y}\right)\hat{U}_s(f_x, f_y) = \hat{U}(f_x, f_y)$$

$$\Big\downarrow F[\bullet]$$

$$U(x, y)\mathrm{comb}(x/X)\mathrm{comb}(y/Y)$$

deep imaging

$$\mathrm{rect}\left(\frac{f_x}{2B_x}\right)\mathrm{rect}\left(\frac{f_y}{2B_y}\right)\hat{U}_s(f_x, f_y) = \hat{U}(f_x, f_y)$$

$F\,[\bullet]$

$$h(x, y) = 4B_xB_y\mathrm{sinc}(2B_xx)\mathrm{sinc}(2B_yy)$$

deep imaging

$$\text{rect}\left(\frac{f_x}{2B_x}\right)\text{rect}\left(\frac{f_y}{2B_y}\right)\hat{U}_s(f_x, f_y) = \hat{U}(f_x, f_y)$$

$F\,[\bullet]$

$$h(x,y) = 4B_xB_y\text{sinc}(2B_xx)\text{sinc}(2B_yy)$$

$$h(x,y) * (U(x,y)\text{comb}(x/X)\text{comb}(y/Y)) = U(x,y)$$

$$\mathrm{rect}\left(\frac{f_x}{2B_x}\right)\mathrm{rect}\left(\frac{f_y}{2B_y}\right)\hat{U}_s(f_x,f_y) = \hat{U}(f_x,f_y)$$

$$\xrightarrow{F[\bullet]} \quad h(x,y) = 4B_xB_y\mathrm{sinc}(2B_xx)\mathrm{sinc}(2B_yy)$$

$$h(x,y) * (U(x,y)\mathrm{comb}(x/X)\mathrm{comb}(y/Y)) = U(x,y)$$

$$U(x,y)\mathrm{comb}(x/X)\mathrm{comb}(y/Y) = XY\sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}U(nX,mY)\delta(x-nX,y-mY)$$

$$\mathrm{rect}\left(\frac{f_x}{2B_x}\right)\mathrm{rect}\left(\frac{f_y}{2B_y}\right)\hat{U}_s(f_x,f_y)=\hat{U}(f_x,f_y)$$

$F[\bullet]$

$$h(x,y)=4B_xB_y\mathrm{sinc}(2B_xx)\mathrm{sinc}(2B_yy)$$

$$h(x,y)*(U(x,y)\mathrm{comb}(x/X)\mathrm{comb}(y/Y))=U(x,y)$$

$U_s(x,y)$ (from beginning) =

$$U(x,y)\mathrm{comb}(x/X)\mathrm{comb}(y/Y)=XY\sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}U(nX,mY)\delta(x-nX,y-mY)$$

$$U(x,y)=4B_xB_yXY\sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}U(nX,mY)\mathrm{sinc}\left[2B_x(x-nX)\right]\mathrm{sinc}\left[2B_y(y-mY)\right]$$

deep imaging

deep imaging

The Sampling Theorem

When sampled appropriately, a discrete signal can *exactly* reproduce a continuous signal:

$$U(x,y) = 4B_x B_y XY \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} U(nX,mY)\mathrm{sinc}\left[2B_x(x-nX)\right]\mathrm{sinc}\left[2B_y(y-mY)\right]$$

Continuous signal:

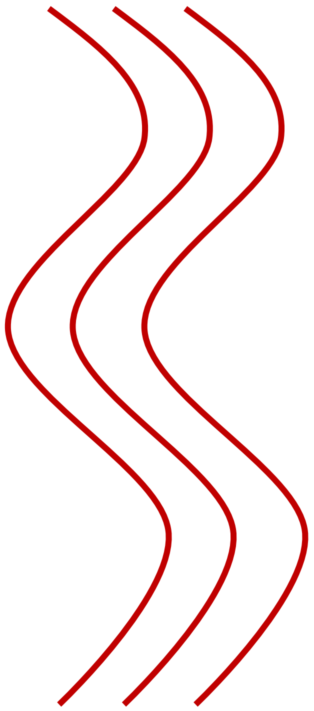
- EM field
- Sound wave
- MR signal

Discretized signal:

- Detected EM field
- Sampled sound wave
- Sampled MR signal

What does the Sampling Theorem mean for us?

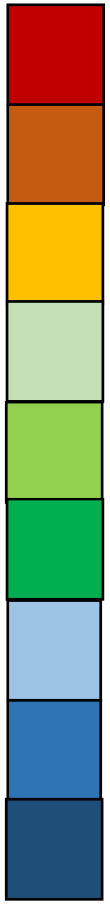
Continuous fields



*

(*) Under certain conditions

Discretize vectors (and matrices)



17
20
22
21
23
25
24
26
29

Conditions to safely apply the sampling theorem

$$U(x,y) = 4B_x B_y XY \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} U(nX, mY)\text{sinc}\left[2B_x(x-nX)\right]\text{sinc}\left[2B_y(y-mY)\right]$$

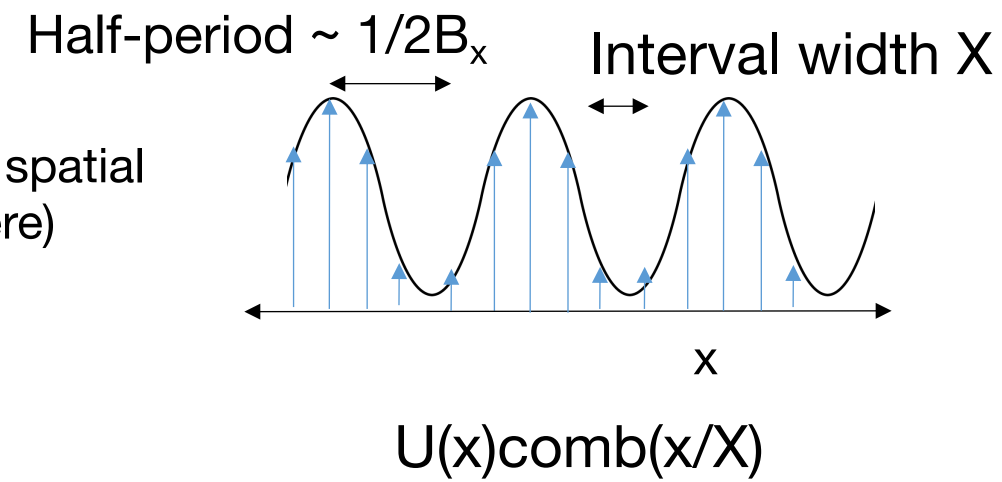
- Sampling must be proportional to bandwidth ($2B_x$ and $2B_y$)
  - "Nyquist" sampling: $X = 1/2B_x$, $Y = 1/2B_y$

