# JPEG: Still Image Coding Standard

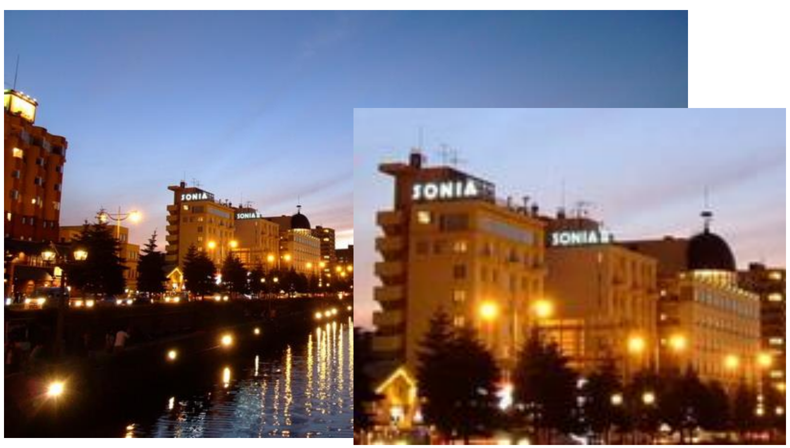

**Chao-Tsung Huang**

**National Tsing Hua University**
**Department of Electrical Engineering**

# JPEG Everywhere

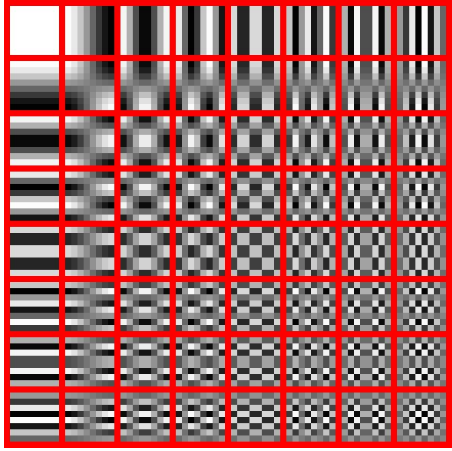
- The most common image format for digital photography
  - Almost everywhere: WWW, digital camera, mobile phone, PC, …
- Typical compression ratio (CR) is 2:1~10:1
  - CR = raw YCbCr data size / compressed data size

CR=6




CR=20

# DCT



- Level shift
  - Encode: for each unsigned P-bit pixel value, subtract $2^{P-1}$ before DCT (ex. 128 for 8-bit inputs)
  - Decode: Add $2^{P-1}$ after DCT, and clamp to the range 0 to $2^P$-1
- 8x8 DCT/IDCT

$$\text{FDCT:} \quad S_{vu} = \frac{1}{4} C_u C_v \sum_{x=0}^{7} \sum_{y=0}^{7} s_{yx} \cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16}$$

$$\text{IDCT:} \quad s_{yx} = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} C_u C_v S_{vu} \cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16}$$

$$C_u, C_v = 1/\sqrt{2} \text{ for } u, v = 0$$
$$C_u, C_v = 1 \text{ otherwise}$$

The order for row and column is not defined.
In practice, floating-point computation inevitably results in distortion.

Ref: *ITU-T Rec. T.81 (JPEG Part-I)*

# Quantization

- Each one coefficient in DCT block has its own Qp. One quantization table for each color component.
- Quantization (informative)
  - Note: f can be adjusted
- Dequantization (normative)

$$Sq_{vu} = round\left(\frac{S_{vu}}{Q_{vu}}\right)$$

$$R_{vu} = Sq_{vu} \times Q_{vu}$$

Default Luma Q-table

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 16 | 11 | 10 | 16 | 24 | 40 | 51 | 61 |
| 12 | 12 | 14 | 19 | 26 | 58 | 60 | 55 |
| 14 | 13 | 16 | 24 | 40 | 57 | 69 | 56 |
| 14 | 17 | 22 | 29 | 51 | 87 | 80 | 62 |
| 18 | 22 | 37 | 56 | 68 | 109 | 103 | 77 |
| 24 | 35 | 55 | 64 | 81 | 104 | 113 | 92 |
| 49 | 64 | 78 | 87 | 103 | 121 | 120 | 101 |
| 72 | 92 | 95 | 98 | 112 | 100 | 103 | 99 |

