Computer Aided Medical Procedures
Department of Computer Science
Technical University of Munich

Image Restoration: Part 1

Seminar: Computational Methods for Image Reconstruction

Berfin Kavşut

Computer Aided Medical Procedures
Department of Computer Science
Technical University of Munich

January 10th, 2024

Overview

- **Image Restoration vs. Image Reconstruction**
- **Discrete Measurement Model**
  - Linear-Shift Invariant Systems & Convolution
- **Continuous-Discrete Model**
- **Matrix-Vector Representations**
  - Fourier Transform of Non-Periodic Signals
  - 1D Matrix-Vector Representation
  - 2D Matrix-Vector Representation
- **Circulant Analysis**
  - Fourier Transform of Periodic Signals
  - Circulant Analysis (1D)
- **Simple Image Restoration Problems**
  - Deconvolution Solution
  - Matrix Inverse Solution

1. Image Restoration vs. Image Reconstruction

Image Reconstruction: The aim is to form an image from measured data that is not interpretable directly as an image, such as a sinogram in tomography.

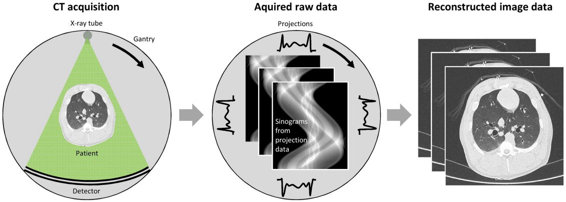


Figure 1 Image reconstruction [1]

1. Image Restoration vs. Image Reconstruction

Image Restoration: The aim is to recover the underlying **true object** from the measured image, which is blurry and noisy, in 2D image restoration problems.

Figure 2 Image restoration

2. Discrete Measurement Model

$$f \longrightarrow \boxed{\text{h}} \longrightarrow g$$

- **System model**: relating the unknown quantities, i.e. input, to the observed measurements, i.e. output.

- This is called as **forward model** in the inverse problems field.

- **Latent image:** $f[m, n] \in \mathbb{R} : m = 0, \dots, M-1,\ n = 0, \dots, N-1$

- **Measurement:** $g[m, n] \in \mathbb{R} : m = 0, \dots, M-1,\ n = 0, \dots, N-1$

2. Discrete Measurement Model

- **Impulse function:**

  Kronecker impulse function: $\delta_{ij} = \begin{cases} 1 & \text{if } i = 0 \text{ and } j = 0 \\ 0 & \text{otherwise} \end{cases}$

  Impulse response: Describes how the system responds to an impulse function. In a non-shift invariant system, the response may vary with the position of the impulse.

- **Linear combination of shifted impulses**:

$$f[m,n] = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} f[k,l]\delta[m-k, n-l]$$

  Linear combination of impulse responses:

$$g[m,n] = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} f[k,l]b[m,n;k,l]$$

2. Discrete Measurement Model

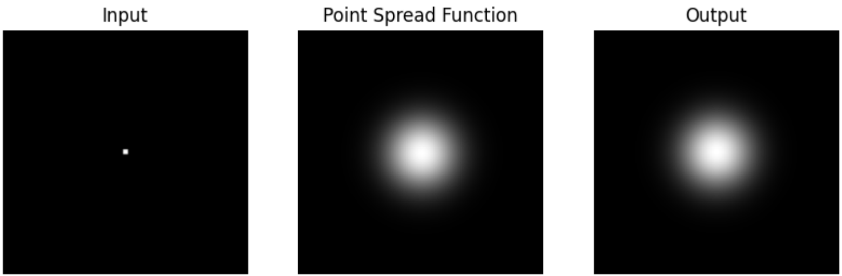


Figure 3 Impulse Response

2.1. Linear-Shift Invariant Systems

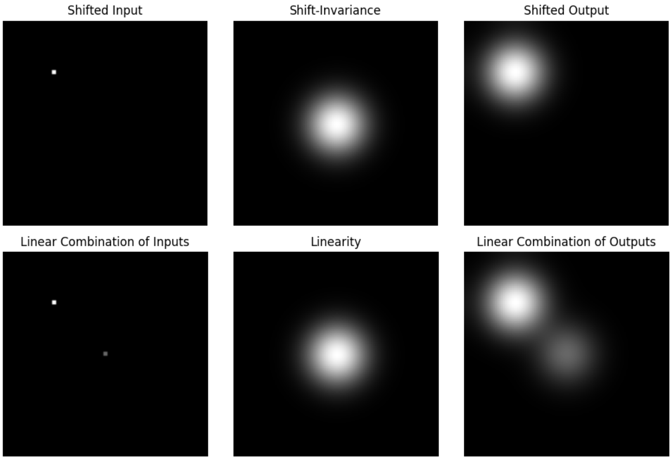


Figure 4 LSI systems

2.1. Linear-Shift Invariant Systems

$$x[n] \xrightarrow{H} y[n]$$

Linearity:

1. Additivity

$x_1[n] \xrightarrow{H} y_1[n]$ and $x_2[n] \xrightarrow{H} y_2[n]$

$x_1[n] + x_2[n] \xrightarrow{H} y_1[n] + y_2[n]$

2. Homogeneity or Scaling

$\alpha x[n] \xrightarrow{H} \alpha y[n]$

Shift-Invariance:

$x[n - n_0] \xrightarrow{H} y[n - n_0]$

2.2. (Linear) Convolution

- **Linear combination of shifted impulses**:

$$f[n] = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} f[k]\delta[n-k]$$

- **Linear combination of shifted impulse responses:**

$$g[n] = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} f[k]b[n-k] = b[n] * f[n]$$

This convolution operation implies a **shift-invariant system** model, where the response to the input at a specific location is determined by the filter's spatial relationship across all positions.

2.2. (Linear) Convolution

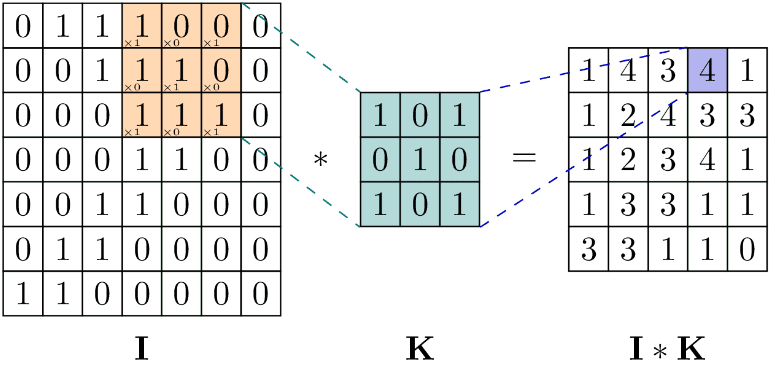


Figure 5 2D Linear Convolution

2.3. Circular Convolution

- Linear convolution becomes N**-point circular convolution** for N-periodic $f[n]$ input, i.e. $f[n] = f[n \mod N]$:

$$g[n] = \tilde{b}[n] \circledast f[n] \triangleq \sum_{k=0}^{N-1} \tilde{b}[(n-k) \mod N] \cdot f[k], \quad n = 0, \dots, N-1$$

  where periodic superposition of $b[n]$ is defined by

$$\tilde{b}[n] = \sum_l b[n - lN]$$

2.3. Circular Convolution

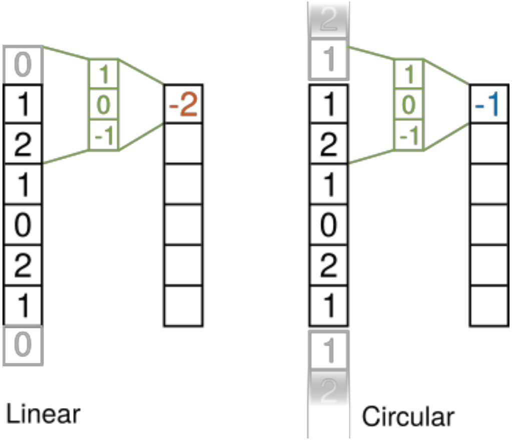


Figure 6 Circular convolution *(N=6)* [2]

2.4. Image Restoration Problem: Blur & Noise

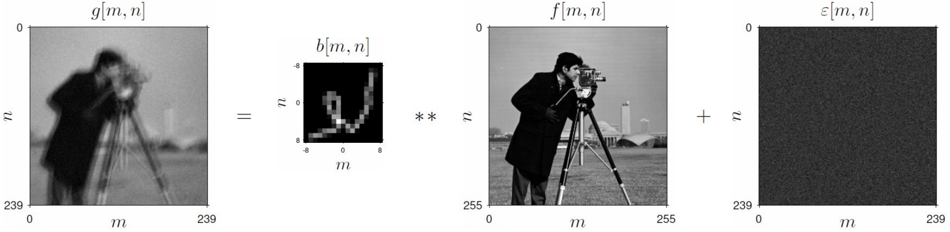


Figure 7 Image restoration

$$g[m,n] = b[m,n] ** f[m,n] + \epsilon[m,n] = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} b[m-k, n-l] f[k,l] + \epsilon[m,n]$$

$b[m,n]$: impulse response of the system, i.e. *point spread function (PSF)*
$\epsilon[m,n]$: additive Gaussian noise

3. Continuous-Discrete Model

- In most image restoration problems, the unknown **"true object"** is a **continuous-space function**, $f(x, y) : x, y \in \mathbb{R}$, so using a discrete-space object and PSF is a *simplification*.

