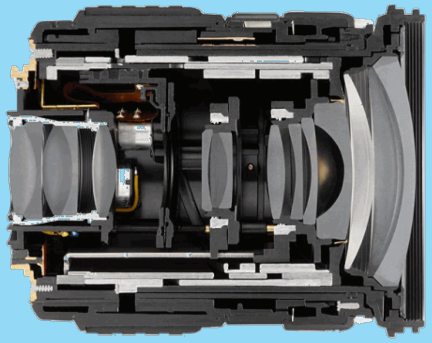
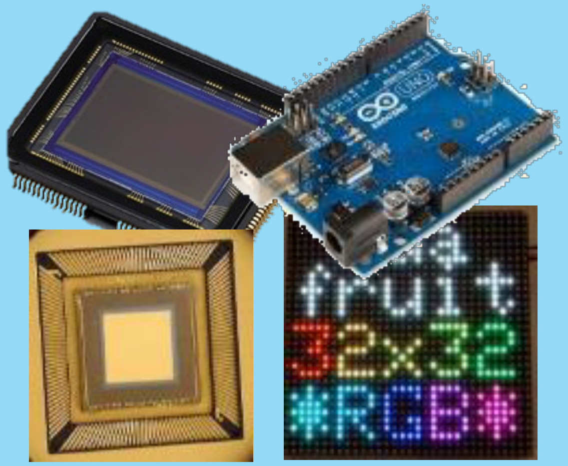


Optics

+



Sensors
&
devices

+

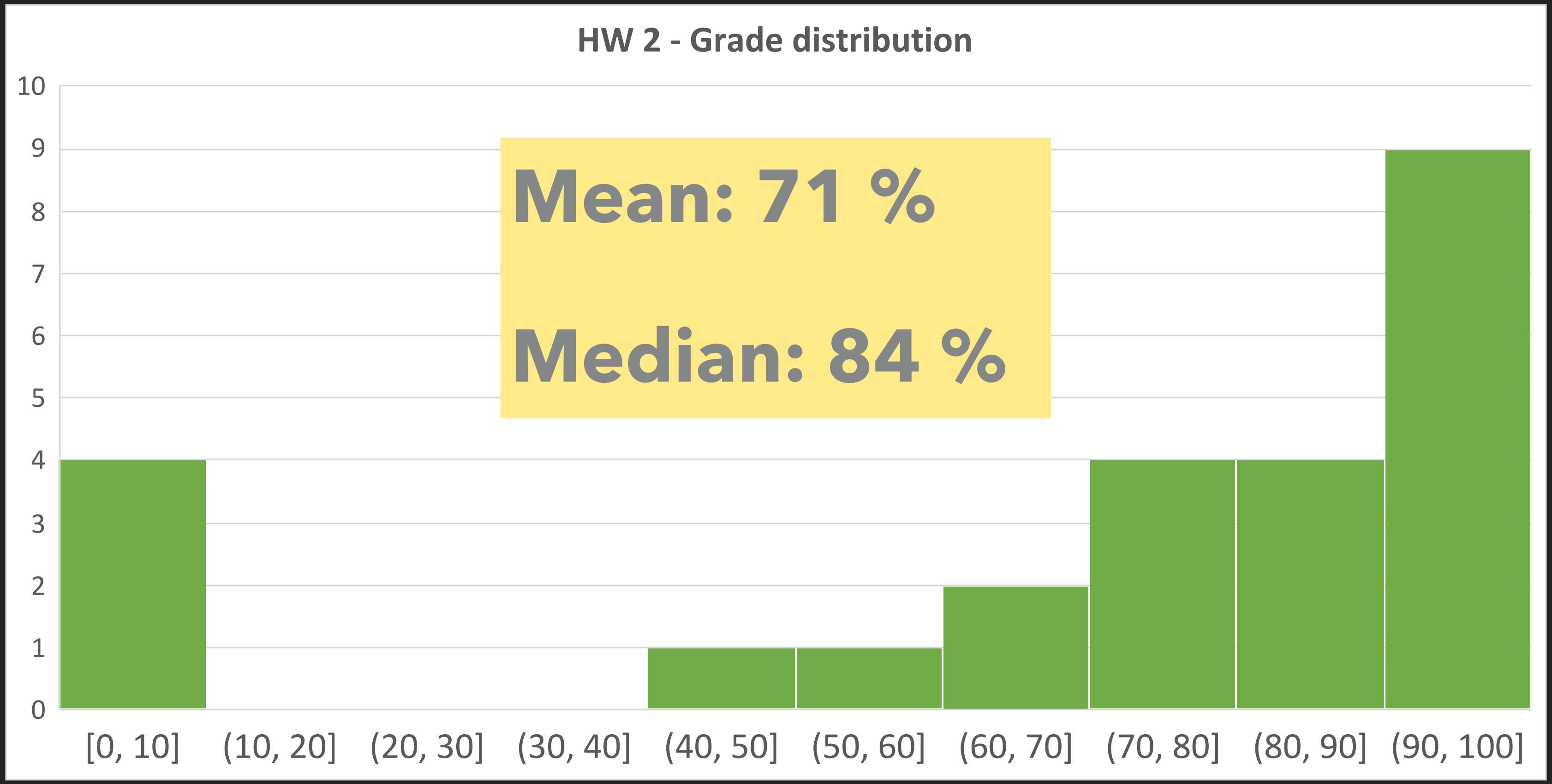
Signal
processing
&
algorithms

COMPUTATIONAL METHODS FOR IMAGING (AND VISION)

LECTURE 15: LSI IMAGING SYSTEMS (MATRIX-VECTOR FORMS)

PROF. JOHN MURRAY-BRUCE

ANNOUNCEMENTS



Homework 3 deadline extended
New due date: Sat, March 13, 2021
(11:59 PM EST)

Halfway Course/
Instruction Feedback
(See Canvas)