Default Chroma Q-table

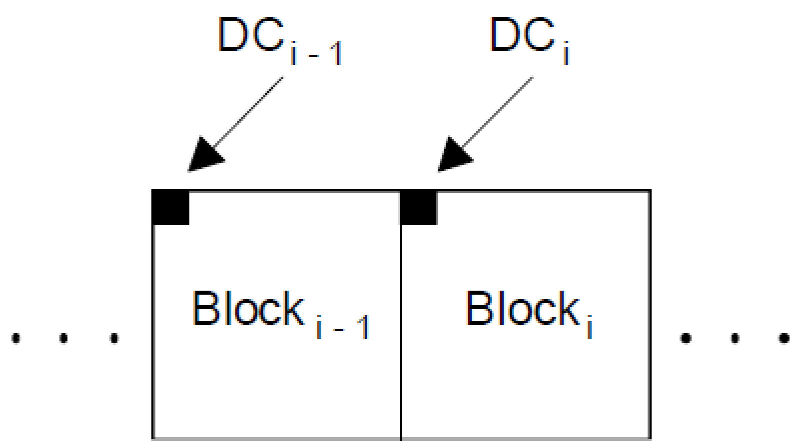
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 17 | 18 | 24 | 47 | 99 | 99 | 99 | 99 |
| 18 | 21 | 26 | 66 | 99 | 99 | 99 | 99 |
| 24 | 26 | 56 | 99 | 99 | 99 | 99 | 99 |
| 47 | 66 | 99 | 99 | 99 | 99 | 99 | 99 |
| 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |
| 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |
| 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |
| 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |

(range: 1~255)

Ref: *ITU-T Rec. T.81 (JPEG Part-I)*

# Preparation of Quantized Coefficients for Entropy Coding

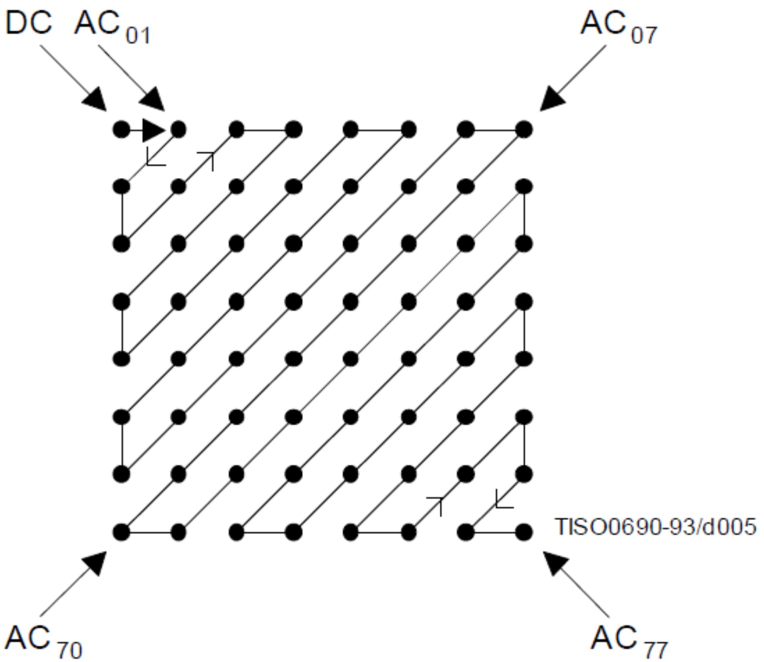
- DC: use DPCM
- AC: use zig-zag scan to transform 2-D data into 1-D sequence




$$DIFF = DC_i - DC_{i-1}$$

Differential DC encoding

(separate predictors for different components, and initialized as zero)




Zig-zag order

Ref: *ITU-T Rec. T.81 (JPEG Part-I)*

# Level Coding of DC Coefficients

- DIFF encoded by one of 12 size categories (SSSS) based on magnitudes
- 'SSSS' encoded by Huffman coding and followed by additional bits for amplitude (bit number is 'SSSS')