Conditions to safely apply the sampling theorem

$$U(x,y) = 4B_xB_yXY \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} U(nX,mY)\mathrm{sinc}\left[2B_x(x-nX)\right]\mathrm{sinc}\left[2B_y(y-mY)\right]$$

- Sampling must be proportional to bandwidth ($2B_x$ and $2B_y$)
  - “Nyquist” sampling: $X = 1/2B_x$, $Y = 1/2B_y$

(Assume largest spatial freq. in signal here)



$X < 1/2B_x$ ✔

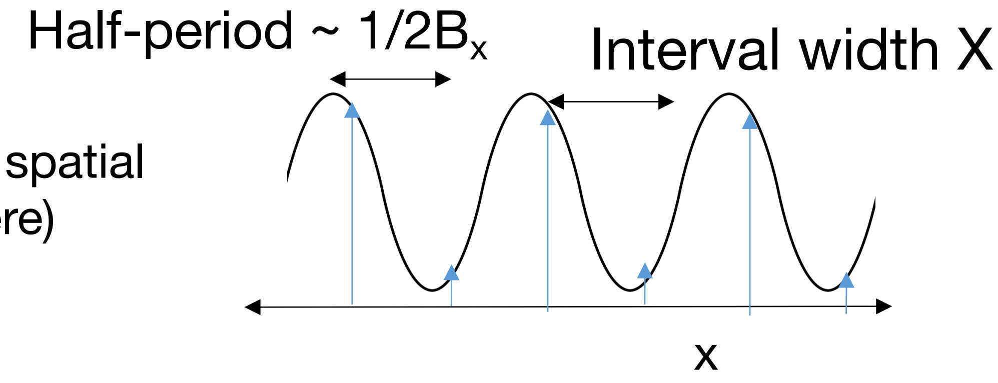
deep imaging

Conditions to safely apply the sampling theorem

$$U(x,y) = 4B_x B_y XY \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} U(nX, mY) \operatorname{sinc}\left[2B_x(x-nX)\right] \operatorname{sinc}\left[2B_y(y-mY)\right]$$

- Sampling must be proportional to bandwidth ($2B_x$ and $2B_y$)
  - "Nyquist" sampling: $X = 1/2B_x$, $Y = 1/2B_y$

(Assume largest spatial freq. in signal here)



U(x)comb(x/X)

$X = 1/2B_x$ ✔

Nyquist sampling – still sampling peak and trough

deep imaging

Conditions to safely apply the sampling theorem

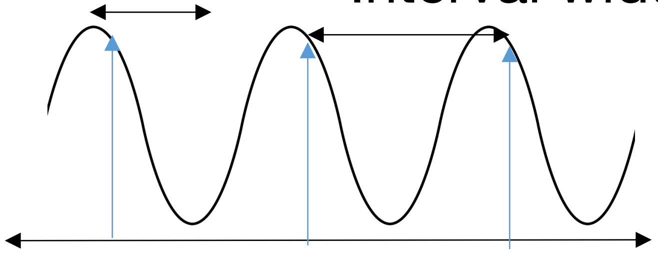
$$U(x,y) = 4B_xB_yXY \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} U(nX,mY)\text{sinc}\left[2B_x(x-nX)\right]\text{sinc}\left[2B_y(y-mY)\right]$$

- Sampling must be proportional to bandwidth ($2B_x$ and $2B_y$)
  - "Nyquist" sampling: $X = 1/2B_x$, $Y = 1/2B_y$

(Assume largest spatial freq. in signal here)

Half-period ~ $1/2B_x$

Interval width X



x

U(x)comb(x/X)

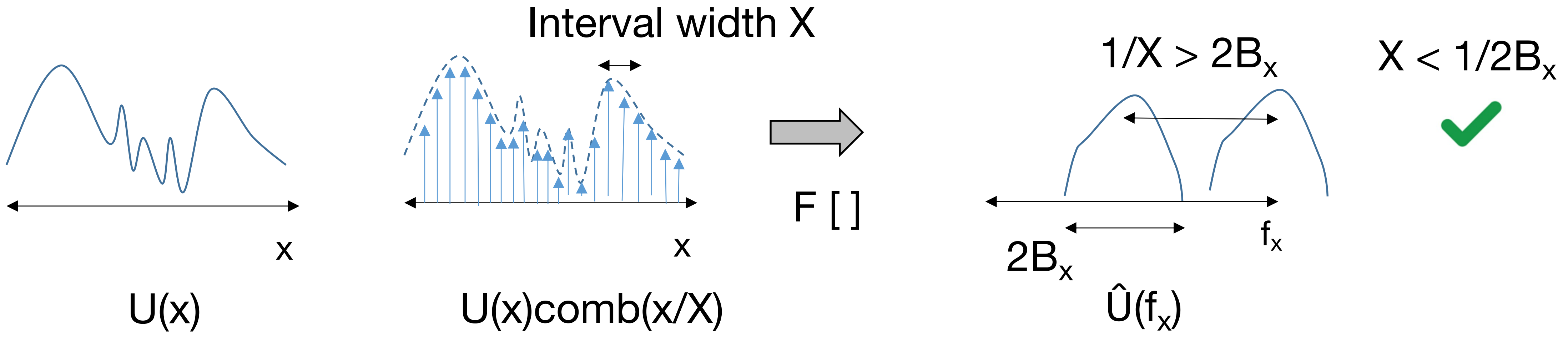
$X > 1/2B_x$ 

Can't detect the frequency anymore!