WHERE ARE WE

Week	Date	Main Topic	Lecture	Readings	Homework	
					Out	Due
1	11-Jan-21		Introduction to computational imaging - Forward and Inverse problems - Common computational imaging problems			
	13-Jan-21		Vectors - Preliminaries			
2	18-Jan-21		**Dr. Martin Luther King, Jr. Holiday (no class)**			
	20-Jan-21	Mathematical preliminaries	Vectors and Vector Spaces - Subspaces, Finite dimensional spaces	IIP Appendix A; FSP 2.1 - 2.2		
3	25-Jan-21		Vector Spaces - Hilbert spaces	IIP Appendix B; FSP 2.3		
	27-Jan-21		Bases and Frames I - Orthonormal and Reisz Bases	IIP Appendix C; FSP 2.4 and 2.B	**HW 1**	
4	1-Feb-21		Bases and Frames II - Orthogonal Bases - Linear operators	IIP Appendix C; FSP 2.5 and 2.B		
	3-Feb-21		Fourier Analysis I - FT (1D and 2D) - FT properties	IIP 2.1, Appendix D; FSP 4.4		
5	8-Feb-21		Sampling and Interpolation - BL functions - Sampling	IIP 2.2, 2.3; FSP 5.4, 5.5		**HW 1**
	10-Feb-21		Fourier Analysis II (DFT)	IIP 2.4; FSP 3.6	**HW 2**	
6	15-Feb-21	Forward Modeling	LSI imaging: Forward problem I - Convolution	IIP 2.5 - 2.6, 3		
	17-Feb-21		LSI imaging: Forward problem I - Transfer functions	IIP 2.6		
7	22-Feb-21		LSI imaging: Forward problem I - Linear operators	IIP 3		
	24-Feb-21		LSI imaging: Forward problem I - Linear operators, Adjoints, and Inverses		**HW 3**	**HW 2**
8	1-Mar-21		**Mid-term Exams**			
	3-Mar-21		LSI imaging: Forward problem II - Sampling and Discretization: Matrix-vector form	IIP 2.7, 4		
9	8-Mar-21		LSI imaging: Forward problem II - Convolution matrix			
	10-Mar-21		LSI imaging: Forward problem II - Sampling and Discretization: Matrix-vector form - PSF, and Transfer functions			**HW3**

WE ARE HERE!

OUTLINE

- Discrete LSI Imaging Systems
- Matrix-Vector models
  - Sampling, discretization and “transfer functions” of discrete LSI systems

LEARNING GOALS

- Understand simple discretization schemes for object and image spaces
- Discretize image space
- Describe effect of such discretization

READING

- IIP 2.5 - 2.7
- IIP 3.4

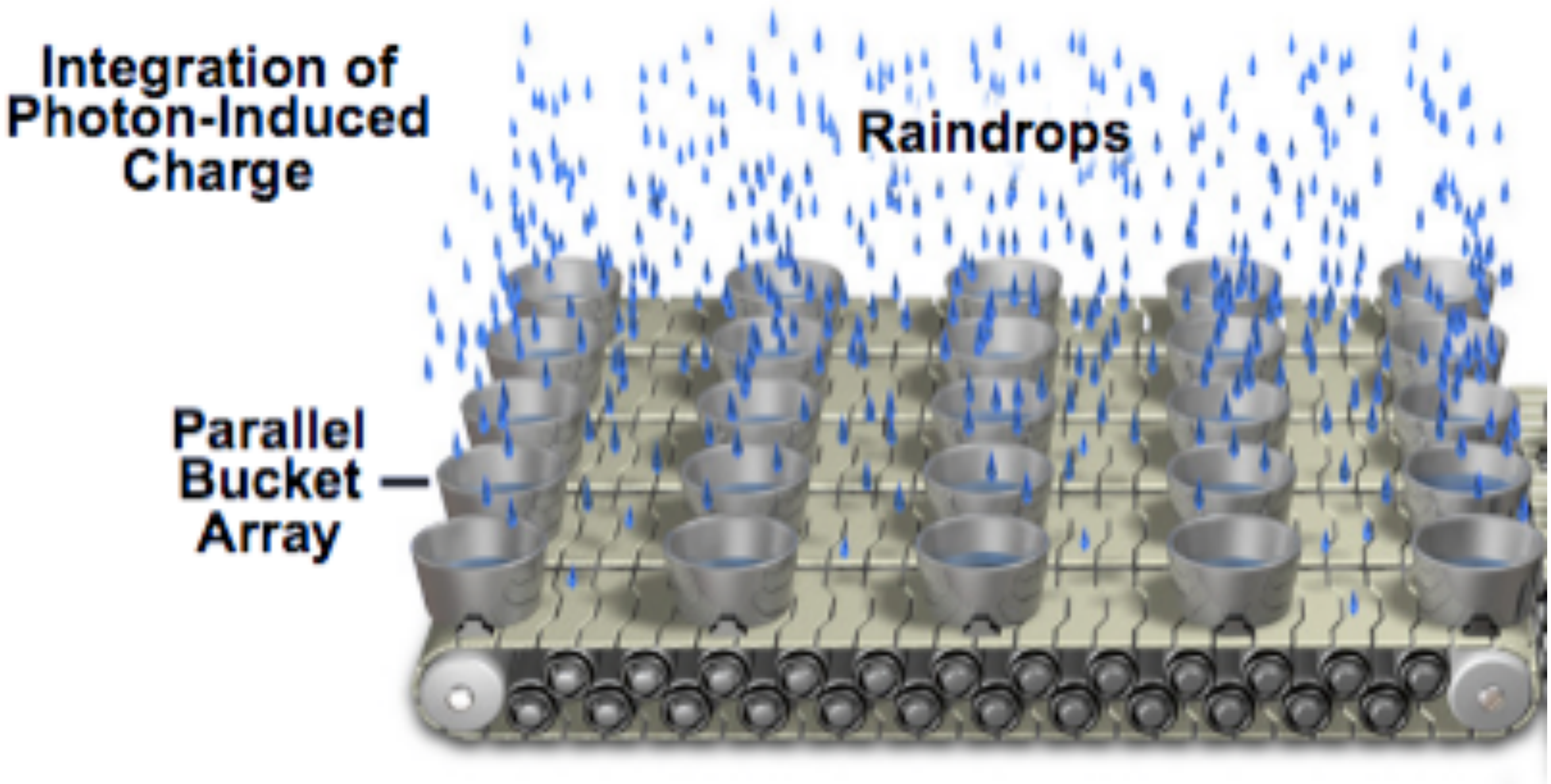
SAMPLING
DISCRETIZATION
TRANSFER FUNCTIONS

HOW DO IMAGING SENSORS WORK?

HOW DO IMAGING DETECTORS WORK?

THE PHYSICS

ARRAY OF DETECTOR ELEMENTS
INTUITION



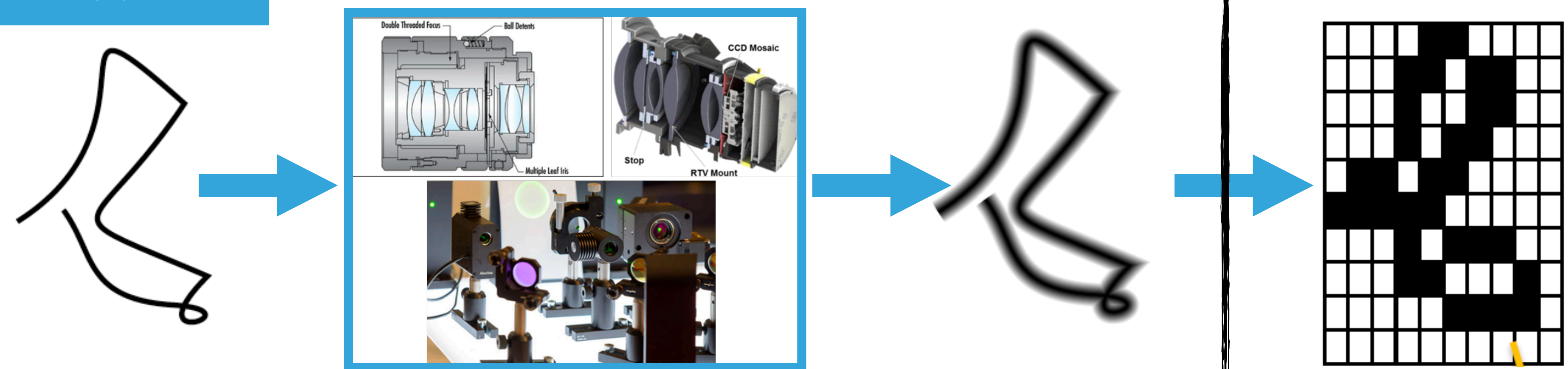
TURNING THE PHYSICS INTO MATH

THE MATHEMATICAL ABSTRACTION

SAMPLING

HOW DO IMAGING SENSORS TYPICALLY WORK?