**(size)** **(amplitude)**

| SSSS | Difference (DIFF) Values | Additional Bits |
|---|---|---|
| 0 | 0 | — |
| 1 | −1, 1 | 0, 1 |
| 2 | −3, −2, 2, 3 | 00, 01, 10, 11 |
| 3 | −7, ..., −4, 4, ..., 7 | 000, ..., 011, 100, ..., 111 |
| 4 | −15, ..., −8, 8, ..., 15 | 0000, ..., 0111, 1000, ..., 1111 |
| 5 | −31, ..., −16, 16, ..., 31 | 00000, ..., 01111, 10000, ..., 11111 |
| 6 | −63, ..., −32, 32, ..., 63 | ..., ... |
| 7 | −127, ..., −64, 64, ..., 127 | ..., ... |
| 8 | −255, ..., −128, 128, ..., 255 | ..., ... |
| 9 | −511, ..., −256, 256, ..., 511 | ..., ... |
| 10 | −1023, ..., −512, 512, ..., 1023 | ..., ... |
| 11 | −2047, ..., −1024, 1024, ..., 2047 | ..., ... |

(SSSS-bit truncated) DIFF if positive; DIFF-1 if negative

Ref: *Image and Video Compression for Multimedia Engineering*

# Huffman Coding of ‘SSSS’ for DC

- Canonical Huffman table formulated from code number and length

Default Table for Luma

| Category | Code length | Code word |
|---|---|---|
| 0 | 2 | 00 |
| 1 | 3 | 010 |
| 2 | 3 | 011 |
| 3 | 3 | 100 |
| 4 | 3 | 101 |
| 5 | 3 | 110 |
| 6 | 4 | 1110 |
| 7 | 5 | 11110 |
| 8 | 6 | 111110 |
| 9 | 7 | 1111110 |
| 10 | 8 | 11111110 |
| 11 | 9 | 111111110 |

Default Table for Chroma

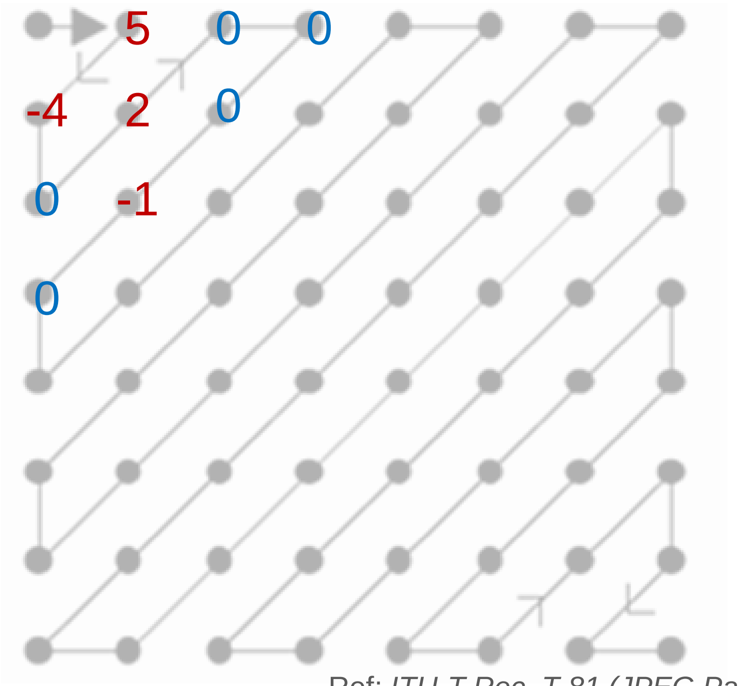
| Category | Code length | Code word |
|---|---|---|
| 0 | 2 | 00 |
| 1 | 2 | 01 |
| 2 | 2 | 10 |
| 3 | 3 | 110 |
| 4 | 4 | 1110 |
| 5 | 5 | 11110 |
| 6 | 6 | 111110 |
| 7 | 7 | 1111110 |
| 8 | 8 | 11111110 |
| 9 | 9 | 111111110 |
| 10 | 10 | 1111111110 |
| 11 | 11 | 11111111110 |

Ref: *ITU-T Rec. T.81 (JPEG Part-I)*

# Run-Length Coding of AC Coefficients

- AC symbol = (Run-length,Size)(Amplitude)
  - Basically encode non-zero coefficients only
  - Run-length: number of zeros before this coefficient
  - Size/Amplitude: same as DC
  - e.g.

For the four non-zero coefficients,
the symbols are:
(0,3)(5);
(0,3)(-4);
(1,2)(2);
(3,1)(-1).