Conditions to safely apply the sampling theorem

$$U(x,y) = 4B_xB_yXY \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} U(nX,mY)\mathrm{sinc}\left[2B_x(x-nX)\right]\mathrm{sinc}\left[2B_y(y-mY)\right]$$

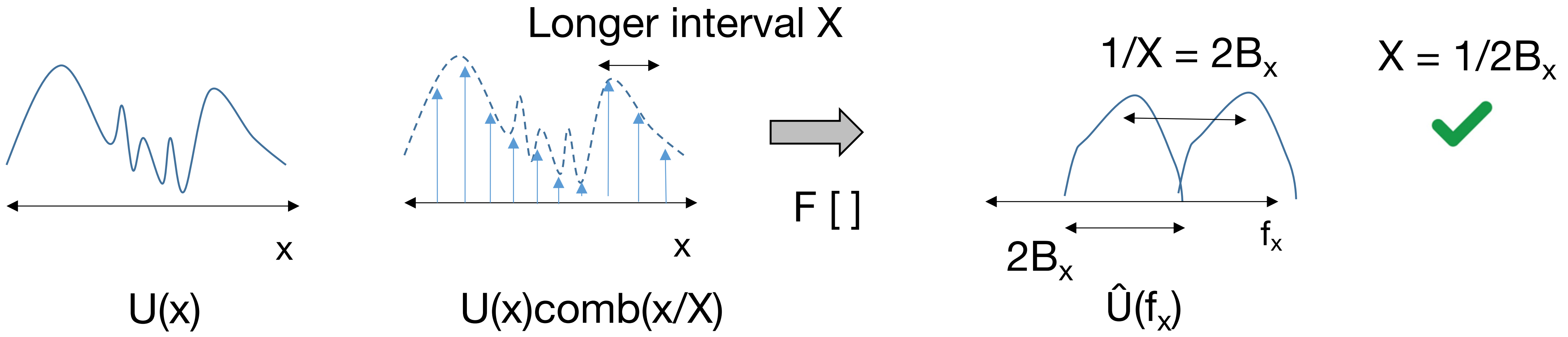
- Sampling must be proportional to bandwidth ($2B_x$ and $2B_y$)
  - "Nyquist" sampling: $X = 1/2B_x$, $Y = 1/2B_y$
  - Needed to avoid aliasing



Conditions to safely apply the sampling theorem

$$U(x,y) = 4B_xB_yXY \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} U(nX,mY)\mathrm{sinc}\left[2B_x(x-nX)\right]\mathrm{sinc}\left[2B_y(y-mY)\right]$$

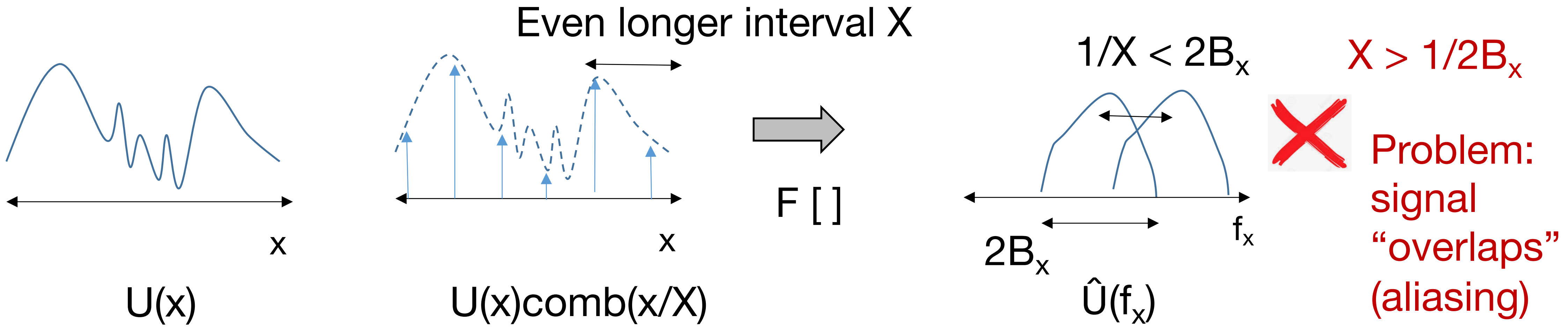
- Sampling must be proportional to bandwidth ($2B_x$ and $2B_y$)
  - "Nyquist" sampling: $X = 1/2B_x$, $Y = 1/2B_y$
  - Needed to avoid aliasing



Conditions to safely apply the sampling theorem

$$U(x,y) = 4B_xB_yXY \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} U(nX,mY)\,\mathrm{sinc}\left[2B_x(x-nX)\right]\mathrm{sinc}\left[2B_y(y-mY)\right]$$

- Sampling must be proportional to bandwidth ($2B_x$ and $2B_y$)
  - "Nyquist" sampling: $X = 1/2B_x$, $Y = 1/2B_y$
  - Needed to avoid aliasing



Linear Algebra – notation and basics

- We'll (try to) write *column* vectors as lower case variables

- Row vectors will be denoted as the transpose

- We'll try to write matrices as upper case variables

- We'll try to denote if a matrix/vector is real, complex etc. and its size with a certain notation

Linear Algebra – notation and basics

Some basic vector operations you should know:

- Conjugate, transpose, conjugate transpose
- Inner product
- Hadamard (element-wise, dot-times) product
- outer product
- Vector (matrix) addition
- matrix-vector product
- convolution

Matrix-vector products – two useful interpretations

1. Inner products per entry:

$$\underset{T(1,:)}{\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}} \quad \underset{u}{\begin{bmatrix} 10 \\ 11 \\ 12 \end{bmatrix}} \longrightarrow \underset{v}{\begin{bmatrix} 68 \\ \; \\ \; \end{bmatrix}} = \underset{T}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}} \underset{u}{\begin{bmatrix} 10 \\ 11 \\ 12 \end{bmatrix}}$$

Matrix-vector products – two useful interpretations

1. Inner products per entry:

T(2,:)

4	5	6

u

10
11
12

v = T u

v

68
167

T

1	2	3
4	5	6
7	8	9

u

10
11
12

Matrix-vector products – two useful interpretations

1. Inner products per entry:

$$\overset{T(3,:)}{\begin{bmatrix} 7 & 8 & 9 \end{bmatrix}} \; \overset{u}{\begin{bmatrix} 10 \\ 11 \\ 12 \end{bmatrix}} \longrightarrow \overset{v}{\begin{bmatrix} 68 \\ 167 \\ 266 \end{bmatrix}} = \overset{T}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}} \; \overset{u}{\begin{bmatrix} 10 \\ 11 \\ 12 \end{bmatrix}}$$

Matrix-vector products – two useful interpretations

1. Inner products per entry:

$$\underset{v}{\begin{bmatrix} 68 \\ 167 \\ 266 \end{bmatrix}} = \underset{T}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}} \underset{u}{\begin{bmatrix} 10 \\ 11 \\ 12 \end{bmatrix}}$$