ANALOG PART



$g(x, y)$ Optics $h(x, y)$ $g(x, y) \star h(x, y)$ Sampling $\{g_{mn}\}$

Where $g(x, y) \star h(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x', y') h(x - x', y - y') \, \mathrm{d}y' \mathrm{d}x'$

$$g_{mn} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\frac{\Delta_x}{2}}^{\frac{\Delta_x}{2}} \int_{-\frac{\Delta_y}{2}}^{\frac{\Delta_y}{2}} g(x, y) h(x' - x, y' - y) \boxed{p(x' - m\Delta_x, y' - n\Delta_y)} \, \mathrm{d}y' \mathrm{d}x' \mathrm{d}y \mathrm{d}x$$

$p(x, y)$ is the pixel sampling function

How can we describe square pixels? **2D RECTANGULAR/BOX-CAR FUNCTIONS**

EFFECT OF PIXEL SAMPLING

PIXEL TRANSFER FUNCTION

- The sample at each pixel is given by

$$g_{mn} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\Delta_y/2}^{\Delta_y/2}\int_{-\Delta_x/2}^{\Delta_x/2} g(x',y')h'(x-x',y-y')\,\underbrace{\boxed{p(x-m\Delta_x, y-n\Delta_y)}}_{\text{PIXEL SAMPLING FUNCTION}}\,\mathrm{d}x\mathrm{d}y\mathrm{d}x'\mathrm{d}y'$$

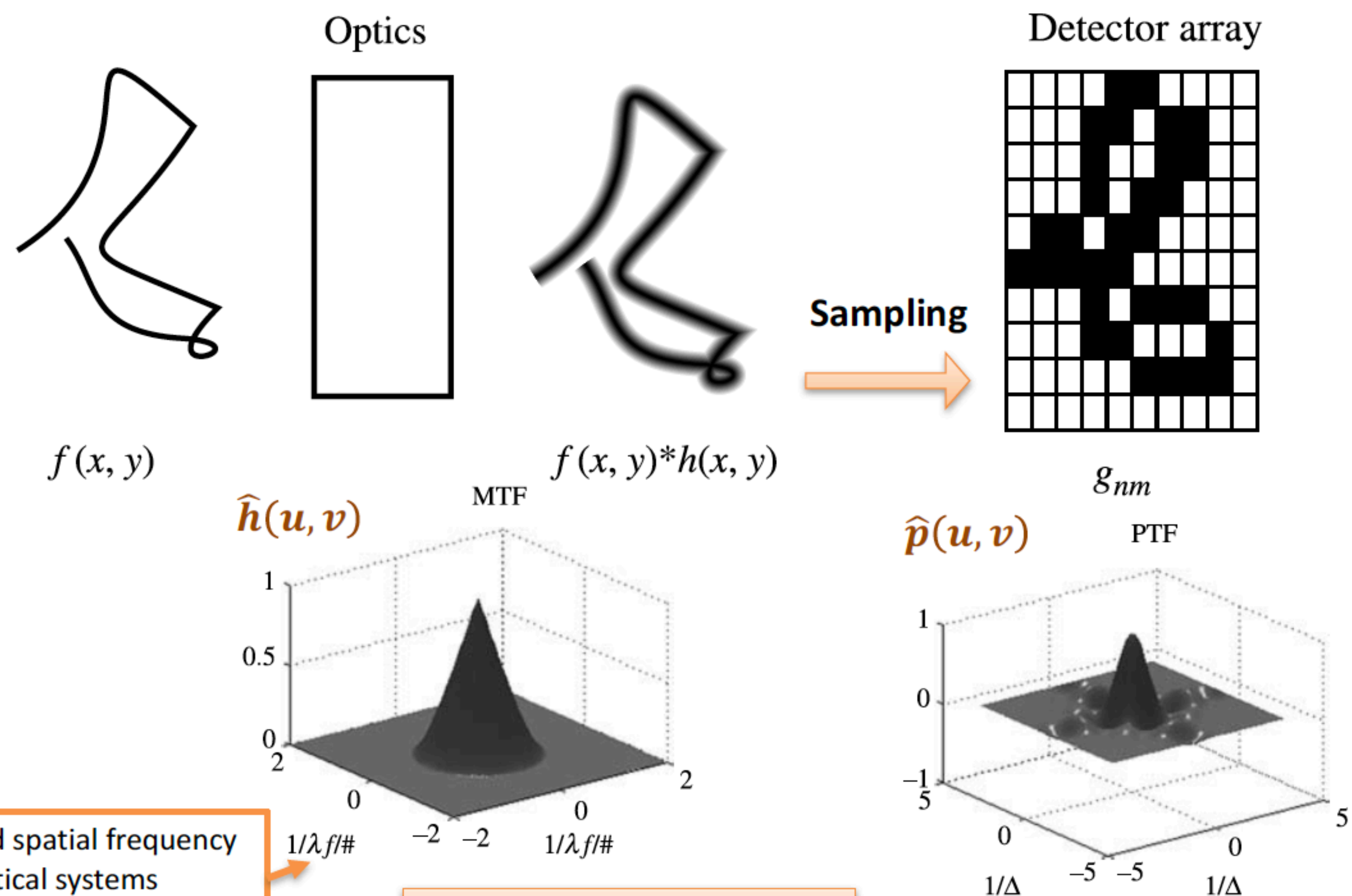
- First, take the **Fourier transform** $\mathscr{F}$ of the continuous version:

$$G_{\text{out}}(\omega_x,\omega_y) = G(\omega_x,\omega_y)H(\omega_x,\omega_y)\,\underbrace{\boxed{P(\omega_x,\omega_y)}}_{\text{PIXEL TRANSFER FUNCTION}}$$

- Take inverse **Fourier transform** $\mathscr{F}^{-1}$ and evaluate at $(m\Delta_x, n\Delta_y)$

$$g_{mn} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G(\omega_x,\omega_y)H(\omega_x,\omega_y)P(\omega_x,\omega_y)e^{\mathrm{j}(m\Delta_x\omega_x+n\Delta_y\omega_y)}\,\mathrm{d}\omega_x\mathrm{d}\omega_y$$