Ref: *ITU-T Rec. T.81 (JPEG Part-I)*

# Huffman Coding of Run/Size for AC

- Canonical Huffman table

Two special symbols:
**EOB**: End of block
**ZRL**: 16 zeros

Default Table for Luma

| Run/Size | Code length | Code word |
|---|---|---|
| 0/0 (EOB) | 4 | 1010 |
| 0/1 | 2 | 00 |
| 0/2 | 2 | 01 |
| 0/3 | 3 | 100 |
| 0/4 | 4 | 1011 |
| 0/5 | 5 | 11010 |
| 0/6 | 7 | 1111000 |
| 0/7 | 8 | 11111000 |
| 0/8 | 10 | 1111110110 |
| 0/9 | 16 | 1111111110000010 |
| 0/A | 16 | 1111111110000011 |
| 1/1 | 4 | 1100 |
| 1/2 | 5 | 11011 |
| … | | |
| F/0 (ZRL) | 11 | 11111111001 |
| F/1 | 16 | 1111111111110101 |
| F/2 | 16 | 1111111111110110 |
| F/3 | 16 | 1111111111110111 |
| F/4 | 16 | 1111111111111000 |
| F/5 | 16 | 1111111111111001 |
| F/6 | 16 | 1111111111111010 |
| F/7 | 16 | 1111111111111011 |
| F/8 | 16 | 1111111111111100 |
| F/9 | 16 | 1111111111111101 |
| F/A | 16 | 1111111111111110 |

Ref: *ITU-T Rec. T.81 (JPEG Part-I)*

# Techniques behind JPEG

- Entropy coding
- DCT

# Entropy

- A coin with 0.5/0.5 probability for head(H)/tail(T)
  - P(H)=p=0.5, P(T)=q=0.5
  - Entropy = (-0.5 $\log_2$0.5) + (-0.5 $\log_2$0.5) = 1 (bit/flip)
  - Entropy coding
    - 0: head (H)
    - 1: tail (T)

- Mathematical fact
  - Optimal bit-length for H = $-\log_2 p$
  - Entropy = average optimal bit per symbol = $-\sum_i p_i \log p_i$

# Cross Entropy

- A coin with 0.75/0.25 probability for head(H)/tail(T)
  - P(H)=p=0.75, P(T)=q=0.25
  - Entropy = (-0.75 $\log_2$0.75) + (-0.25 $\log_2$0.25) = 0.81 (bit/flip)
  - Huffman coding for two-flip symbol (1.68 bit/symbol = 0.84 bit/flip)
    - HH: 0
    - HT: 10
    - TH: 110
    - TT: 111

- Mathematical fact
  - Designed bit-length = $-\log_2 q$
  - Average bit per symbol = $-\sum_i p_i \log q_i \geq -\sum_i p_i \log p_i$ (entropy)

# Entropy of Joint Events

$$H(X,Y) = -\sum_{x,y} p(x,y) \cdot \log p(x,y)$$

**while**

$$H(X) = -\sum_{x,y} p(x,y) \cdot \log p(x)$$

$$H(Y) = -\sum_{x,y} p(x,y) \cdot \log p(y)$$

$$H(X) + H(Y) = -\sum_{x,y} p(x,y) \cdot \log[p(x) \cdot p(y)]$$

**so**

$$\boxed{\begin{gathered} H(X,Y) \leq H(X) + H(Y), \\ "=" \; if \; p(x,y) = p(x) \cdot p(y) \end{gathered}}$$

**Entropy of joint events is less than or equal to separate individual entropy. Equality if events are independent.**

Ref: *The Mathematical Theory of Communication*

# Decorrelation

- Independent => Zero covariance (no correlation)
- Decorrelation: diagonalize covariance matrix

**Signal modeled as**
**1-D vector of random variable**

$$\vec{z} = (z_1, z_2, \ldots, z_N)^T$$

**Covariance matrix**

$$C_{\vec{z}} = E[(\vec{z} - m_{\vec{z}})(\vec{z} - m_{\vec{z}})^T]$$

$$c_{i,j} = E[(z_i - m_i)(z_j - m_j)] = \mathrm{Cov}(z_i, z_j)$$

$$m_{\vec{z}} = E[\vec{z}] = (m_1, m_2, \ldots, m_N)^T$$

**Transform basis (orthonormal)**

$$\Phi = (\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_N)^T$$

**Transformed signal**

$$\vec{y} = \Phi \vec{z}$$

**Covariance matrix after transform**

$$C_{\vec{y}} = \Phi C_{\vec{z}} \Phi^T = \begin{bmatrix} \lambda_1 & & & & 0 \\ & \lambda_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ 0 & & & & \lambda_N \end{bmatrix}$$