2. Weighted column sum:

$$\underset{v}{\begin{bmatrix} 68 \\ 167 \\ 266 \end{bmatrix}} = 10 \underset{T(:,1)}{\begin{bmatrix} 1 \\ 4 \\ 7 \end{bmatrix}} + 11 \underset{T(:,2)}{\begin{bmatrix} 2 \\ 5 \\ 8 \end{bmatrix}} + 12 \underset{T(:,3)}{\begin{bmatrix} 3 \\ 6 \\ 9 \end{bmatrix}}$$

deep imaging

Discrete convolution

$$V(x_o) = \int_{-\infty}^{\infty} U(x_i) h(x_o - x_i) dx_i$$

$$\downarrow$$

$$v[x_0] = \sum_{x_i=-M}^{M} u[x_i] h[x_o - x_i]$$

Discrete 1D Convolution – an example

Steps to follow:

Step 1	List the index 'k' covering a sufficient range
Step 2	List the input x[k]
Step 3	Obtain the reversed sequence h[-k], and align the rightmost element of h[n-k] to the leftmost element of x[k]
Step 4	Cross-multiply and sum the nonzero overlap terms to produce y[n]
Step 5	Slide h[n-k] to the right by one position
Step 6	Repeat step 4; stop if all the output values are zero or if required.

http://host.uniroma3.it/laboratori/sp4te/teaching/sp4bme/documents/LectureConvolution.pdf

Example 2: Find the convolution of the two sequences x[n] and h[n] given by,

$$x[k] = \underset{\uparrow}{[3}\ 1\ 2] \qquad h[k] = [3\ \underset{\uparrow}{2}\ 1]$$

k:	-2	-1	0	1	2	3	4	5
x[k]:			3	1	2			
h[-k]:	1	2	3					
h[1-k]:		1	2	3				
h[2-k]:			1	2	3			
h[3-k]:				1	2	3		
h[4-k]:					1	2	3	
h[5-k]:						1	2	3

Hint: The value of k starts from **(– length of h + 1)** and continues till **(length of h + length of x – 1)**

Here ***k*** starts from **-3 + 1 = -2** and continues till **3 + 3 – 1 = 5**

http://host.uniroma3.it/laboratori/sp4te/teaching/sp4bme/documents/LectureConvolution.pdf

Example 2: Find the convolution of the two sequences x[n] and h[n] given by,

$$x[k] = [\underset{\uparrow}{3}\ 1\ 2] \qquad h[k] = [\underset{\uparrow}{3}\ 2\ 1]$$

k:	-2	-1	0	1	2	3	4	5
x[k]:			3	1	2			
h[-k]:	1	2	3					
h[1-k]:		1	2	3				
h[2-k]:			1	2	3			
h[3-k]:				1	2	3		
h[4-k]:					1	2	3	
h[5-k]:						1	2	3
y:			9					

deep imaging

Example 2: Find the convolution of the two sequences x[n] and h[n] given by,

$$x[k] = [\underset{\uparrow}{3}\ 1\ 2] \qquad h[k] = [\underset{\uparrow}{3}\ 2\ 1]$$

k:	-2	-1	0	1	2	3	4	5
x[k]:			$\boxed{3}$	$\boxed{1}$	2			
h[-k]:	1	2	3					
h[1-k]:		1	$\boxed{2}$	$\boxed{3}$				
h[2-k]:			1	2	3			
h[3-k]:				1	2	3		
h[4-k]:					1	2	3	
h[5-k]:						1	2	3
y:			9	6+3				

Example 2: Find the convolution of the two sequences x[n] and h[n] given by,

$$x[k] = [\underset{\uparrow}{3}\ 1\ 2] \qquad h[k] = [3\ \underset{\uparrow}{2}\ 1]$$

k:	-2	-1	0	1	2	3	4	5
x[k]:			3	1	2			
h[-k]:	1	2	3					
h[1-k]:		1	2	3				
h[2-k]:			1	2	3			
h[3-k]:				1	2	3		
h[4-k]:					1	2	3	
h[5-k]:						1	2	3
y:			9	6+3	3+2+6			

deep imaging

Example 2: Find the convolution of the two sequences x[n] and h[n] given by,

x[k] = [3 1 2] h[k] = [3 2 1]

k:	-2	-1	0	1	2	3	4	5
x[k]:			3	$\boxed{1}$	$\boxed{2}$	$\boxed{\ }$		
h[-k]:	1	2	3					
h[1-k]:		1	2	3				
h[2-k]:			1	2	3			
h[3-k]:				$\boxed{1}$	$\boxed{2}$	$\boxed{3}$		
h[4-k]:					1	2	3	
h[5-k]:						1	2	3
y:			9	6+3	3+2+6	1+4+0		

deep imaging

Example 2: Find the convolution of the two sequences x[n] and h[n] given by,

$$x[k] = [\underset{\uparrow}{3}\ 1\ 2] \qquad h[k] = [3\ \underset{\uparrow}{2}\ 1]$$

k:	-2	-1	0	1	2	3	4	5
x[k]:			3	$\boxed{1}$	$\boxed{2}$	$\boxed{\ }$		
h[-k]:	1	2	3					
h[1-k]:		1	2	3				
h[2-k]:			1	2	3			
h[3-k]:				$\boxed{1}$	$\boxed{2}$	$\boxed{3}$		
h[4-k]:					1	2	3	
h[5-k]:						1	2	3
y:			9	6+3	3+2+6	1+4+0		

y: [9 9 11 5 2 0]