- Continuous-discrete model for image restoration problems:

$$g[m,n] = \bar{g}[m,n] + \varepsilon[m,n], \quad \bar{g}[m,n] = \iint b(m,n;x,y) f(x,y) \, dx \, dy,$$

  where $b(m,n;x,y)$ denotes the contribution that an impulse object located at (x,y) would make to the expected value of $g[m,n]$.

- **System model:** $b(m,n;x,y)$.

3.1. Ill-posed problems

- **Hadamard's Well-Posedness Definition** A mathematical problem is said to be **well-posed** if it satisfies the following three conditions:

  1. **Existence:** There exists a solution to the problem.
  2. **Uniqueness:** The solution is unique.
  3. **Stability:** The solution depends continuously on the data or the input. A small change in the input should result in a small change in the solution.

- *Non-uniqueness* is the main challenge for imaging problems. To make the problem well-posed, we need to impose prior knowledge such as minimum norm estimate.

4. Matrix-Vector Representation

- **Fourier Transform of Non-Periodic Signals**
  - DTFT, Linear Convolution & Non-Periodic Signals (1D)
  - DSFT, Linear Convolution & Non-Periodic Images (2D)
  - Convolution Theorem
- **1D-Matrix Vector Representation**
  - End conditions: Zero, Extended, Periodic
- **2D-Matrix Vector Representation**
  - Representing the Matrix A as Block Matrix
  - End conditions: Zero, Extended, Periodic

4.1. DTFT, Linear Convolution & Non-Periodic Signals

- **Discrete Time Fourier Transform (DTFT)**
  Mathematical tool used to analyze the frequency content of a discrete signal defined in the time domain (1D).

$$b[n] \overset{DTFT}{\longleftrightarrow} B(\Omega) = \sum_{n=-\infty}^{\infty} b[n] e^{-i\Omega n}$$

- **Convolution Theorem**
  Relationship between ***convolution in the time domain*** and ***multiplication in the frequency domain***.

$$g[n] = b[n] * f[n] \overset{DTFT}{\longleftrightarrow} \bar{G}(\Omega) = B(\Omega) \cdot F(\Omega)$$

4.2. DSFT, Linear Convolution & Non-Periodic Images

- **Discrete Space Fourier Transform (DSFT)**
  Mathematical tool used to analyze the frequency content of a discrete signal defined in a spatial domain (2D).

$$B(\Omega_1, \Omega_2) = \sum_{m,n} b[m,n] \cdot e^{-i(\Omega_1 m + \Omega_2 n)}$$

- **Convolution Theorem**
  Relationship between ***convolution in the spatial domain*** and ***multiplication in the frequency domain***.

$$\bar{g}[m,n] = b[m,n] ** f[m,n] \overset{DSFT}{\longleftrightarrow} \bar{G}(\Omega_1, \Omega_2) = B(\Omega_1, \Omega_2) \cdot F(\Omega_1, \Omega_2)$$

4.2. Discrete Space Fourier Transform (DSFT)

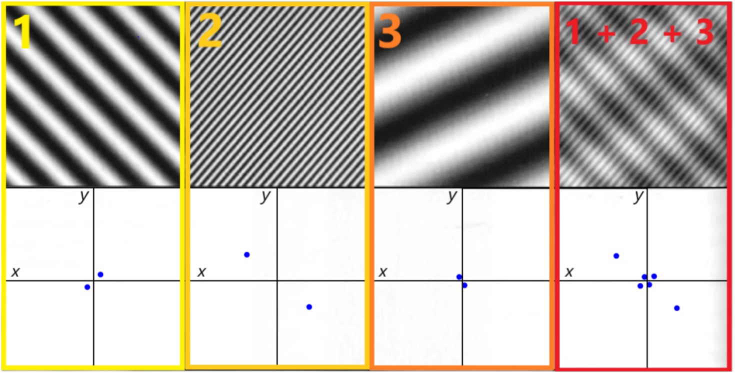


Figure 8 Decomposition of an image [3]