**(if diagonalized)**

Ref: *Image and Video Compression for Multimedia Engineering*

# Karhunen-Loeve Transform (KLT)

- Discrete version of KLT: **Diagonalization of covariance matrix**

**Covariance matrix is symmetric and real**

$$\text{ex. } C_z = \begin{bmatrix} 4 & 3 & 2 & 1 \\ 3 & 4 & 2 & 1 \\ 2 & 2 & 4 & 1 \\ 1 & 1 & 1 & 4 \end{bmatrix}$$

**Diagonalizable**
**e.g. Singular Value Decomposition (SVD)**

$$\Phi = \begin{bmatrix} -0.5736 & -0.5736 & -0.4947 & -0.3117 \\ 0.2265 & 0.2265 & 0.0699 & -0.9447 \\ -0.3459 & -0.3459 & 0.8662 & -0.1018 \\ -0.7071 & -0.7071 & 0.0000 & 0.0000 \end{bmatrix}$$

**Transform matrix**

$$C_y = \Phi C_z \Phi^T = \begin{bmatrix} 9.2682 & 0 & 0 & 0 \\ 0 & 3.4465 & 0 & 0 \\ 0 & 0 & 2.2854 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**De-correlated covariance matrix**

**Note: KLT is statistics-dependent**

# Gauss-Markov Model

- A reasonable approximation of images

$$z_i = \rho \cdot z_{i-1} + n_i$$

Pixel value z only depends on its closest neighbor.
n is zero-mean white-noise and independent of z.

$$z_i = \rho^i \cdot z_0 + (\rho^{i-1} n_1 + \rho^{i-2} n_2 + \cdots + n_i)$$

$$Cov(z_i, z_0) = \rho^i \cdot {\sigma_z}^2$$

$$Cov(z_i, z_j) = \rho^{|i-j|} \cdot {\sigma_z}^2$$

$$C_{z,M} = \begin{bmatrix} 1 & \rho^1 & \rho^2 & \rho^3 \\ \rho^1 & 1 & \rho^1 & \rho^2 \\ \rho^2 & \rho^1 & 1 & \rho^1 \\ \rho^3 & \rho^2 & \rho^1 & 1 \end{bmatrix}$$

# KLT for Gauss-Markov Model

- Solution:
  - Given parameters: $N, \rho$

**1. Find the N solutions of $w_k$**
**(k for basis index)**

$$\tan(Nw_k) = \frac{-(1-\rho^2)\sin w_k}{(1+\rho^2)\cos w_k - 2\rho}$$

**2. N corresponding eigenvalues**

$$\lambda_k = \frac{(1-\rho^2)}{1+\rho^2-2\rho\cos w_k}$$

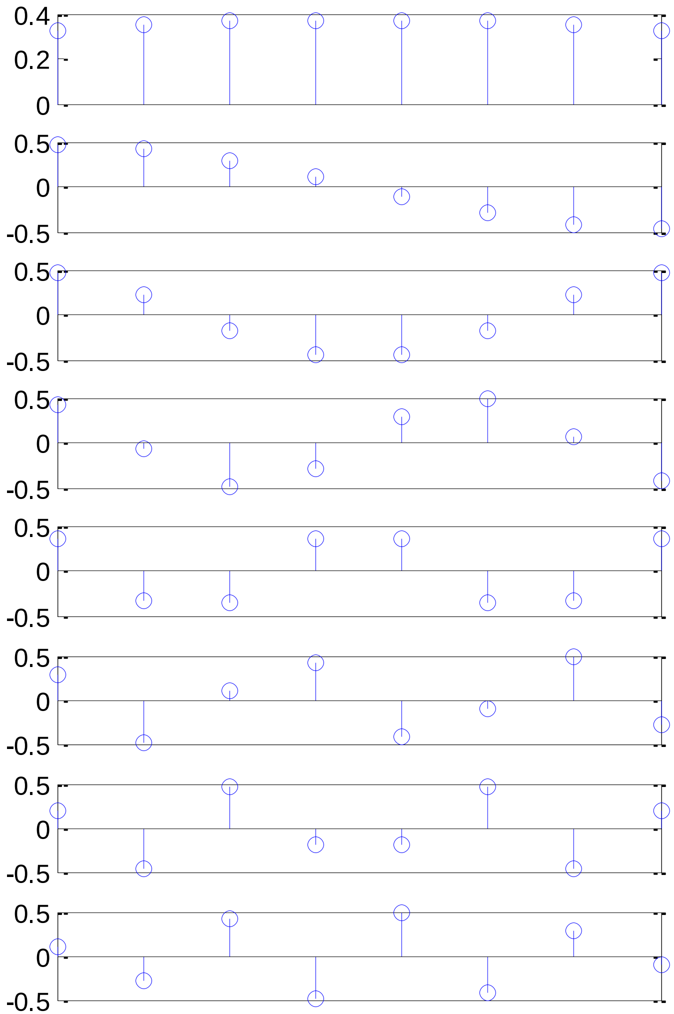
**3. Corresponding eigenvectors**
**(basis vectors)**

$$t_{kn} = \left(\frac{2}{N+\lambda_k}\right)^{\frac{1}{2}} \cdot \sin\left[w_k\left(n-\frac{N+1}{2}\right)+\frac{k}{2}\pi\right]$$

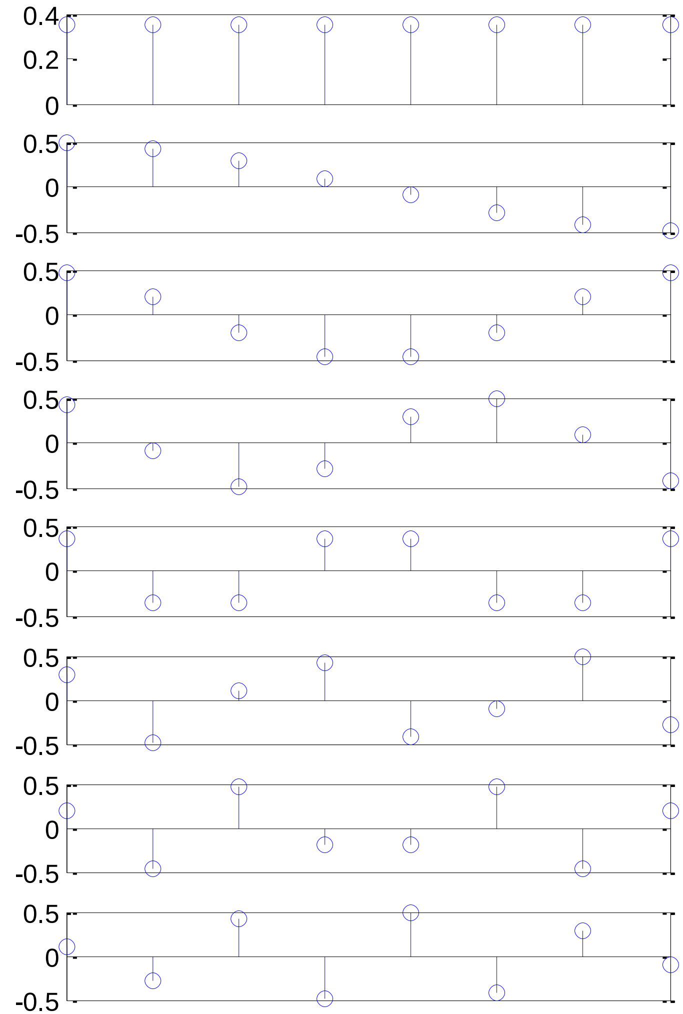
$$\boxed{\boldsymbol{\Lambda} = \boldsymbol{T}\boldsymbol{C}_{\boldsymbol{z},\boldsymbol{M}}\boldsymbol{T}^{-1}, T_{kn} = t_{kn}, \Lambda_{kk} = \lambda_k, \Lambda_{ij,i\neq j} = 0}$$

Ref: W. Ray, *et. al.*, "Further decomposition of the K-L series representation of a stationary random process," *IEEE Trans. Inf. Theory*, 1970.

# 8-point KLT v.s. DCT




**KLT**




**DCT**

**almost the same**

# De-correlation Efficiency

- $\eta_E = 1 - \frac{Y}{X}$

$\boldsymbol{C_{z,M}}$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.0000 | 0.9100 | 0.8281 | 0.7536 | 0.6857 | 0.6240 | 0.5679 | 0.5168 |
| 0.9100 | 1.0000 | 0.9100 | 0.8281 | 0.7536 | 0.6857 | 0.6240 | 0.5679 |
| 0.8281 | 0.9100 | 1.0000 | 0.9100 | 0.8281 | 0.7536 | 0.6857 | 0.6240 |
| 0.7536 | 0.8281 | 0.9100 | 1.0000 | 0.9100 | 0.8281 | 0.7536 | 0.6857 |
| 0.6857 | 0.7536 | 0.8281 | 0.9100 | 1.0000 | 0.9100 | 0.8281 | 0.7536 |
| 0.6240 | 0.6857 | 0.7536 | 0.8281 | 0.9100 | 1.0000 | 0.9100 | 0.8281 |
| 0.5679 | 0.6240 | 0.6857 | 0.7536 | 0.8281 | 0.9100 | 1.0000 | 0.9100 |
| 0.5168 | 0.5679 | 0.6240 | 0.6857 | 0.7536 | 0.8281 | 0.9100 | 1.0000 |

$X = \sum_{i \neq j} |C_{ij}|$

$\Downarrow$ $\eta_E = 100\%$

$\boldsymbol{\Lambda}$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 6.3583 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0.9308 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0.2981 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0.1481 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0.0930 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0.0677 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0.0551 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0490 |

$Y = \sum_{i \neq j} |\Lambda_{ij}|$

# 8-point DCT De-correlation

- For Gaussian-Markov model with $N = 8, \rho = 0.91$

$\boldsymbol{C_{z,M}}$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.0000 | 0.9100 | 0.8281 | 0.7536 | 0.6857 | 0.6240 | 0.5679 | 0.5168 |
| 0.9100 | 1.0000 | 0.9100 | 0.8281 | 0.7536 | 0.6857 | 0.6240 | 0.5679 |
| 0.8281 | 0.9100 | 1.0000 | 0.9100 | 0.8281 | 0.7536 | 0.6857 | 0.6240 |
| 0.7536 | 0.8281 | 0.9100 | 1.0000 | 0.9100 | 0.8281 | 0.7536 | 0.6857 |
| 0.6857 | 0.7536 | 0.8281 | 0.9100 | 1.0000 | 0.9100 | 0.8281 | 0.7536 |
| 0.6240 | 0.6857 | 0.7536 | 0.8281 | 0.9100 | 1.0000 | 0.9100 | 0.8281 |
| 0.5679 | 0.6240 | 0.6857 | 0.7536 | 0.8281 | 0.9100 | 1.0000 | 0.9100 |
| 0.5168 | 0.5679 | 0.6240 | 0.6857 | 0.7536 | 0.8281 | 0.9100 | 1.0000 |

$$X = \sum_{i \neq j} |C_{ij}|$$

$$\eta_E = 98.05\%$$

$$\boldsymbol{D} = \boldsymbol{T_{dct} C_{z,M}} T_{dct}^{\;\boldsymbol{-1}}$$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 6.3435 | 0.0000 | -0.2910 | 0.0000 | -0.0659 | 0.0000 | -0.0209 | -0.0000 |
| 0.0000 | 0.9299 | 0.0000 | -0.0266 | 0.0000 | -0.0080 | 0 | -0.0020 |
| -0.2910 | 0.0000 | 0.3120 | -0.0000 | -0.0013 | -0.0000 | -0.0004 | -0.0000 |
| 0.0000 | -0.0266 | -0.0000 | 0.1490 | 0.0000 | -0.0009 | -0.0000 | -0.0002 |
| -0.0659 | -0.0000 | -0.0013 | -0.0000 | 0.0937 | -0.0000 | -0.0001 | -0.0000 |
| 0.0000 | -0.0080 | -0.0000 | -0.0009 | 0.0000 | 0.0678 | 0.0000 | -0.0001 |
| -0.0209 | -0.0000 | -0.0004 | 0.0000 | -0.0001 | 0.0000 | 0.0552 | 0.0000 |
| -0.0000 | -0.0020 | -0.0000 | -0.0002 | -0.0000 | -0.0001 | -0.0000 | 0.0490 |

$$Y = \sum_{i \neq j} |D_{ij}|$$