PIXEL TRANSFER FUNCTION
EXAMPLE



Normalized spatial frequency unit for optical systems
System Bandwidth = $1/\lambda f/\#$

Set the bandwidth of the signal before sampling

D. Brady, Optical Imaging and Spectroscopy, Chap 7.2.

SAMPLING
GENERAL DESCRIPTION

- Taking discrete samples from a continuous function $g(x)$ to produce a vector

$$\mathbf{g} = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{M-1} \end{bmatrix}$$

- Let the m-th pixel have the sampling function $p_m(x)$
  - Note that we have gone from a 2D indexing to 1D index, this can be achieved by a simple **lexicographic ordering**
- Then $g_m = \int g(x)p_m(x)\mathrm{d}x = \langle g(x), p_m(x) \rangle$
- Thus, $\mathbf{g} = \left[\langle g(x), p_0(x) \rangle, \langle g(x), p_1(x) \rangle, \cdots, \langle g(x), p_{M-1}(x) \rangle \right]^T$

VECTORIZATION
LEXICOGRAPHIC ORDERING

MATRIX

(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(1, 6)
(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)	(2, 6)
(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)	(3, 6)
(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)	(4, 6)
(5, 1)	(5, 2)	(5, 3)	(5, 4)	(5, 5)	(5, 6)

1	6	11	16	21	26
2	7	12	17	22	27
3	8	13	18	23	28
4	9	14	19	24	29
5	10	15	20	25	30

VECTOR

(1, 1)
(2, 1)
(3, 1)
(4, 1)
(5, 1)
(1, 2)
(2, 6)
(3, 6)
(4, 6)
(5, 6)
(1, 3)
(2, 3)
(3, 3)
(4, 3)
(5, 3)
(1, 4)
(2, 4)
(3, 4)
(4, 4)
(5, 4)
(1, 5)
(2, 5)
(3, 5)
(4, 5)
(5, 5)
(1, 6)
(2, 6)
(3, 6)
(4, 6)
(5, 6)

SAMPLING
DISCRETIZATION
TRANSFER FUNCTIONS

REPRESENTATIONS FOR THE OBJECT AND IMAGING SYSTEM

DISCRETIZATION

DISCRETIZING THE OBJECT FUNCTION

- The goal is to represent the continuous function $f(x)$ by a vector

$$\mathbf{f} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \end{bmatrix}$$

- Assume a set of basis functions $\{\psi_0(x), \psi_1(x), \ldots, \psi_{N-1}(x)\}$
- Then, the continuous function:

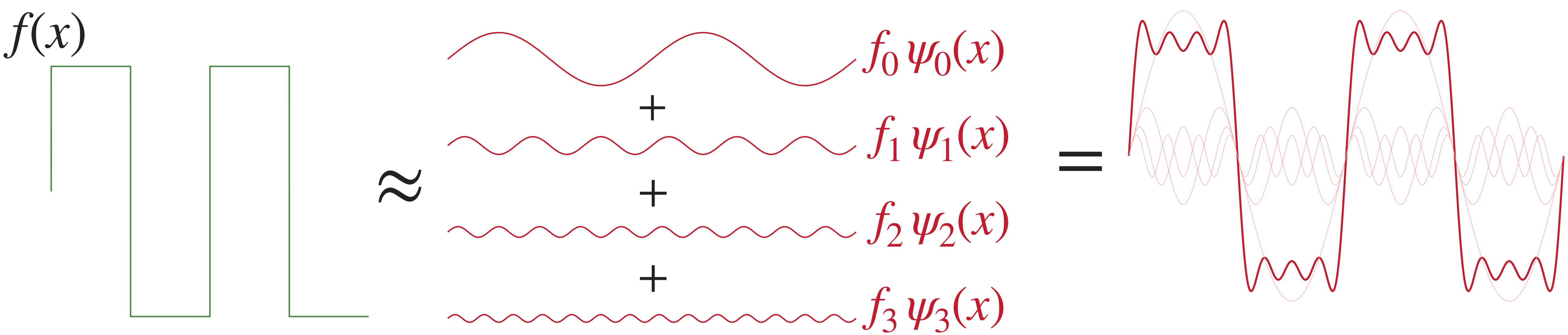
$$f(x) = \sum_{n=0}^{N-1} f_n \psi_n(x)$$

DISCRETIZATION OF OBJECT FUNCTION

- The goal is to represent the continuous function $f(x)$ using a vector $\mathbf{f}$
  - Using some basis functions $\{\psi_0(x), \psi_1(x), \ldots, \psi_{N-1}(x)\}$
  - Then, the continuous function can be reconstructed as:

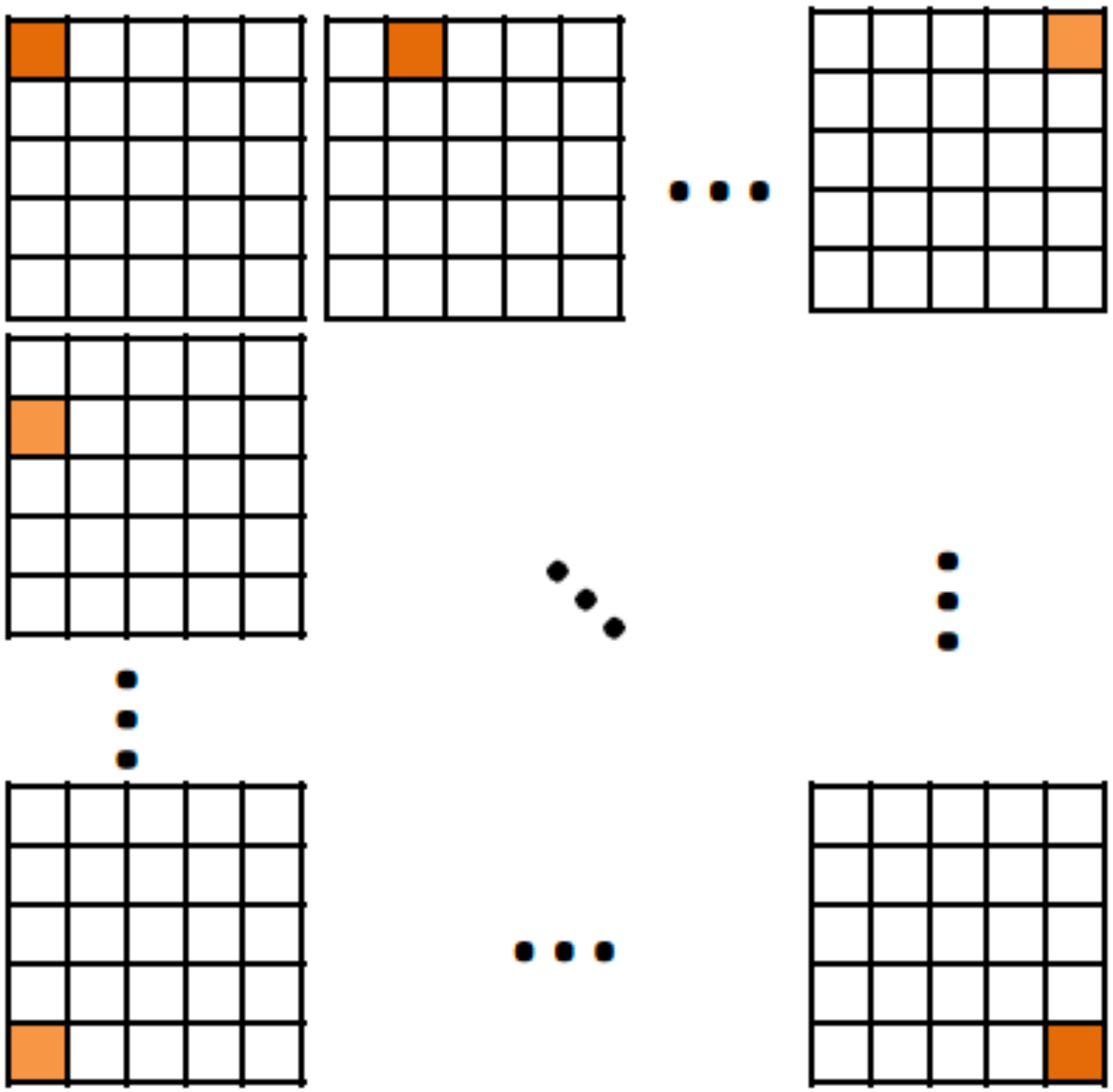
$$f(x) = \sum_{n=0}^{N-1} f_n \psi_n(x)$$