deep imaging

Discrete convolution

$$V(x_o) = \int_{-\infty}^{\infty} U(x_i) h(x_o - x_i) dx_i$$

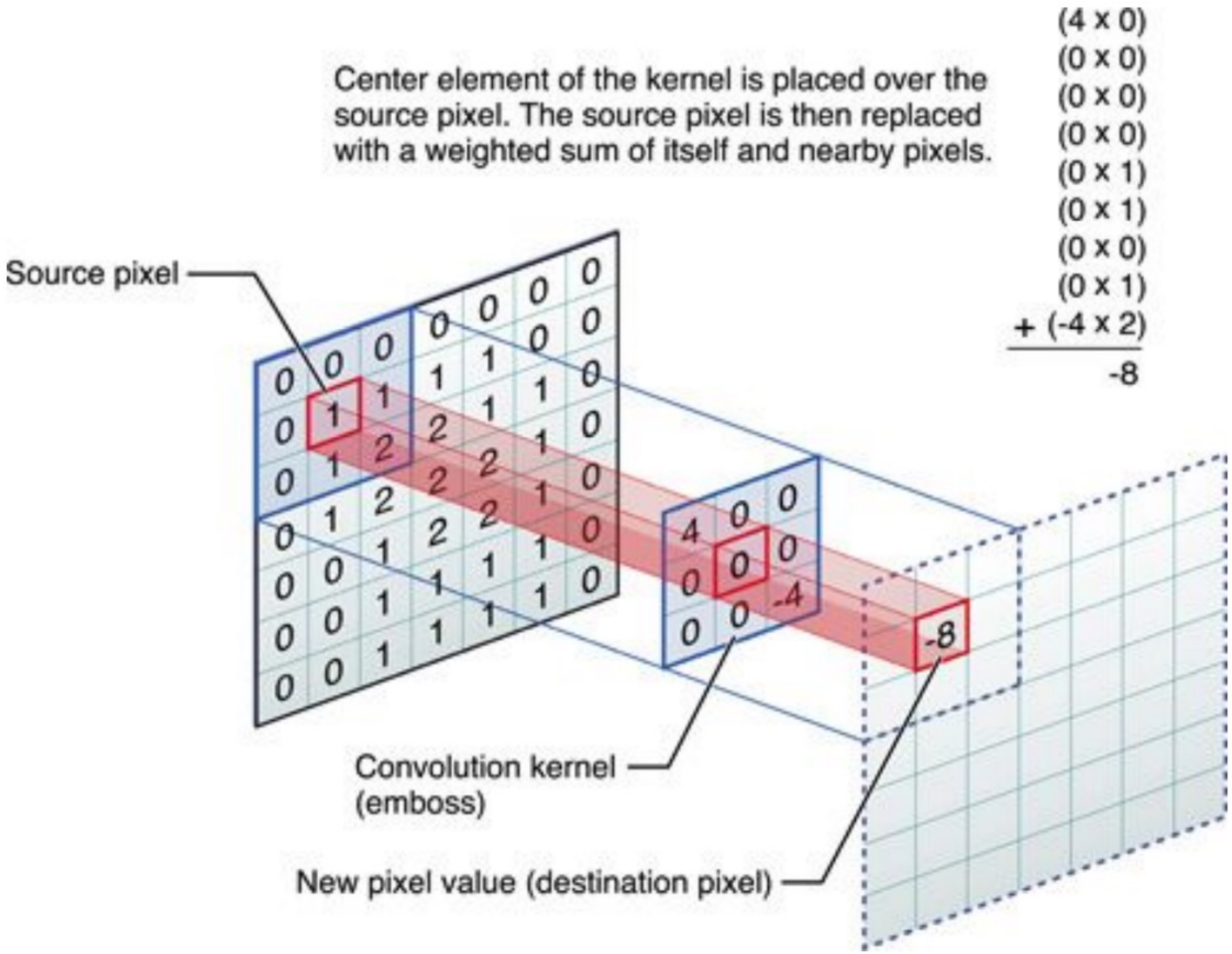
$$v[x_0] = \sum_{x_i=-M}^{M} u[x_i] h[x_o - x_i]$$

Discrete 2D convolution

$$V(x_o, y_o) = \iint_{-\infty}^{\infty} U(x_i, y_i) h(x_o - x_i, y_o - y_i) dx_i dy_i$$

$$v[x_0, y_o] = \sum_{y_i=-L}^{L} \sum_{x_i=-M}^{M} u[x_i, y_i] h[x_o - x_i, y_o - y_i]$$

Discrete 2D convolution



https://www.psi.toronto.edu/~jimmy/ece521/Tut1.pdf

Discrete 2D convolution: edge conditions and even kernels

deep imaging

From MATLAB definition of conv2:

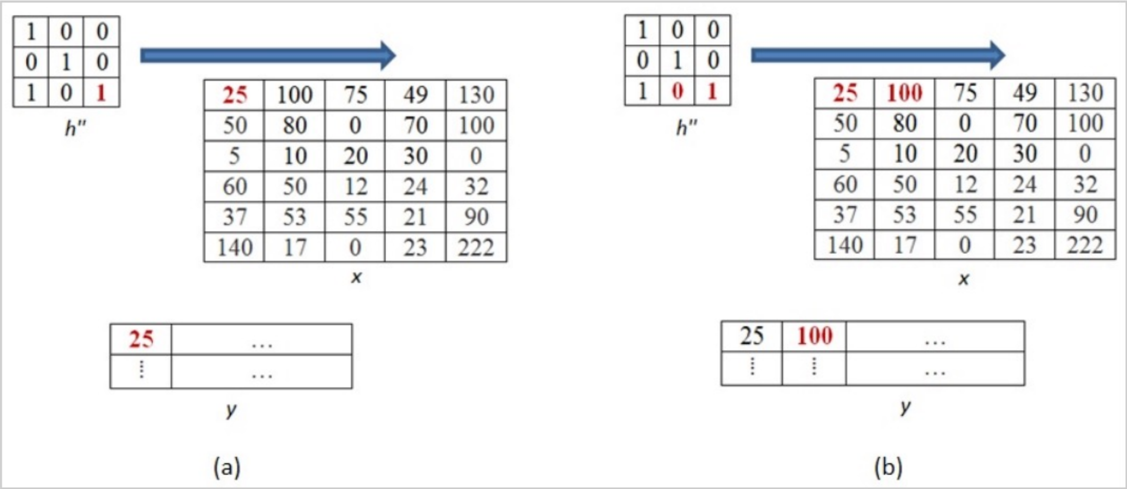
2-D Convolution

For discrete, two-dimensional variables A and B, the following equation defines the convolution of A and B:

$$C(j,k) = \sum_p \sum_q A(p,q)B(j-p+1,k-q+1)$$

p and q run over all values that lead to legal subscripts of $A(p,q)$ and $B(j\text{-}p\text{+}1,k\text{-}q\text{+}1)$.

i=1, j=1: Start in the upper left corner at A(1,1) with *lower right* of flipped version of B [B(1,1)]:



Output: 8 x 8

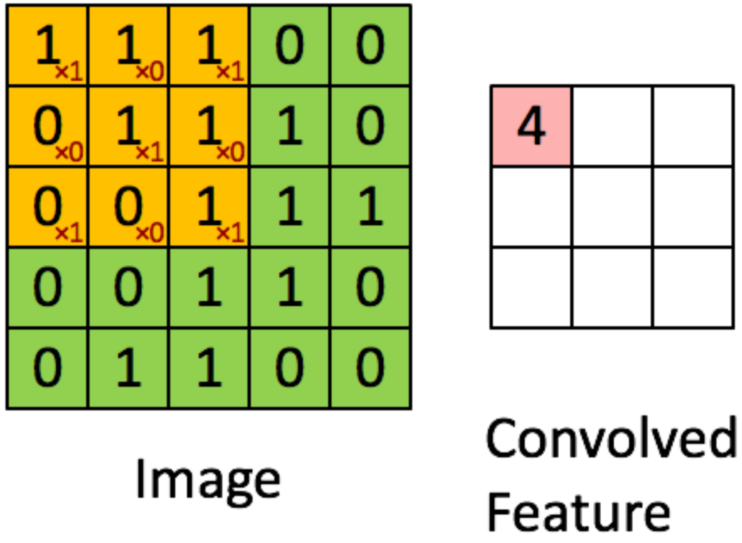
25	100	100	149	205	49	130
50	105	150	225	149	200	100
5	60	130	140	165	179	130
60	55	132	174	74	94	132
37	113	147	96	189	83	90
140	54	253	145	255	137	254
0	140	54	53	78	243	90
0	0	140	17	0	23	222

- For corner-to-corner alignment, doesn't matter if matrix size is even or odd
- Output matrix will be larger than input matrices

Discrete 2D convolution: edge conditions and even kernels

From Tensorflow definition of conv2:

```
output[b, i, j, k] =
    sum_{di, dj, q} input[b, strides[1] * i + di, strides[2] * j + dj, q]
                    * filter[di, dj, q, k]
```



Start convolution kernel *inside* image: align upper-left of image A with *upper right* of kernel B

- Output matrix will be smaller than input image and filter
- We will work through these numbers carefully!

Linear Algebra – notation and basics

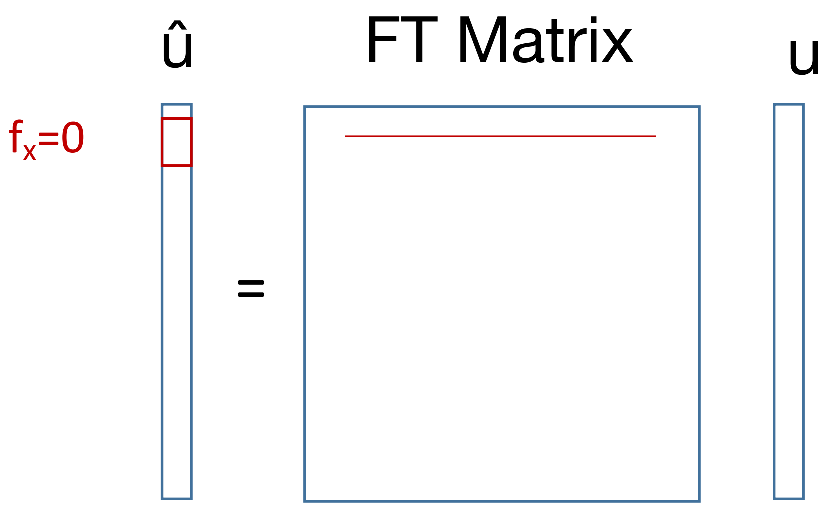
Some basic types of matrices & terms that you should know about:

- Symmetric (Hermitian) matrix: $A=A^T$ if A is real, $A=A^H$ if A is complex
- Square, hot-dog and hamburger matrices
- Invertible matrix
- Diagonal matrix
- Toeplitz matrix
- Banded matrix

Discrete Fourier Transforms

$$\hat{U}(f_x) = \int_{-\infty}^{\infty} U(x) \exp\left(-2\pi i (f_x x)\right) dx$$

$$\hat{u}[f_x] = \sum_{x=0}^{M-1} u[x] \exp\left(-2\pi i f_x x / M\right)$$



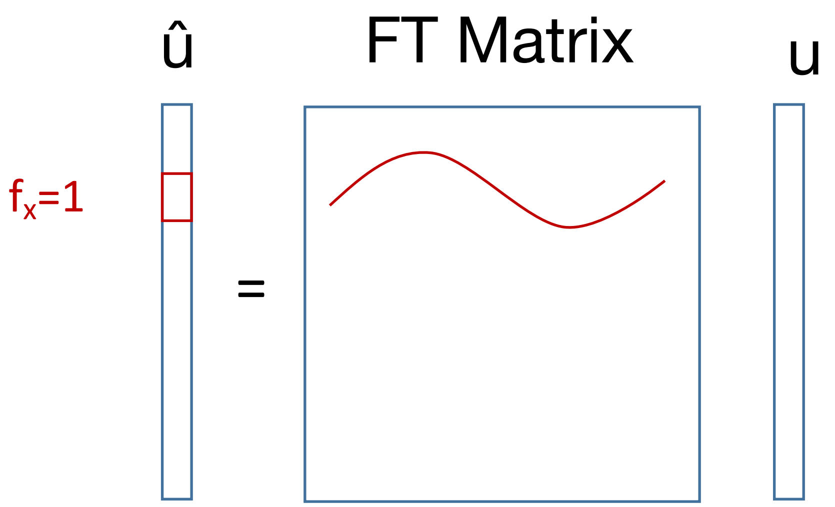
Inner product of u with different complex expon.

- np.fft(u), np.fftshift(np.fft(np.ifftshift(u)))
- fft = fast Fourier transform, much more comp. efficient than matrix multiplication!

Discrete Fourier Transforms

$$\hat{U}(f_x) = \int_{-\infty}^{\infty} U(x) \exp\left(-2\pi i (f_x x)\right) dx$$

$$\hat{u}[f_x] = \sum_{x=0}^{M-1} u[x] \exp\left(-2\pi i f_x x / M\right)$$



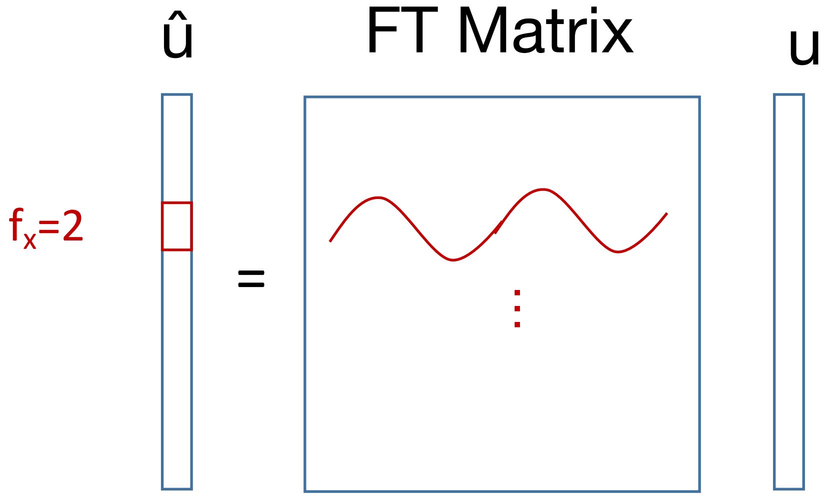
Inner product of u with different complex expon.