4.3. Convolution Theorem

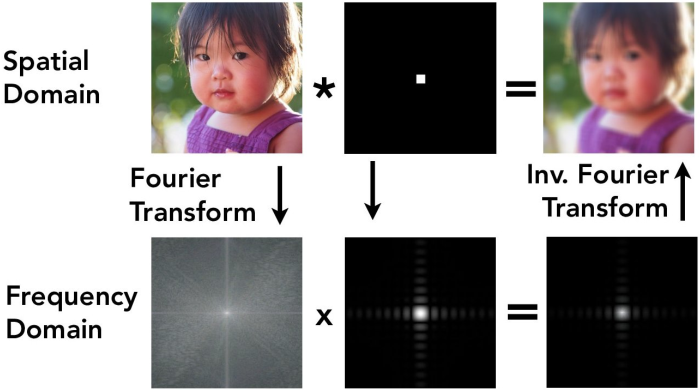


Figure 9 Representation of convolution theorem [4]

5. 1D Matrix-Vector Representation

- Consider a 1D convolution relationship for an input signal $f[n]$ as follows:

$$g[n] = b[n] * f[n] + \varepsilon[n] = \sum_{k=0}^{N-1} b[n-k] \cdot f[k] + \varepsilon[n], \quad n = 0, \ldots, N-1$$

- We want to represent the preceding “DSP-style” formula in this matrix-vector form:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\varepsilon}$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} g[0] \\ \vdots \\ g[N-1] \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} f[0] \\ \vdots \\ f[N-1] \end{bmatrix}$$

5. 1D Matrix-Vector Representation

- The matrix representation of the convolution operation is given by:

$$\underbrace{\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[N-1] \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} b[0] & 0 & \dots & 0 \\ b[1] & b[0] & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ b[N-1] & b[N-2] & \dots & b[0] \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[M-1] \end{bmatrix}}_{\mathbf{x}}$$

- **A** is a matrix defining the convolution operation, i.e. ***system matrix***, **x** is the ***input vector***, and ε is the ***error*** or ***noise vector***.

- **End conditions:** (1) zero end condition, (2) extended end condition, (3) periodic end condition.

5.1. End-Conditions: Zero

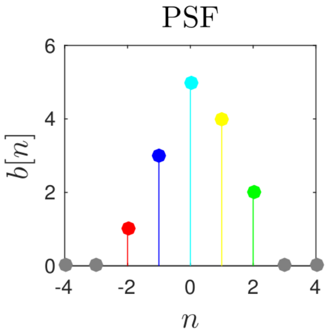
- **Assumption:** $f[n]$ is zero for $n < 0$ and $n \geq N$, using the zero end condition or *Dirichlet boundary condition*.

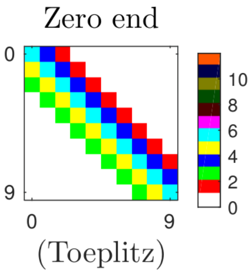
- **Matrix-Vector Representation:** NxN system matrix $\mathbf{A}$ is ***Toeplitz*** (constant along the diagonals).

$$\mathbf{A} = \begin{bmatrix} b[0] & b[-1] & 0 & \cdots & 0 \\ b[1] & b[0] & b[-1] & \cdots & 0 \\ 0 & b[1] & b[0] & b[-1] & \vdots \\ \vdots & & & \ddots & \\ 0 & \cdots & 0 & b[1] & b[0] \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} f[0] \\ \vdots \\ f[N-1] \end{bmatrix}$$

- The elements a_{ij} of matrix $\mathbf{A}$ are connected to a general impulse response function $b[n]$ through the relation $a_{ij} = b[i-j], \quad i,j = 1,\ldots,N.$

5.1. End-Conditions: Zero





5.1. End-Conditions: Zero

- In **shift-invariant** problems, the matrix-vector representation is often used for mathematical *analysis* convenience rather than direct *implementation*.

- In **shift-variant** problems, storing A as a matrix is frequently appropriate. In such cases, a **sparse matrix** representation is often preferred for efficient implementation.

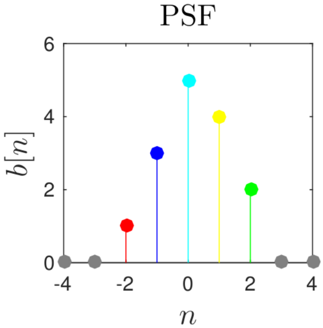
5.2. End-Conditions: Extended

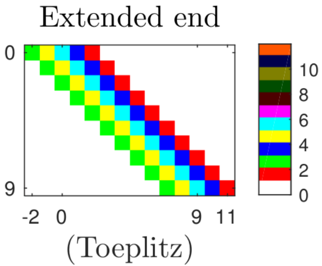
- In many situations, e.g., optical imaging, the measurements are influenced by a larger scene than the field of view of the aperture due to the spreading caused by the imaging system PSF.

- **Matrix-Vector Representation:** *A* is a $Nx(N+L-1)$ rectangular matrix where L is the length of the impulse response ($L = 3$ for this particular $b[n]$).

$$\mathbf{A} = \begin{bmatrix} b[1] & b[0] & b[-1] & 0 & \cdots & 0 & \\ 0 & b[1] & b[0] & b[-1] & \cdots & 0 & \\ & & \ddots & \ddots & \ddots & & \\ 0 & \cdots & 0 & b[1] & b[0] & b[-1] & \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} f[-1] \\ f[0] \\ \vdots \\ f[N-1] \\ f[N] \end{bmatrix}$$

5.2. End-Conditions: Extended





5.2. End-Conditions: Extended

- Extended end conditions are often **more realistic** for restoration problems and should be used when feasible.

- However, often engineers may use **approximations** like "replicated," "mirror," and "periodic" end conditions to save computation.

- Similar to zero end conditions, again the matrix representation of convolution is more useful for *analysis* than for *computation*.

5.3. End-Conditions: Periodic

- **Assumption:** $f[n]$ is N-periodic, i.e. $f[n] = f[n \mod N]$.

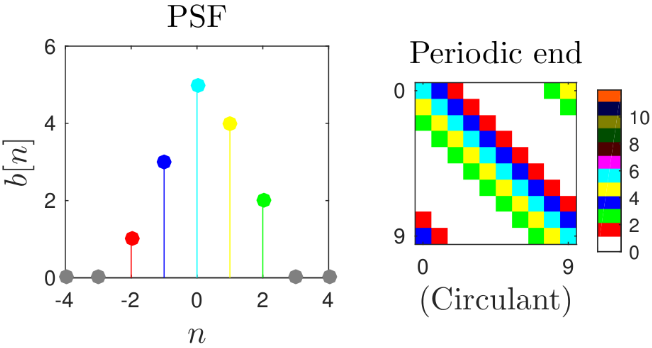
- **Periodic superposition:** $\tilde{b}[n] = \sum_l b[n - lN]$

- **Matrix-Vector Representation:** NxN system matrix *A* is **periodic/circulant**.

$$\mathbf{A} = \begin{bmatrix} \tilde{b}[0] & \tilde{b}[N-1] & \tilde{b}[N-2] & \cdots & \tilde{b}[2] & \tilde{b}[1] \\ \tilde{b}[1] & \tilde{b}[0] & \tilde{b}[N-1] & \tilde{b}[N-2] & \cdots & \tilde{b}[2] \\ & & & \ddots & & \\ \tilde{b}[N-1] & \tilde{b}[N-2] & \tilde{b}[N-3] & \cdots & \tilde{b}[1] & \tilde{b}[0] \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} f[0] \\ \vdots \\ f[N-1] \end{bmatrix}$$

- $a_{ij} = \tilde{b}[(i - j) \mod N], \quad i, j = 1, \ldots, N.$

5.3. End-Conditions: Periodic



6. 2D Matrix-Vector Representation

- **Vectorization operation:** vec : $\mathbb{R}^{M\times N} \rightarrow \mathbb{R}^{MN}$ such that $\mathbf{y} = \text{vec}(\mathbf{g})$.

- **Lexicographic ordering:** Represent a finite-sized image $g[m, n]$ as a "long" vector $\mathbf{y}$ using lexicographic order:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_{MN} \end{bmatrix}, \quad y_i = g[m(i), n(i)], \quad i = 1, \ldots, MN$$

- Vector index i maps to pixel coordinates $[m(i), n(i)]$ as follows:
$m(i) = (i-1) \mod M, \quad n(i) = \left\lfloor \frac{i-1}{M} \right\rfloor$

6.1. 2D Matrix-Vector Representation: Vectorization

$y_1 = g[0,0]$	$y_2 = g[1,0]$	$\dots$	$y_M = g[M-1,0]$
$y_{M+1} = g[0,1]$	$y_{M+2} = g[1,1]$	$\dots$	$y_{2M} = g[M-1,1]$
		$\vdots$	
$y_{M(N-1)+1} = g[0,N-1]$	$y_{M(N-1)+2} = g[1,N-1]$	$\dots$	$y_{MN} = g[M-1,N-1]$

$\rightarrow m$

$\downarrow n$

Figure 10 Vectorization

6.1. 2D Matrix-Vector Representation: Vectorization

Figure 11 Vectorization

6.1. Representing the Matrix A as Block Matrix

- The input-output relationship of any 2D discrete-space **linear** system which are **shift-variant** is as follows:

$$\bar{g}[m,n] = \sum_{k,l} b[m,n;k,l] f[k,l], \quad m = 0,\dots,M-1, \quad n = 0,\dots,N-1.$$

- If $f[m,n]$ is zero outside the domain $m = 0,\dots,M-1$, $n = 0,\dots,N-1$, then the corresponding system matrix A has size $MN \times MN$ and has entries:

$$a_{1+m+nM,1+m+nM} = b[m,n;k,l], \quad m,k = 0,\dots,M-1, \quad n,l = 0,\dots,N-1$$

6.2. Representing the Matrix A as Block Matrix

Block matrix form of system matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{0,0} & \cdots & \mathbf{A}_{0,N-1} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{N-1,0} & \cdots & \mathbf{A}_{N-1,N-1} \end{bmatrix},$$

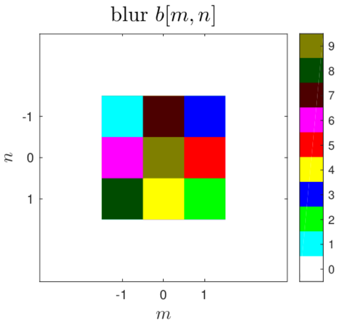
where the $M \times M$ submatrix $\mathbf{A}_{nl}$ describes how the l-th row of the input image contributes to the n-th row of the output image and has elements $[\mathbf{A}_{nl}]_{mk} = b[m, n; k, l]$.

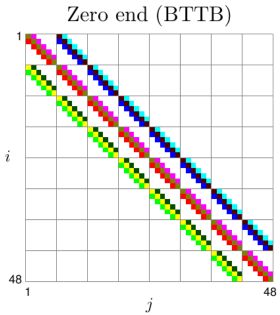
6.3. 2D End conditions: Zero

- **2D linear shift-invariant systems:** $\mathbf{A}$ has the block form where the submatrices have elements:
$$[\mathbf{A}_{nl}]_{mk} = b[m-k, n-l]$$
- Because of the $(m-k)$ dependence, each of the blocks is **Toeplitz** in this shift invariant case, so $\mathbf{A}$ is said to have **Toeplitz blocks**. Furthermore, because of the $(n-l)$ dependence, all of the blocks along each "diagonal" in the block form are the same, so $\mathbf{A}$ is said to be **block Toeplitz**. Combined, we say any such $\mathbf{A}$ is **block Toeplitz with Toeplitz blocks (BTTB).**

6.3. 2D End conditions: Zero

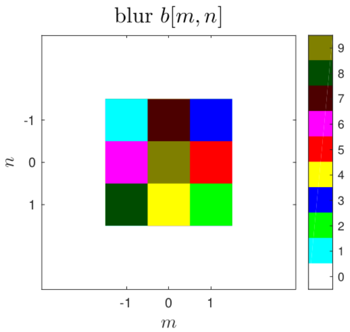
- **Shift-invariant 2D Blur:** Block Toeplitz with Toeplitz Blocks (BTTB)

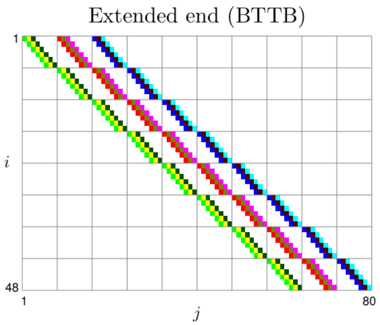




6.4. 2D End Conditions: Extended

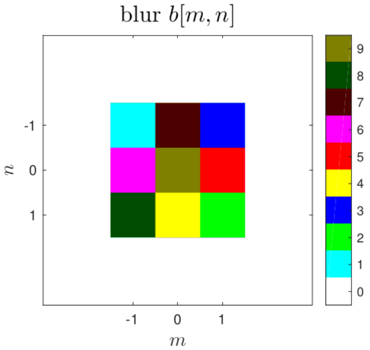
- **Shift-invariant 2D Blur:** Block Toeplitz with Toeplitz Blocks (BTTB)

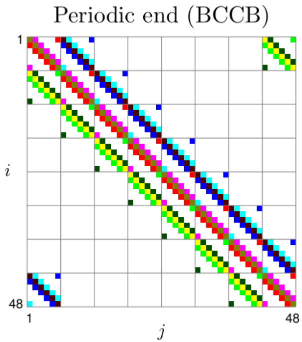




6.5. 2D End Conditions: Periodic

- Block Circulant with Circulant Blocks (BCCB)





7. Circulant Analysis of Shift-Invariant Blur

- Analysis using **circulant matrices** helps us relate matrix algebra solutions to signal processing principles. We reduce computation by replacing large matrix operations with simpler **fast Fourier transform (FFT)** calculations.

- The link between **circulant matrices** and **circular convolution** is the convolution property of the **discrete Fourier transform (DFT)**.

7.1. DFT, Circular Convolution, & Periodic Signals

- **Discrete Fourier Transform:** The N-point DFT of $b[n]$ is given by:

$$b[n] \quad \overset{DFT}{\longleftrightarrow} \quad B_k = \sum_{n=0}^{N-1} b[n] e^{-i\frac{2\pi}{N}kn}, \quad n, k = 0, \ldots, N-1$$

- $\vec{\mathbf{B}} = \mathbf{Q}\mathbf{b}$ where $\mathbf{Q}$ is the $N \times N$ **DFT matrix** having elements

$$q_{kn} = e^{-i\frac{2\pi}{N}kn}, \quad \text{for } k, n = 0, \ldots, N-1$$

- **Convolution Theorem:**

$$\bar{g}[n] = b[n] \circledast_N f[n] \quad \overset{DFT}{\longleftrightarrow} \quad \bar{G}_k = B_k F_k, \quad n, k = 0, \ldots, N-1$$

7.1. DFT, Circular Convolution, & Periodic Signals

- **Discrete Fourier Transform Matrix:**

$$W_N = \begin{bmatrix} \omega^{0\cdot 0} & \omega^{0\cdot 1} & \omega^{0\cdot 2} & \cdots & \omega^{0\cdot (N-1)} \\ \omega^{1\cdot 0} & \omega^{1\cdot 1} & \omega^{1\cdot 2} & \cdots & \omega^{1\cdot (N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega^{(N-1)\cdot 0} & \omega^{(N-1)\cdot 1} & \omega^{(N-1)\cdot 2} & \cdots & \omega^{(N-1)\cdot (N-1)} \end{bmatrix}, \quad \omega = e^{-2\pi i/N}$$

7.1. DFT vs. DTFT

Time Domain	Transform	Frequency Domain
non-periodic, discrete	DTFT	continuous, periodic
periodic, discrete	DFT	discrete, periodic

- **Discrete Fourier Transform:** The N-point DFT of $b[n]$ is given by:

$$b[n] \overset{DFT}{\longleftrightarrow} B_k = \sum_{n=0}^{N-1} b[n] e^{-i\frac{2\pi}{N}kn} = B(\Omega)\Big|_{\Omega=\frac{2\pi}{N}k}, \quad k = 0, \ldots, N-1$$

7.1. DFT vs. DTFT

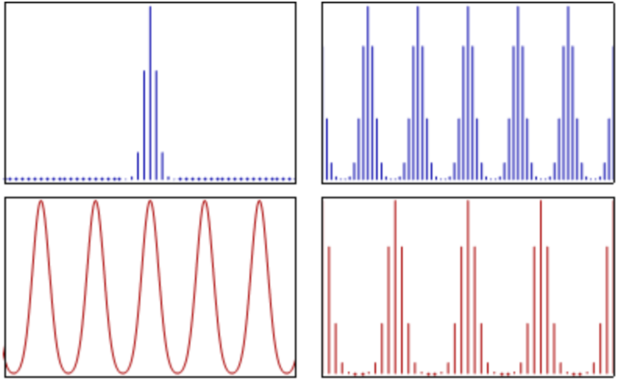


Figure 12 Top: Non-periodic and periodic signals, **Bottom:** Fourier transforms [5]

7.2. Circulant Analysis

$$\underbrace{\begin{bmatrix} \bar{G}_0 \\ \vdots \\ \bar{G}_{N-1} \end{bmatrix}}_{\text{N x 1}} = \underbrace{\begin{bmatrix} B_0 & \cdots & 0 \\ & \ddots & \\ 0 & & B_{N-1} \end{bmatrix}}_{\text{N x N}} \underbrace{\begin{bmatrix} F_0 \\ \vdots \\ F_{N-1} \end{bmatrix}}_{\text{N x 1}}, \text{i.e.} \quad \vec{\mathbf{G}} = \mathbf{\Gamma}\vec{\mathbf{F}}$$

$\vec{\mathbf{G}} = \mathbf{Q}\mathbf{y}$ and $\vec{\mathbf{F}} = \mathbf{Q}\mathbf{x}$, then $\vec{\mathbf{y}} = \mathbf{Q}^{-\mathbf{1}}\mathbf{\Gamma}\mathbf{Q}\mathbf{x}$

$\mathbf{Q}^{-\mathbf{1}} = \frac{1}{N}\mathbf{Q}'$, where $\mathbf{Q}'$ is the Hermitian transpose (conjugate transpose) of $\mathbf{Q}$.

7.2. Circulant Analysis

- Any circulant matrix, e.g., the circulant system matrix for periodic end conditions, has the following matrix decomposition:

$$\mathbf{A} = \mathbf{Q}^{-1}\mathbf{\Gamma}\mathbf{Q} = \frac{1}{N}\mathbf{Q}'\mathbf{\Gamma}\mathbf{Q}$$

- Consider a circulant matrix $\mathbf{A}$ with eigenvalues B_k. The eigenvector decomposition can be expressed as:

$$\mathbf{A} = \mathbf{Q}^{-1}\mathbf{\Gamma}\mathbf{Q}$$

  where Q is the DFT matrix and $\mathbf{\Gamma}$ is a diagonal matrix with the eigenvalues B_k.

- When $\mathbf{A}$ is a circularly shift-invariant filter, frequency response is embedded in the diagonal elements of $\mathbf{\Gamma}$.

8. Simple Image Restoration Problems

- Deconvolution Solution
- Matrix Inverse Solution

8.1. The Deconvolution Solution

- **Convolution Property of Fourier Transform:**
  The **convolution property** of the Fourier transform, where $G(\Omega_1, \Omega_2) = B(\Omega_1, \Omega_2) F(\Omega_1, \Omega_2)$, suggests the following **inverse-filter** solution:

$$\hat{F}(\Omega_1, \Omega_2) = \begin{cases} \frac{G(\Omega_1,\Omega_2)}{B(\Omega_1,\Omega_2)}, & \text{if } B(\Omega_1, \Omega_2) \neq 0 \\ 0, & \text{if } B(\Omega_1, \Omega_2) = 0 \end{cases}$$

- Equivalently, in the spatial domain: $\hat{f}[m, n] = b_{\text{inv}}[m, n] * *g[m, n]$ where $b_{\text{inv}}[m, n]$ is the inverse Fourier transform of $\frac{1}{B(\Omega_1,\Omega_2)}$.

8.2 Matrix Inverse Solution

- **Algebraic reconstruction techniques** are based on linear algebra concepts. From the matrix-vector representation of system functions:

$$\hat{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{y}$$

- **Circulant Matrix Case:**
For initial understanding, consider the case where $\mathbf{A}$ is **circulant**. Then, $\mathbf{A}^{-1} = \mathbf{Q}^{-1}\mathbf{\Gamma}^{-1}\mathbf{Q}$, and the solution becomes:

$$\hat{\mathbf{x}} = \mathbf{Q}^{-1}\mathbf{\Gamma}^{-1}\mathbf{Q}\mathbf{y}$$

$\mathbf{Q}$ corresponds to the DFT matrix, and $\mathbf{\Gamma}^{-1}$ has reciprocals of samples of the system frequency response $B(\Omega)$ along its diagonal.

References I

[1] Wolfram Stiller. “Basics of iterative reconstruction methods in computed tomography: A vendor-independent overview”. In: *European Journal of Radiology* 109 (2018), pp. 147–154. ISSN: 0720-048X. DOI: https://doi.org/10.1016/j.ejrad.2018.10.025.

[2] Stefan Schubert et al. “Circular Convolutional Neural Networks for Panoramic Images and Laser Data”. In: June 2019, pp. 653–660. DOI: 10.1109/IVS.2019.8813862.

[3] Cheat Sheets for Computational Biochemistry. *Waves, Fourier Transform and Reciprocal Space*. (accessed Jan. 09, 2024). June 2020. URL: https://bionerdnotes.wordpress.com/2020/06/03/waves-fourier-transform-and-reciprocal-space/.

[4] Ren Ng. *Computer Graphics and Imaging*. Presented at University of California, Berkeley. 2023.

[5] Wikipedia. *Discrete Fourier transform — Wikipedia, The Free Encyclopedia*. [Online; accessed 9-January-2024]. 2024. URL: https://en.wikipedia.org/w/index.php?title=Discrete_Fourier_transform&oldid=1193803403.