- We have actually seen a previous **example**
  - $f_0 = 1, f_1 = \frac{1}{3}, f_2 = \frac{1}{5}, f_3 = \frac{1}{7}$
  - $\psi_0(x) = \sin(x),\ \psi_1(x) = \sin(3x),\ \psi_2(x) = \sin(5x),\ \psi_3(x) = \sin(7x)$
- **Why is this useful?**



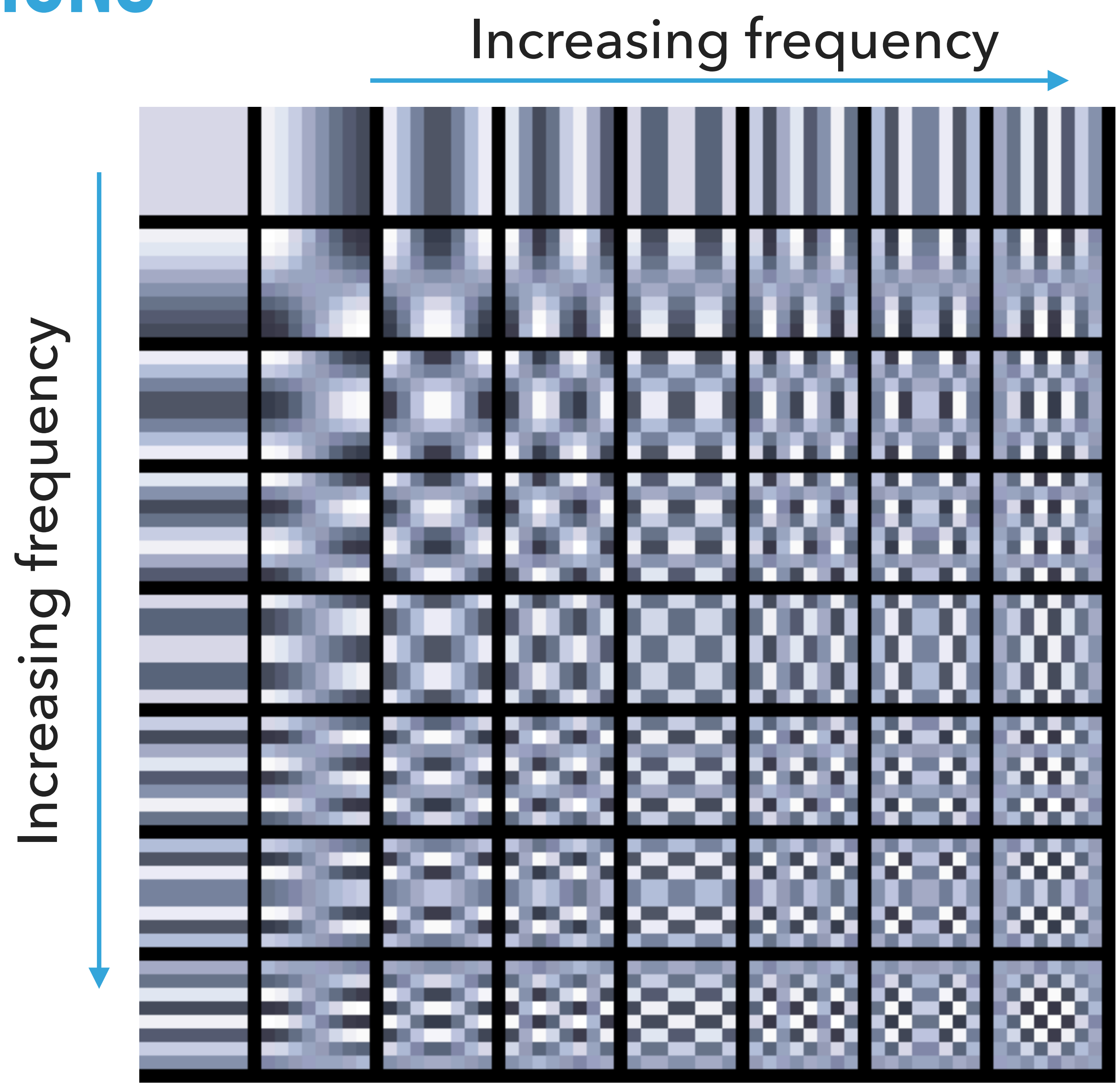
ALSO POSSIBLE IN 2D OR 3D AND SO ON