- np.fft(u), np.fftshift(np.fft(np.ifftshift(u)))
- fft = fast Fourier transform, much more comp. efficient than matrix multiplication!

Discrete Fourier Transforms

$$\hat{U}(f_x) = \int_{-\infty}^{\infty} U(x) \exp\left(-2\pi i (f_x x)\right) dx$$

$$\hat{u}[f_x] = \sum_{x=0}^{M-1} u[x] \exp\left(-2\pi i f_x x / M\right)$$



Inner product of u with different complex expon.

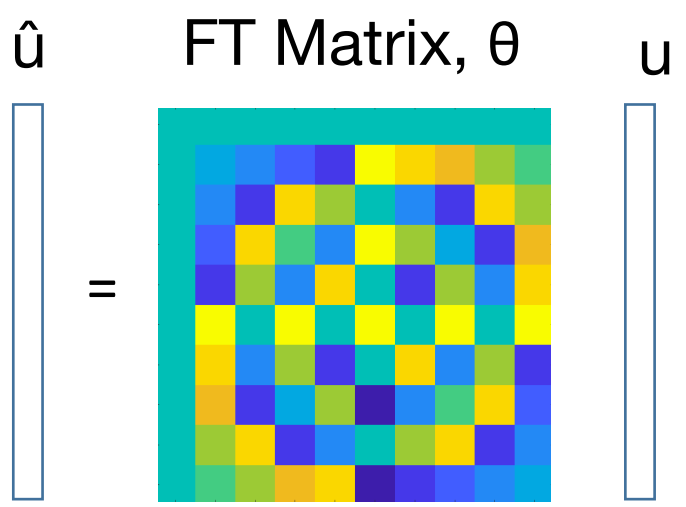
- np.fft(u), np.fftshift(np.fft(np.ifftshift(u)))
- fft = fast Fourier transform, much more comp. efficient than matrix multiplication!

Discrete Fourier Transforms

deep imaging

$$\hat{U}(f_x) = \int_{-\infty}^{\infty} U(x) \exp\left(-2\pi i (f_x x)\right) dx$$

$$\hat{u}[f_x] = \sum_{x=0}^{M-1} u[x] \exp\left(-2\pi i f_x x / M\right)$$



np.fft(np.eye(10))

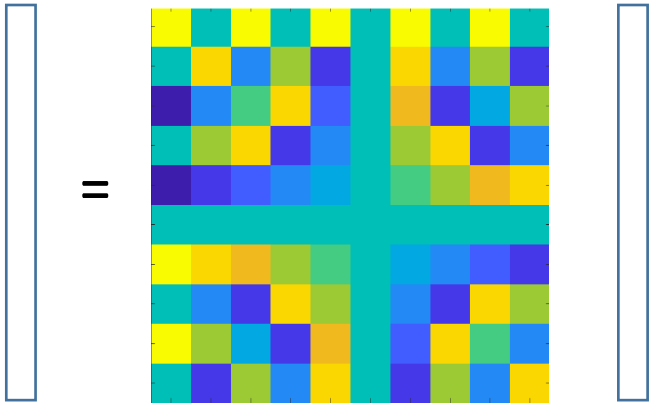
Treats 1st entry of û as f_x=0

Discrete Fourier Transforms

$$\hat{U}(f_x) = \int_{-\infty}^{\infty} U(x) \exp\left(-2\pi i (f_x x)\right) dx$$

$$\hat{u}[f_x] = \sum_{x=0}^{M-1} u[x] \exp\left(-2\pi i f_x x / M\right)$$

û FT Matrix, θ u



np.fftshift(np.fft(np.ifftshift(np.eye(10)))))

Treats middle entry of û as f_x=0

Discrete convolution theorem

Convolution theorem. If $\mathcal{F}\{g(x, y)) = G(f_X, f_Y)$ and $\mathcal{F}\{h(x, y)\} = H(f_X, f_Y)$, then

$$\mathcal{F}\left\{\iint_{-\infty}^{\infty} g(\xi, \eta)\, h(x - \xi, y - \eta)\, d\xi\, d\eta\right\} = G(f_X, f_Y)\, H(f_X, f_Y). \qquad (2\text{-}15)$$

deep imaging

Discrete convolution theorem

Convolution theorem. If $\mathcal{F}\{g(x, y)) = G(f_X, f_Y)$ and $\mathcal{F}\{h(x, y)\} = H(f_X, f_Y)$, then

$$\mathcal{F}\left\{\iint_{-\infty}^{\infty} g(\xi, \eta)\, h(x-\xi, y-\eta)\, d\xi\, d\eta\right\} = G(f_X, f_Y)\, H(f_X, f_Y). \qquad (2\text{-}15)$$

If $\mathcal{F}[g[x, y]] = G[f_x, f_y]$ and $\mathcal{F}[h[x, y]] = H[f_x, f_y]$, and if we know that