BASIS FUNCTIONS
THE STANDARD/CANONICAL BASIS



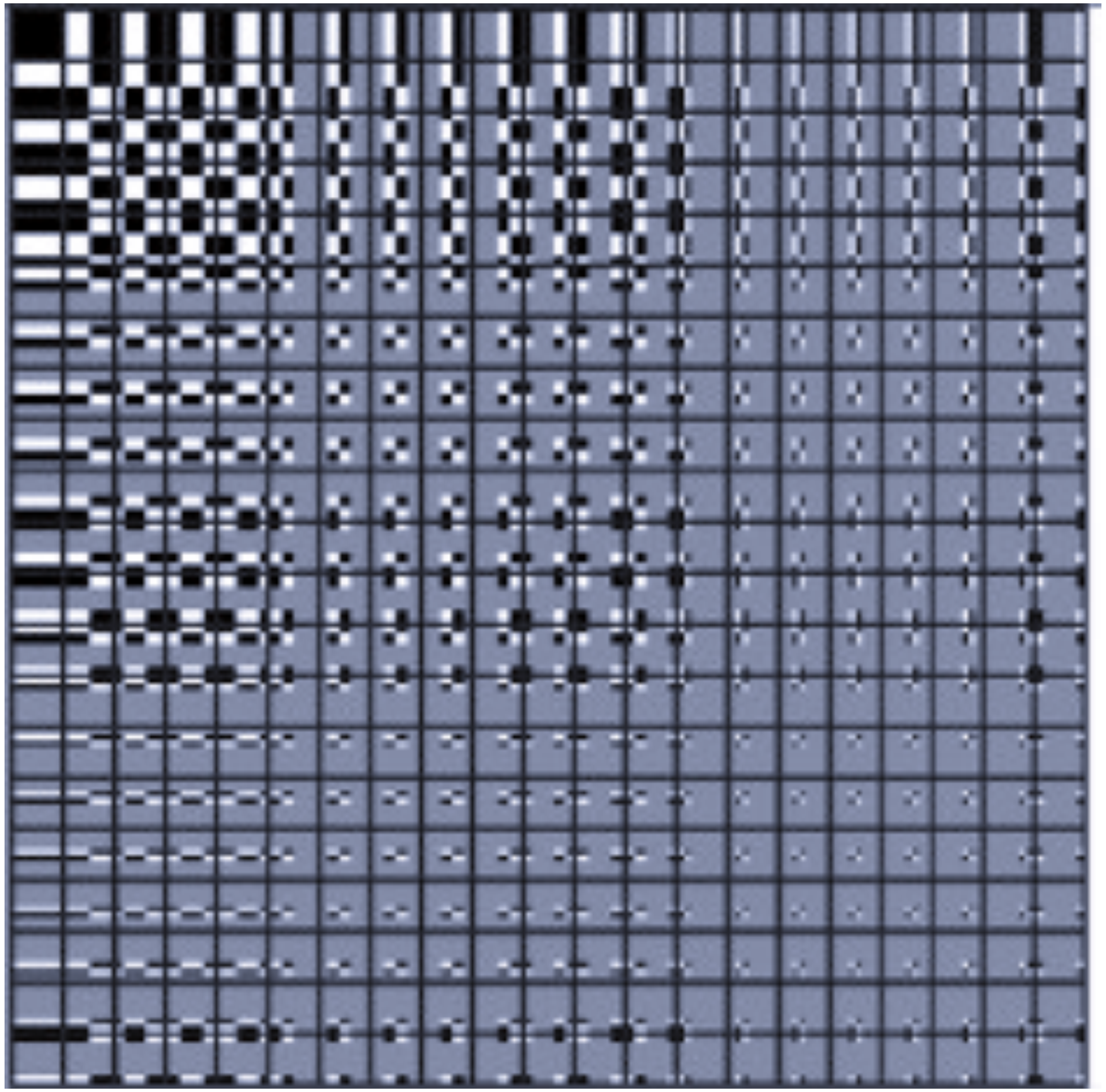
CAN THINK OF THIS AS THE SIMPLEST (OR THE "OBVIOUS") BASIS SET

BASIS FUNCTIONS
FOURIER BASIS



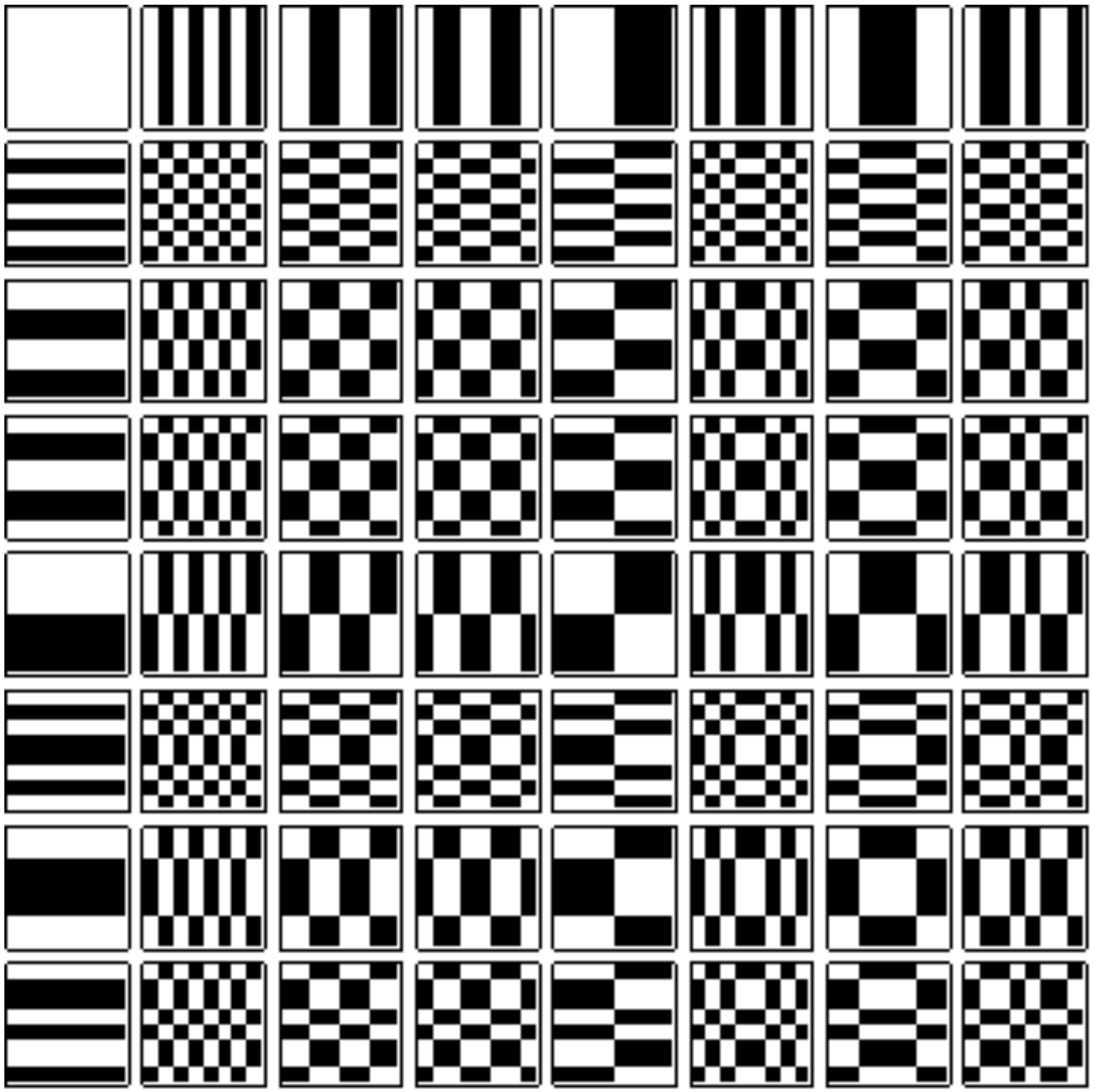
CAN THINK OF EACH BLOCK (WITHIN BLACK BOARDERS) AS SINUSOIDAL WAVES IN 2 DIMENSIONS

BASIS FUNCTIONS
HAAR (WAVELET) BASIS



BASIS FUNCTIONS

HADAMARD BASIS



OBJECT SPACE DISCRETIZATION

- After choosing an appropriate basis set $\left\{\psi_n(x)\right\}_{n=0}^{N-1}$
- The continuous function is $f(x)$
- The discrete version is the vector

$$\mathbf{f} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \end{bmatrix}$$

- Such that, given the vector $\mathbf{f}$, the continuous version function may be recovered as follows: $f(x) = \sum_{n=0}^{N-1} f_n \psi_n(x)$

FULLY DISCRETE LINEAR FORWARD MODEL

- The m-th sample

$$
\begin{aligned}
g_m &= \langle f(x), \phi_m(x) \rangle \\
&= \left\langle \sum_{n=0}^{N-1} f_n \psi_n(x), \phi_m(x) \right\rangle \\
&= \sum_{n=0}^{N-1} \langle \psi_n(x), \phi_m(x) \rangle f_n = \sum_{n=0}^{N-1} \mathbf{A}_{mn} f_n
\end{aligned}
$$

- Thus, $\mathbf{A}_{mn} \in \mathbb{R}^{M \times N}$ is such that

$$
\mathbf{A}_{mn} = \langle \psi_n(x), \phi_m(x) \rangle
$$

- $\mathbf{A}_{mn} = \langle \psi_n(x), \phi_m(x) \rangle = \langle \psi_n(x), A^* p_m(x) \rangle = \langle A\psi_n(x), p_m(x) \rangle$

FULLY DISCRETE LINEAR FORWARD MODEL

EXAMPLE

- Assuming ideal impulse sampling

$$p_m(x) = \delta(x - x_m)$$

- Standard basis as the object function

$$\psi_n(x) = \mathrm{rect}\left(\frac{x - x_n}{\Delta_x}\right) \approx \Delta_x \delta(x - x_n), \text{ as } \Delta_x \to 0$$

- Shift-invariant system model $h(x)$

- What is $\mathbf{A}$?

$$\boxed{\mathbf{A}_{mn} = \Delta_x h(x_m - x_n)}$$

WHY?

FULLY DISCRETE LINEAR FORWARD MODEL

EXAMPLE – DERIVING MATRIX $\mathbf{A}_{mn}$

- The matrix $\mathbf{A}$ has entries: $\mathbf{A}_{mn} = \langle A\psi_n(x), p_m(x)\rangle$
  - Impulse sampling: $p_m(x) = \delta(x - x_m)$, and
  - Standard basis for object space: $\psi_n(x) \approx \Delta_x \delta(x - x_n)$
- Operator A means that:

$$A\psi_n(x) = \int_{-\infty}^{\infty} \psi_n(x')h(x - x')\,\mathrm{d}x' = \Delta_x \int_{-\infty}^{\infty} \delta(x - x_n)h(x - x')\,\mathrm{d}x'$$
$$= \Delta_x h(x - x_n)$$

- Then $\mathbf{A}_{mn} = \langle \Delta_x h(x - x_n), p_m(x)\rangle = \int_{-\infty}^{\infty} \Delta_x h(x - x_n)\, p_m(x)\,\mathrm{d}x$

$$= \int_{-\infty}^{\infty} \Delta_x h(x - x_n)\,\delta(x - x_m)\,\mathrm{d}x$$
$$= \Delta_x h(x_m - x_n)$$

DISCRETE LINEAR SHIFT-INVARIANT SYSTEM

MATRIX-VECTOR FORMS

$$\boxed{\mathbf{A}_{mn} = \Delta_x h(x_m - x_n)}$$

- The matrix element (m, n) depends only on the difference $(x_m - x_n)$
  - Thus, we have a **discrete LSI system**
- Where, we recall that, the ***m*-th sample** is:

$$g_m = \sum_{n=0}^{N-1} \mathbf{A}_{mn} f_n$$

We can now specify the exact matrix-vector form!

DISCRETE LINEAR SHIFT-INVARIANT SYSTEM

MATRIX-VECTOR FORMS

Matrix-Vector form: simply rewrite the summation $g_m = \sum_{n=0}^{N-1} \mathbf{A}_{mn} f_n$, as the matrix-vector product

$$\mathbf{g} = \mathbf{A}\mathbf{f}$$

where $\mathbf{g} = \begin{bmatrix} g_0 \\ \vdots \\ g_{M-1} \end{bmatrix}$, $\quad \mathbf{f} = \begin{bmatrix} f_0 \\ \vdots \\ f_{N-1} \end{bmatrix}$,

(m, n) entry of matrix $\mathbf{A}$ is given by $\mathbf{A}_{mn} = \Delta_x h(x_m - x_n)$

DISCRETE LINEAR SHIFT-INVARIANT SYSTEM

- Have an LSI system, with **sampling set** $\{p_m(x)\}_m$, **basis set** $\{\psi_n\}_n$, if **imaging system PSF** is LSI

- Notice also that: "imaging/sensing/system matrix" $\mathbf{A}$ satisfies

$$\mathbf{g}_m = \sum_{n=0}^{N-1} \mathbf{A}_{mn}\,\mathbf{f}_n = \sum_{n=0}^{N-1} \mathbf{a}_{m-n}\,\mathbf{f}_n$$

  - Where $\mathbf{a} \in \mathbb{R}^M$ (a vector)

  - Matrix $\mathbf{A}$ is ***Toeplitz*** with the (m, n) entry given by $\mathbf{A}_{mn} = \mathbf{a}_{m-n}$

AN EXAMPLE

AN EXAMPLE MATRIX

- Consider the LSI system with PSF $h(x) = e^{-x^2}$,
- Using **Dirac delta sampling and Dirac object space discretization**

$$\mathbf{A}_{mn} = \Delta\, e^{-(x_m - x_n)^2}$$

  - With uniform sampling Δ: $x_m = m\Delta$ (i.e. pixels uniformly spaced)
  - With uniform spacing δ for basis: $x_n = n\delta$
  - Thus, $\mathbf{A}_{mn} = e^{-(m\Delta - n\delta)^2}$
- Thus $\mathbf{A}_{mn} = e^{-(m-n)^2\Delta^2}$, if $\Delta = \delta$ for example
  - The element $\mathbf{A}_{mn} = \Delta\, e^{-(m-n)^2\Delta^2}$ depends the difference $(m-n)$ not their exact values

AN EXAMPLE MATRIX

$$\mathbf{A}_{m,n} = e^{-(m-n)^2\Delta^2}$$

$$M = 5, N = 3$$

$$m = 0, n = 0$$

$$\mathbf{A}_{0,0} = e^{-(0-0)^2\Delta^2}$$

$$\mathbf{A}_{0,0} = e^{-0} = 1$$

$$\mathbf{A} = \begin{bmatrix} \boxed{1} & \mathbf{A}_{0,1} & \mathbf{A}_{0,2} \\ \mathbf{A}_{1,0} & \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,0} & \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \\ \mathbf{A}_{3,0} & \mathbf{A}_{3,1} & \mathbf{A}_{3,2} \\ \mathbf{A}_{4,0} & \mathbf{A}_{4,1} & \mathbf{A}_{4,2} \end{bmatrix}$$

AN EXAMPLE MATRIX

$$\mathbf{A}_{m,n} = e^{-(m-n)^2\Delta^2}$$

$$M = 5, N = 3$$

$$m = 0, n = 1$$

$$\mathbf{A}_{0,1} = e^{-(0-1)^2\Delta^2}$$

$$\mathbf{A}_{0,1} = e^{-\Delta^2}$$

$$\mathbf{A} = \begin{bmatrix} 1 & \boxed{e^{-\Delta^2}} & \mathbf{A}_{0,2} \\ \mathbf{A}_{1,0} & \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,0} & \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \\ \mathbf{A}_{3,0} & \mathbf{A}_{3,1} & \mathbf{A}_{3,2} \\ \mathbf{A}_{4,0} & \mathbf{A}_{4,1} & \mathbf{A}_{4,2} \end{bmatrix}$$

AN EXAMPLE MATRIX

$$\mathbf{A}_{m,n} = e^{-(m-n)^2\Delta^2}$$

$$M = 5, N = 3$$

$$m = 0, n = 2$$

$$\mathbf{A}_{0,2} = e^{-(0-2)^2\Delta^2}$$

$$\mathbf{A}_{0,2} = e^{-4\Delta^2}$$

$$\mathbf{A} = \begin{bmatrix} 1 & e^{-\Delta^2} & \boxed{e^{-4\Delta^2}} \\ \mathbf{A}_{1,0} & \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,0} & \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \\ \mathbf{A}_{3,0} & \mathbf{A}_{3,1} & \mathbf{A}_{3,2} \\ \mathbf{A}_{4,0} & \mathbf{A}_{4,1} & \mathbf{A}_{4,2} \end{bmatrix}$$

AN EXAMPLE MATRIX

$$\mathbf{A}_{m,n} = e^{-(m-n)^2\Delta^2}$$

$$M = 5, N = 3$$

$$m = 1, n = 0$$

$$\mathbf{A}_{1,0} = e^{-(1-0)^2\Delta^2}$$

$$\mathbf{A}_{1,0} = e^{-\Delta^2}$$

$$\mathbf{A} = \begin{bmatrix} 1 & e^{-\Delta^2} & e^{-4\Delta^2} \\ \boxed{e^{-\Delta^2}} & \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,0} & \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \\ \mathbf{A}_{3,0} & \mathbf{A}_{3,1} & \mathbf{A}_{3,2} \\ \mathbf{A}_{4,0} & \mathbf{A}_{4,1} & \mathbf{A}_{4,2} \end{bmatrix}$$

AN EXAMPLE MATRIX

$$\mathbf{A}_{m,n} = e^{-(m-n)^2\Delta^2}$$

$$M = 5,\ N = 3$$

$$m = 1, n = 1$$

$$\mathbf{A}_{1,1} = e^{-(1-1)^2\Delta^2}$$

$$\mathbf{A}_{1,1} = e^{-0} = 1$$

$$\mathbf{A} = \begin{bmatrix} 1 & e^{-\Delta^2} & e^{-4\Delta^2} \\ e^{-\Delta^2} & \boxed{1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,0} & \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \\ \mathbf{A}_{3,0} & \mathbf{A}_{3,1} & \mathbf{A}_{3,2} \\ \mathbf{A}_{4,0} & \mathbf{A}_{4,1} & \mathbf{A}_{4,2} \end{bmatrix}$$

AN EXAMPLE MATRIX

$$\mathbf{A}_{m,n} = e^{-(m-n)^2\Delta^2}$$

$$M = 5, N = 3$$

$$m = 1, n = 2$$

$$\mathbf{A}_{1,2} = e^{-(1-2)^2\Delta^2}$$

$$\mathbf{A}_{1,2} = e^{-\Delta^2}$$

$$\mathbf{A} = \begin{bmatrix} 1 & e^{-\Delta^2} & e^{-4\Delta^2} \\ e^{-\Delta^2} & 1 & \boxed{e^{-\Delta^2}} \\ \mathbf{A}_{2,0} & \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \\ \mathbf{A}_{3,0} & \mathbf{A}_{3,1} & \mathbf{A}_{3,2} \\ \mathbf{A}_{4,0} & \mathbf{A}_{4,1} & \mathbf{A}_{4,2} \end{bmatrix}$$

AN EXAMPLE MATRIX

$$\mathbf{A}_{m,n} = e^{-(m-n)^2\Delta^2}$$

$$M = 5, N = 3$$

$$m = 4, n = 2$$

$$\mathbf{A}_{4,2} = e^{-(4-2)^2\Delta^2}$$

$$\mathbf{A}_{4,2} = e^{-4\Delta^2}$$

$$\mathbf{A} = \begin{bmatrix} 1 & e^{-\Delta^2} & e^{-4\Delta^2} \\ e^{-\Delta^2} & 1 & e^{-\Delta^2} \\ e^{-4\Delta^2} & e^{-\Delta^2} & 1 \\ e^{-9\Delta^2} & e^{-4\Delta^2} & e^{-\Delta^2} \\ e^{-16\Delta^2} & e^{-9\Delta^2} & \boxed{\overset{\mathbf{A}_{4,2}}{e^{-4\Delta^2}}} \end{bmatrix}$$

AN EXAMPLE MATRIX

$$\mathbf{A}_{m,n} = e^{-(m-n)^2\Delta^2}$$

$$M = 5, N = 3$$

Toeplitz matrix: diagonal terms are equal

$$\mathbf{A} = \begin{bmatrix} 1 & e^{-\Delta^2} & e^{-4\Delta^2} \\ e^{-\Delta^2} & 1 & e^{-\Delta^2} \\ e^{-4\Delta^2} & e^{-\Delta^2} & 1 \\ e^{-9\Delta^2} & e^{-4\Delta^2} & e^{-\Delta^2} \\ e^{-16\Delta^2} & e^{-9\Delta^2} & e^{-4\Delta^2} \end{bmatrix}$$

WHAT WE COVERED TODAY

- Discretizing LSI imaging systems
  - Imaging systems represented as a linear system of equations (Matrix-Vector equations)
    - Sampling (imaging sensor/detector)
    - Discretization of object space and imaging system
    - Discrete transfer functions