$$g[x, y] * h[x, y] = \sum_{l=-L}^{L} \sum_{m=-M}^{M} g[m, l] h[x-m, y-l],$$

then from the Convolution Theorem we have,

$$\mathcal{F}[g[x, y] * h[x, y]] = G[f_x, f_y] H[f_x, f_y]$$

Discrete convolution theorem example –same thing as continuous case

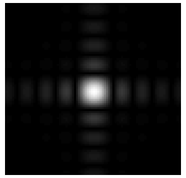
deep imaging

Input image

$U_1(x,y)$

Convolution filter h

$*$



$=$

Output image

$U_2(x,y)$

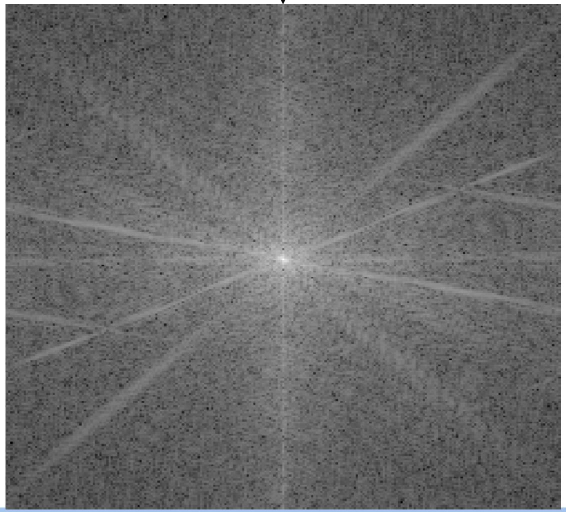
$F\,[U_1]$

$F\,[h]$

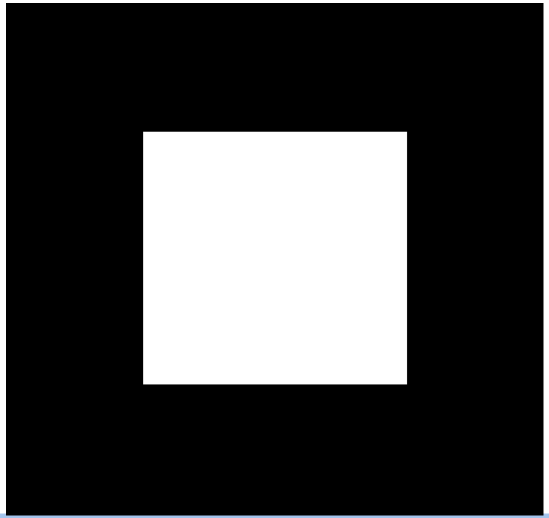
$F^{\,-1}[H\hat{U}_1]$

Input spectrum

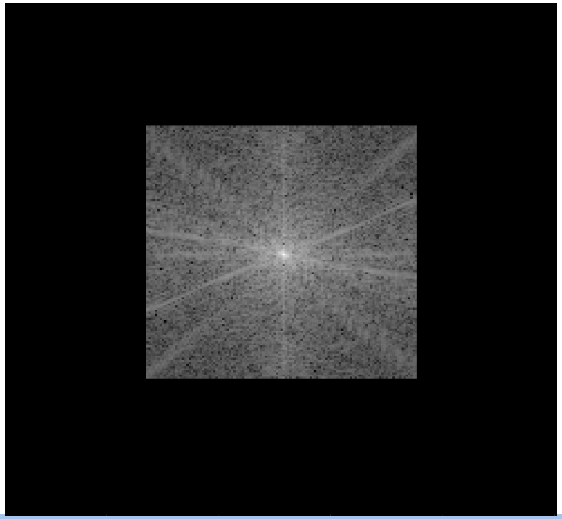
$\hat{U}_1(f_x,f_y)$



$\cdot$



$=$



Convolutions as a big matrix multiplication

$$(u * h)[x] = \sum_{m=0}^{N+M-2} u[m]h[x-m] \longrightarrow y = u * h = \begin{bmatrix} h_1 & 0 & \dots & 0 & 0 \\ h_2 & h_1 & \dots & \vdots & \vdots \\ h_3 & h_2 & \dots & 0 & 0 \\ \vdots & h_3 & \dots & h_1 & 0 \\ h_{m-1} & \vdots & \dots & h_2 & h_1 \\ h_m & h_{m-1} & \vdots & \vdots & h_2 \\ 0 & h_m & \dots & h_{m-2} & \vdots \\ 0 & 0 & \dots & h_{m-1} & h_{m-2} \\ \vdots & \vdots & \vdots & h_m & h_{m-1} \\ 0 & 0 & 0 & \dots & h_m \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_n \end{bmatrix}$$

Last thing – matrix and vector derivatives

deep imaging

$$\mathbf{u} = \mathbf{W}\mathbf{v}$$

$$\frac{d\mathbf{u}}{d\mathbf{v}} =$$

Last thing – matrix and vector derivatives

deep imaging

$$\mathbf{u} = \mathbf{W}\mathbf{v}$$

$$\frac{d\mathbf{u}}{d\mathbf{v}} =$$

$$\mathbf{u}_3 = W_{3,1}v_1 + W_{3,2}v_2 + \ldots + W_{3,M}v_M$$

$$\frac{\partial u_3}{\partial v_2} = \frac{\partial}{\partial v_2}(W_{3,1}v_1 + W_{3,2}v_2 + \ldots + W_{3,M}v_M) = \frac{\partial}{\partial v_2}W_{3,2}v_2 = W_{3,2}$$

Last thing – matrix and vector derivatives

$$\mathbf{u} = \mathbf{W}\mathbf{v}$$

$$\frac{d\mathbf{u}}{d\mathbf{v}} =$$

$$\mathbf{u}_3 = W_{3,1}v_1 + W_{3,2}v_2 + \ldots + W_{3,M}v_M$$

$$\frac{\partial u_3}{\partial v_2} = \frac{\partial}{\partial v_2}(W_{3,1}v_1 + W_{3,2}v_2 + \ldots + W_{3,M}v_M) = \frac{\partial}{\partial v_2}W_{3,2}v_2 = W_{3,2}$$

$$\frac{\partial u_i}{\partial v_j} = W_{i,j}$$

$$\frac{d\mathbf{u}}{d\mathbf{v}} = \mathbf{W}$$

Last thing – matrix and vector derivatives

$$\mathbf{u} = \mathbf{W}\mathbf{v}$$

$$\frac{d\mathbf{u}}{d\mathbf{v}} =$$

$$\mathbf{u}_3 = W_{3,1}v_1 + W_{3,2}v_2 + \ldots + W_{3,M}v_M$$

$$\frac{\partial u_3}{\partial v_2} = \frac{\partial}{\partial v_2}(W_{3,1}v_1 + W_{3,2}v_2 + \ldots + W_{3,M}v_M) = \frac{\partial}{\partial v_2}W_{3,2}v_2 = W_{3,2}$$

$$\frac{\partial u_i}{\partial v_j} = W_{i,j}$$

$$\frac{d\mathbf{u}}{d\mathbf{v}} = \mathbf{W}$$

- When confused, write out one entry, solve derivative and generalize
- Use dimensionality to help (if **x** has N elements, and **y** has M, then d**y**/d**x** must be NxM
- Take advantage of *The Matrix Cookbook*:
  - https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf