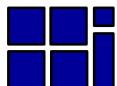


---

# Transform Coding

---

- Principle of block-wise transform coding
- Properties of orthonormal transforms
- Discrete cosine transform (DCT)
- Bit allocation for transform coefficients
- Entropy coding of transform coefficients
- Typical coding artifacts
- Fast implementation of the DCT

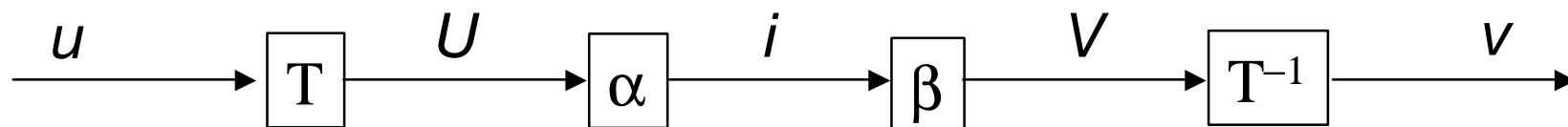


# Transform Coding Principle

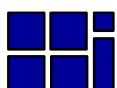
- Structure



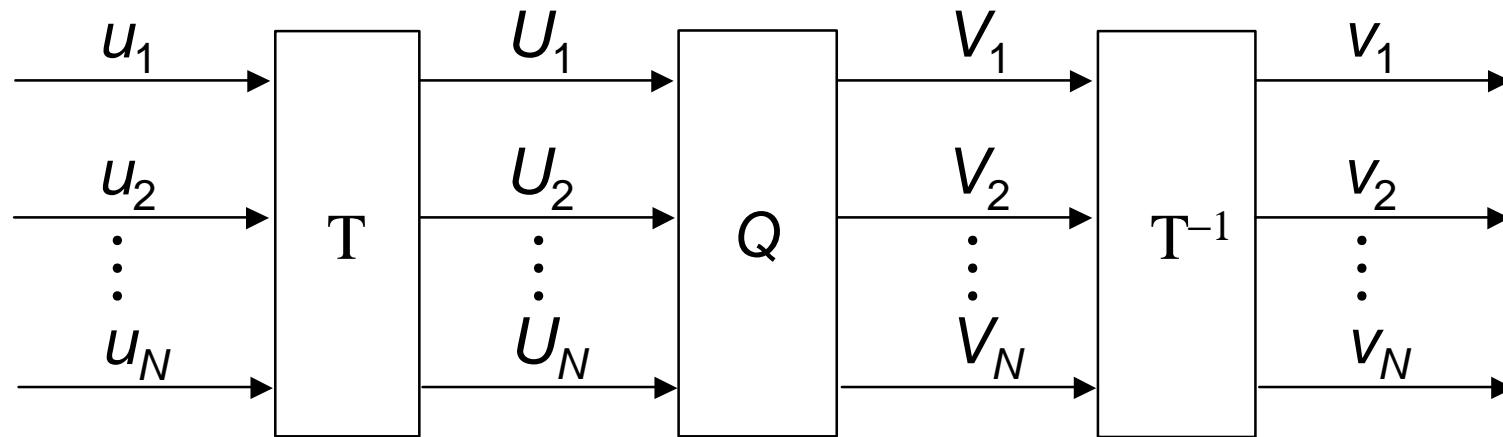
- Transform coder ( $T\alpha$ ) / decoder ( $\beta T^{-1}$ ) structure



- Insert entropy coding ( $\gamma$ ) and transmission channel

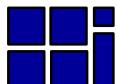


# Transform Coding and Quantization



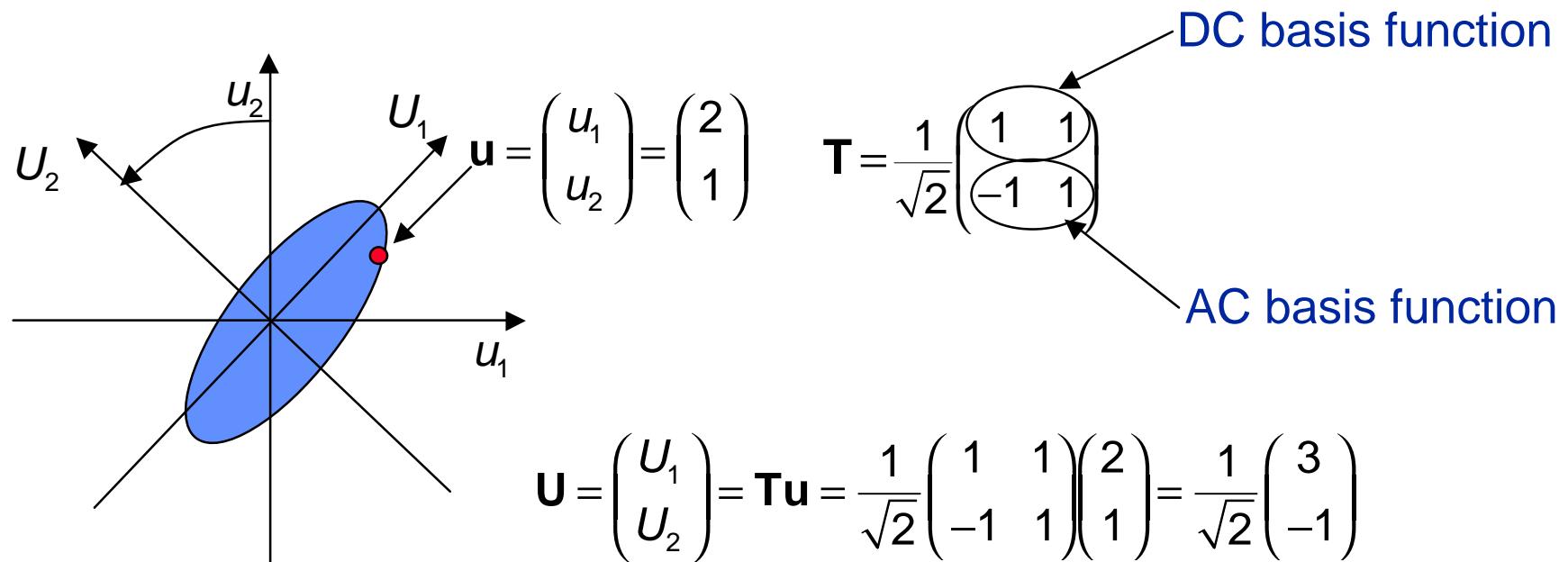
- Transformation of vector  $\mathbf{u}=(u_1,u_2,\dots,u_N)$  into  $\mathbf{U}=(U_1,U_2,\dots,U_N)$
- Quantizer  $Q$  maybe applied to coefficients  $U_i$ 
  - separately (scalar quantization: low complexity)
  - jointly (vector quantization, may require high complexity, exploiting of redundancy between  $U_i$ )
- Inverse transformation of vector  $\mathbf{V}=(V_1,V_2,\dots,V_N)$  into  $\mathbf{v}=(v_1,v_2,\dots,v_N)$

*But what provides a transform ?*



# Geometrical Interpretation

- A linear transform can decorrelate random variables
- An orthonormal transform is a rotation of the signal vector around the origin
- Parseval's Theorem holds for orthonormal transforms



# Properties of Orthonormal Transforms

- Forward transform

$$\mathbf{U} = \mathbf{T}\mathbf{u}$$

$N$  transform coefficients      input signal block of size  $N$   
Transform matrix  
of size  $N \times N$

- Orthonormal transform property: inverse transform

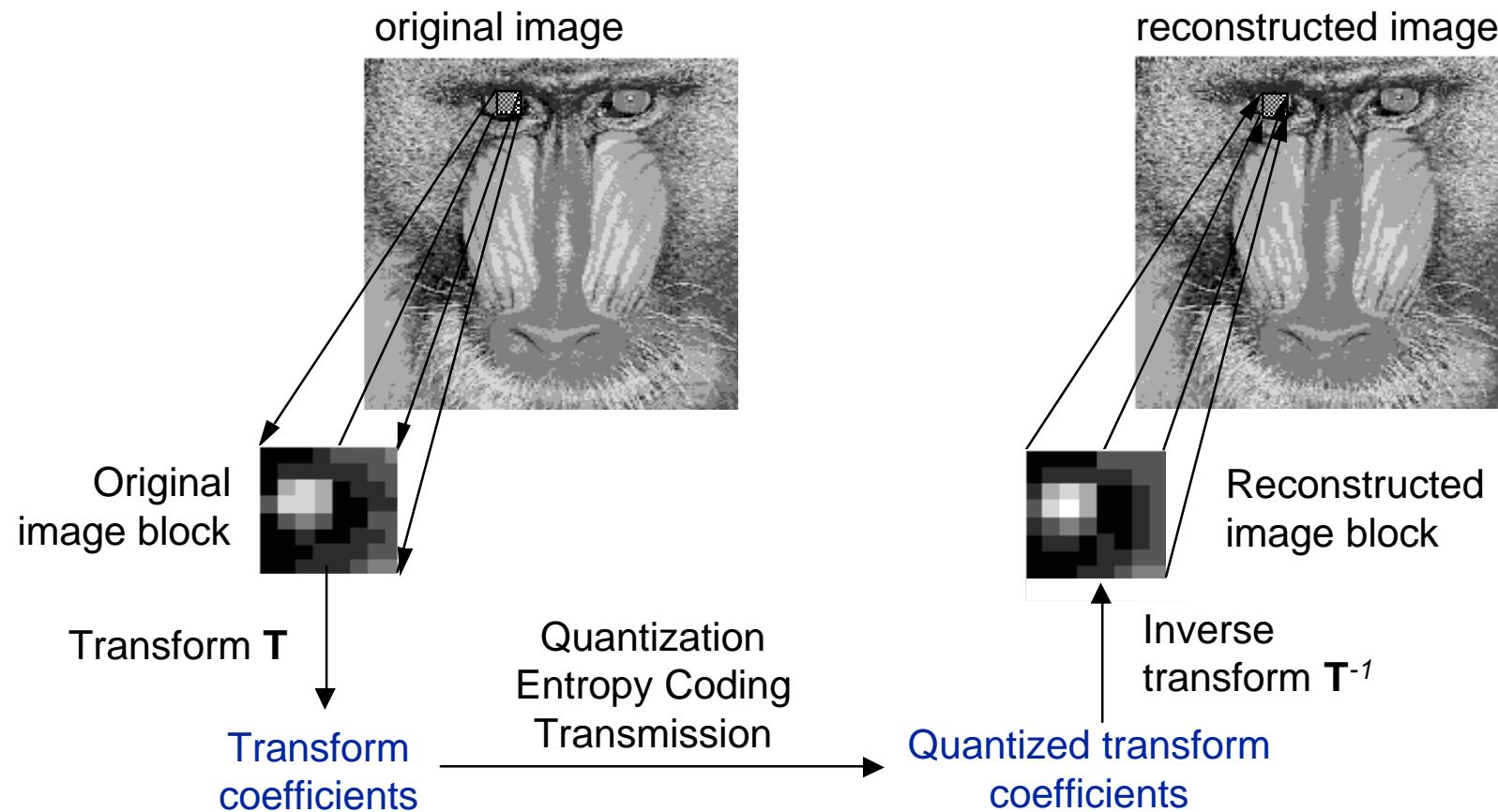
$$\mathbf{u} = \mathbf{T}^{-1}\mathbf{U} = \mathbf{T}^T\mathbf{U}$$

- Linearity:  $\mathbf{u}$  is represented as linear combination of “basis functions”



# Transform Coding of Images

Exploit horizontal and vertical dependencies by processing blocks

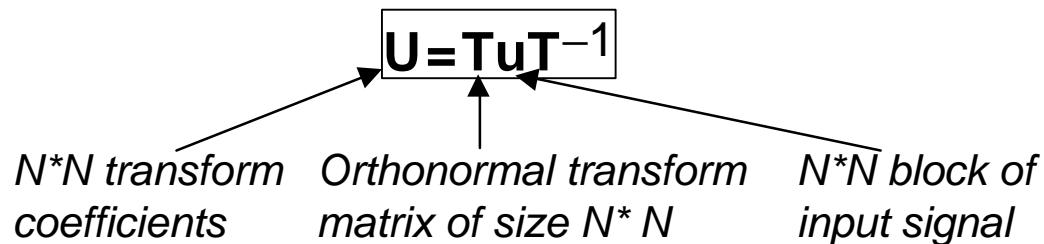


# Separable Orthonormal Transforms, I

- Problem: size of vectors  $N \times N$  (typical value of  $N$ : 8)
- An orthonormal transform is separable, if the transform of a signal block of size  $N \times N$ -can be expressed by

$$\mathbf{U} = \mathbf{T} \mathbf{u} \mathbf{T}^{-1}$$

*N\*N transform coefficients*    *Orthonormal transform matrix of size N \* N*    *N\*N block of input signal*



- The inverse transform is

$$\mathbf{u} = \mathbf{T}^T \mathbf{U} \mathbf{T}$$

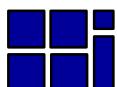
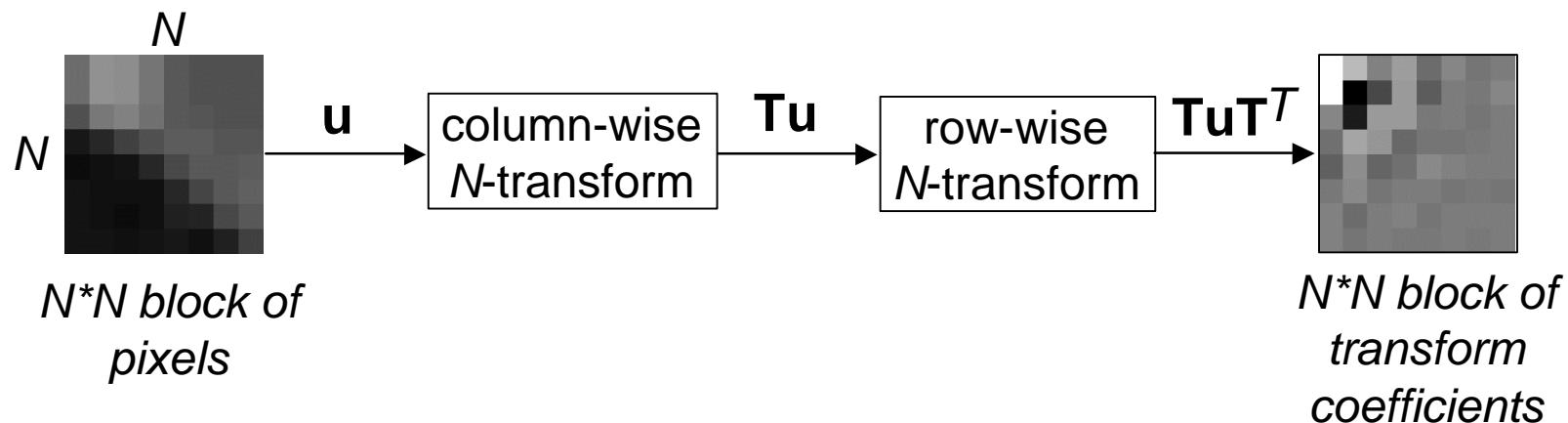
- Great practical importance: transform requires 2 matrix multiplications of size  $N \times N$  instead one multiplication of a vector of size  $1 \times N^2$  with a matrix of size  $N^2 \times N^2$
- Reduction of the complexity from  $O(N^4)$  to  $O(N^3)$



# Separable Orthonormal Transforms, II

Separable 2-D transform is realized by two 1-D transforms

- along rows and
- columns of the signal block

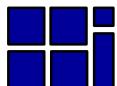


---

# Criteria for the Selection of a Particular Transform

---

- Decorrelation, energy concentration
  - KLT, DCT, ...
  - Transform should provide energy compaction
- Visually pleasant basis functions
  - pseudo-random-noise, m-sequences, lapped transforms, ...
  - Quantization errors make basis functions visible
- Low complexity of computation
  - Separability in 2-D
  - Simple quantization of transform coefficients



---

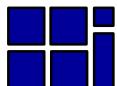
# Karhunen Loève Transform (KLT)

---

- Decorrelate elements of vector  $\mathbf{u}$

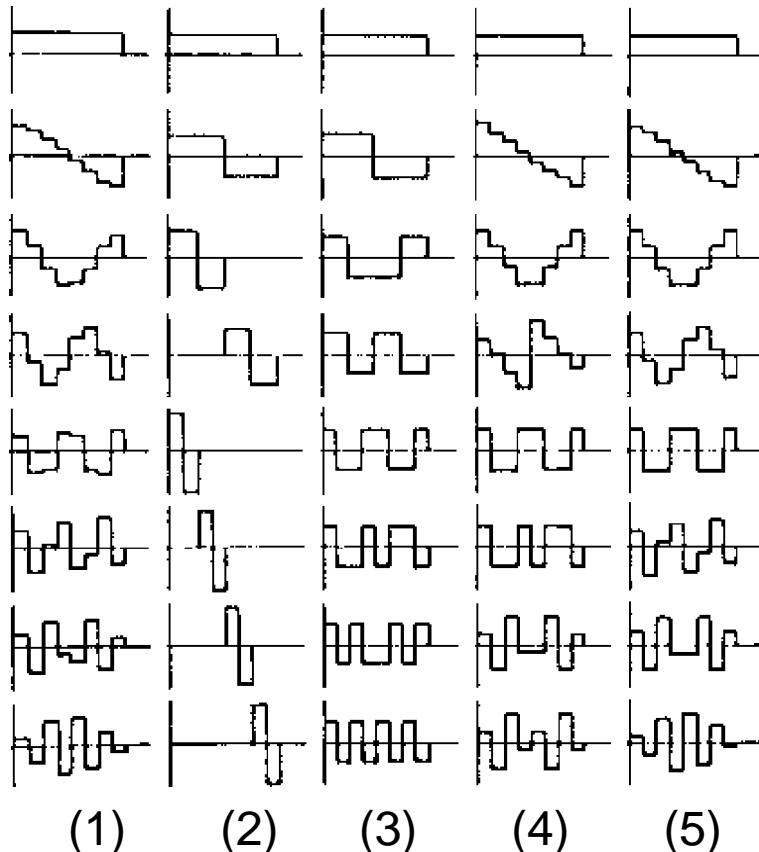
$$\mathbf{R}_u = E\{\mathbf{u}\mathbf{u}^T\}, \rightarrow \mathbf{U} = \mathbf{T}\mathbf{u}, \rightarrow \mathbf{R}_U = E\{\mathbf{U}\mathbf{U}^T\} = \mathbf{T}E\{\mathbf{u}\mathbf{u}^T\}\mathbf{T}^T = \mathbf{T}\mathbf{R}_u\mathbf{T}^T = \text{diag}\{\alpha_i\}$$

- Basis functions are eigenvectors of the covariance matrix of the input signal.
- KLT achieves optimum energy concentration.
- Disadvantages:
  - KLT dependent on signal statistics
  - KLT not separable for image blocks
  - Transform matrix cannot be factored into sparse matrices.



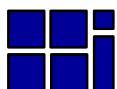
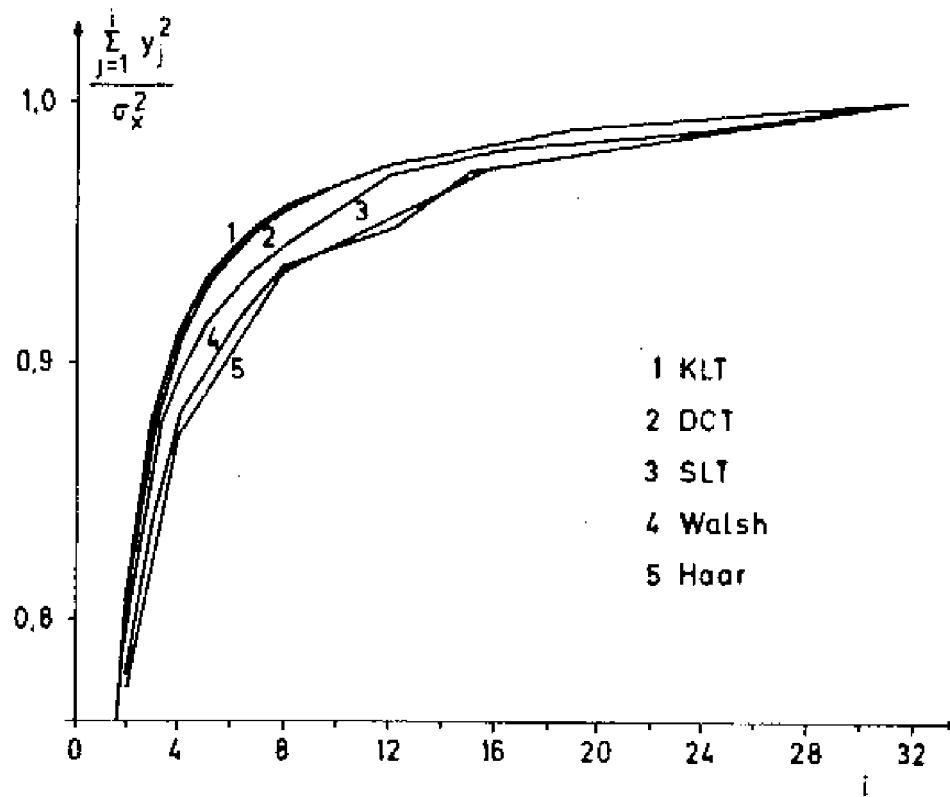
# Comparison of Various Transforms, I

Comparison of 1D basis functions for block size  $N=8$



# Comparison of Various Transforms, II

- Energy concentration measured for typical natural images, block size 1x32 (Lohscheller)
- KLT is optimum
- DCT performs only slightly worse than KLT



# DCT

- Type II-DCT of blocksize  $M \times M$  is defined by transform matrix A containing elements

$$a_{ik} = a_i \cdot \cos \frac{p(2k+1)i}{2M}$$

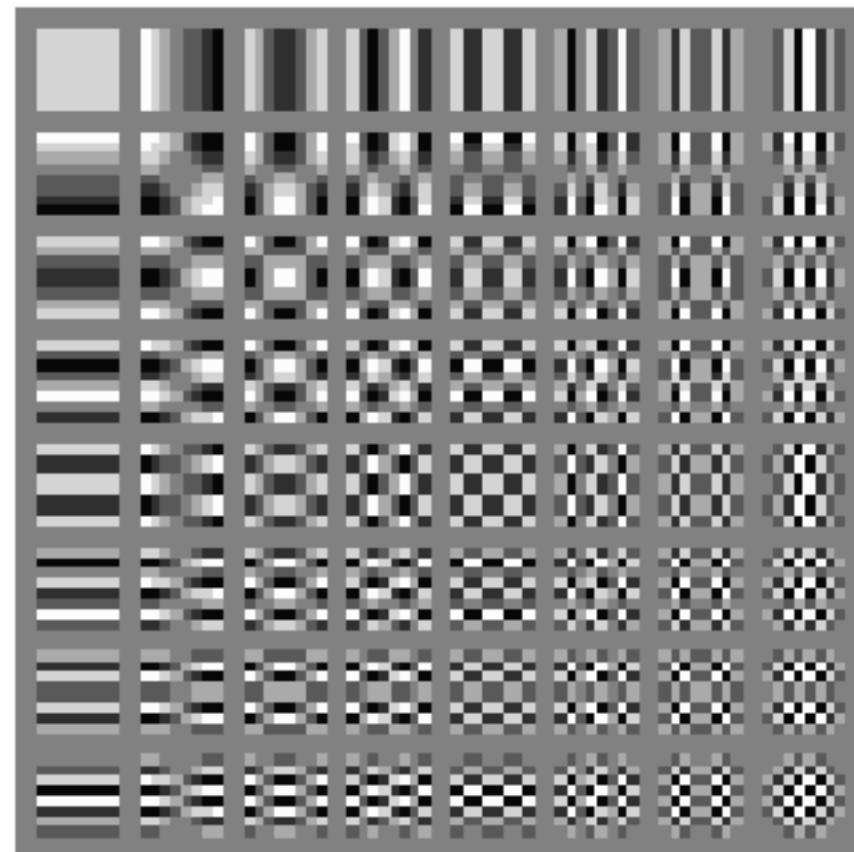
$$i, k = 0 \dots (M-1)$$

with

$$a_0 = \sqrt{\frac{1}{M}}$$

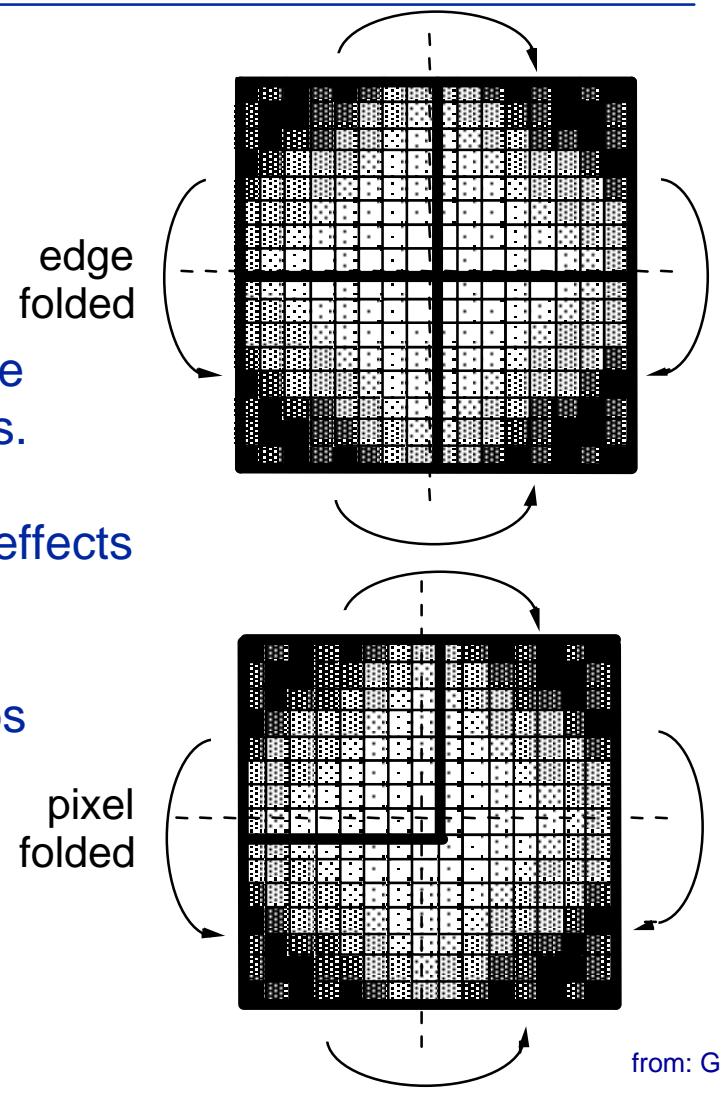
$$a_i = \sqrt{\frac{2}{M}} \quad i \neq 0$$

- 2D basis functions of the DCT:



# Discrete Cosine Transform and Discrete Fourier Transform

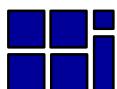
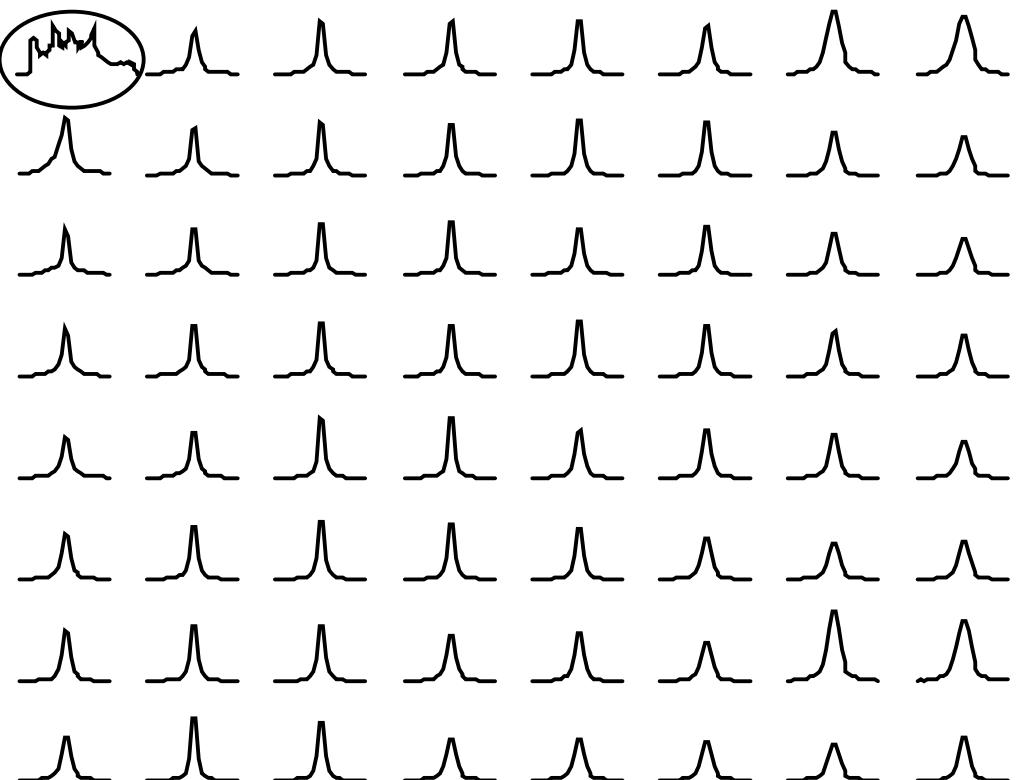
- Transform coding of images using the Discrete Fourier Transform (DFT):
- For stationary image statistics, the energy concentration properties of the DFT converge against those of the KLT for large block sizes.
- Problem of blockwise DFT coding: blocking effects
- DFT of larger symmetric block -> “DCT” due to circular topology of the DFT and Gibbs phenomena.
- Remedy: reflect image at block boundaries



# Histograms of DCT Coefficients:



- Image: Lena,  
256x256 pixel
- DCT: 8x8 pixels
- DC coefficient:  
approximately  
uniformly distributed
- AC coefficients:  
approximately  
Laplacian pdf



# Bit Allocation for Transform Coefficients I

- *Problem:* divide bit-rate  $R$  among  $N$  transform coefficients such that resulting distortion  $D$  is minimized.

$$D(R) = \frac{1}{N} \sum_{i=1}^N D_i(R_i), \quad \text{s.t.} \quad \frac{1}{N} \sum_{i=1}^N R_i \leq R$$

Average distortion      Distortion contributed by coefficient  $i$       Rate for coefficient  $i$       Average rate

- *Approach:* minimize Lagrangian cost function

$$\frac{d}{dR_i} \sum_{i=1}^N D_i(R_i) + \lambda \sum_{i=1}^N R_i = \frac{dD_i(R_i)}{dR_i} + \lambda = 0$$

- *Solution:* Pareto condition

$$\boxed{\frac{dD_i(R_i)}{dR_i} = -\lambda}$$

- Move bits from coefficient with small distortion reduction per bit to coefficient with larger distortion reduction per bit



# Bit Allocation for Transform Coefficients II

- *Assumption:* high rate approximations are valid

$$D_i(R_i) \approx a s_i^2 2^{-2R_i}, \quad \rightarrow \quad \frac{dD_i(R_i)}{dR_i} \approx -2a \ln 2 s_i^2 2^{-2R_i} = -I$$
$$R_i \approx \log_2 s_i + \log_2 \sqrt{\frac{2a \ln 2}{I}} \rightarrow R_i \approx \log_2 s_i - \log_2 \tilde{s} + R = \log_2 \frac{s_i}{\tilde{s}} + R$$
$$R = \frac{1}{N} \sum_{i=1}^N R_i = \underbrace{\frac{1}{N} \sum_{i=1}^N \log_2 s_i}_{\log_2 \tilde{s}} + \log_2 \sqrt{\frac{2a \ln 2}{I}}$$

- Operational Distortion Rate function for transform coding:

$$D(R) = \frac{1}{N} \sum_{i=1}^N D_i(R_i) \approx a s_i^2 2^{-2R_i} = a s_i^2 2^{-2 \log_2 \frac{s_i}{\tilde{s}} - 2R} = a \tilde{s}^2 2^{-2R}$$

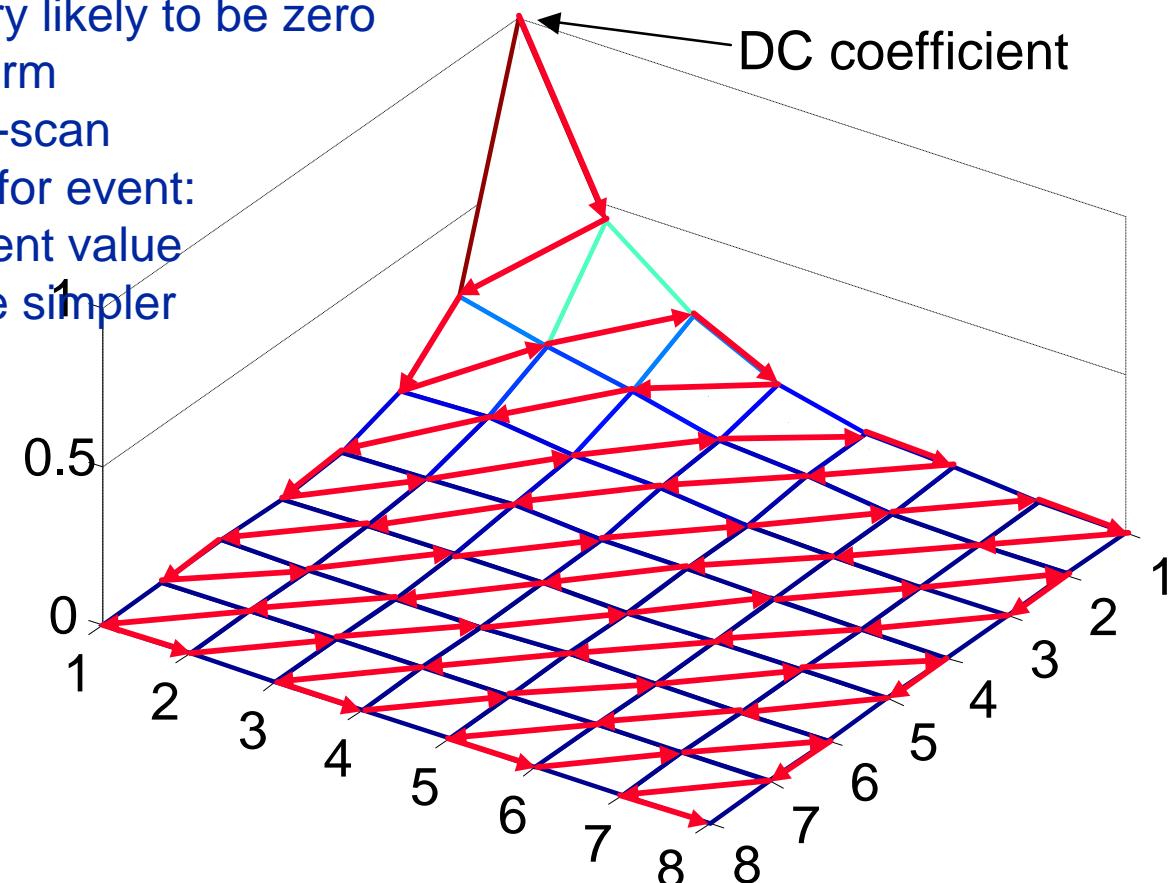
Geometric mean:  $\tilde{s} = \left( \prod_{i=1}^N s_i \right)^{\frac{1}{N}}$



# Entropy Coding of Transform Coefficients

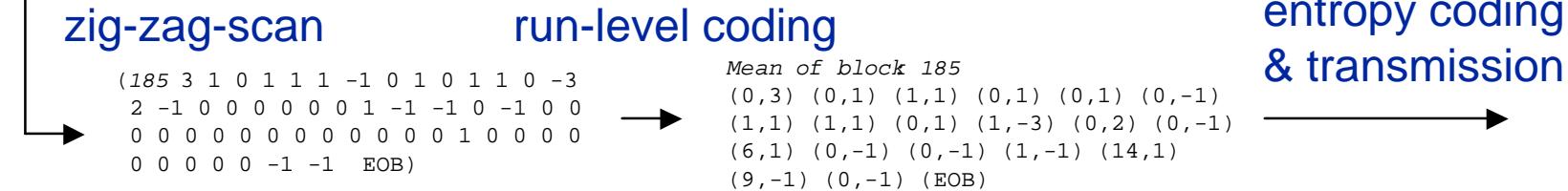
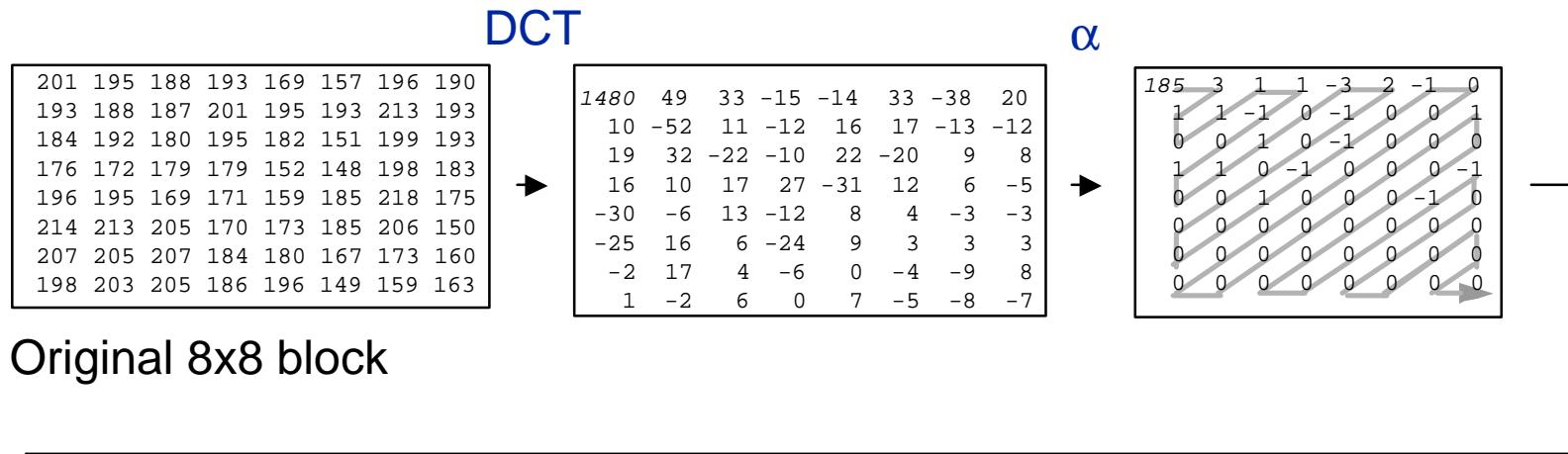
- Previous derivation assumes:  $R_i = \frac{1}{2} \max[(\log_2 \frac{s_i^2}{D}), 0] \text{ bit}$
- AC coefficients are very likely to be zero
- Ordering of the transform coefficients by zig-zag-scan
- Design Huffman code for event: # of zeros and coefficient value
- Arithmetic code maybe simpler

Probability  
that coefficient  
is not zero  
when quantizing with:  
 $V_i = 100 * \text{round}(U_i / 100)$

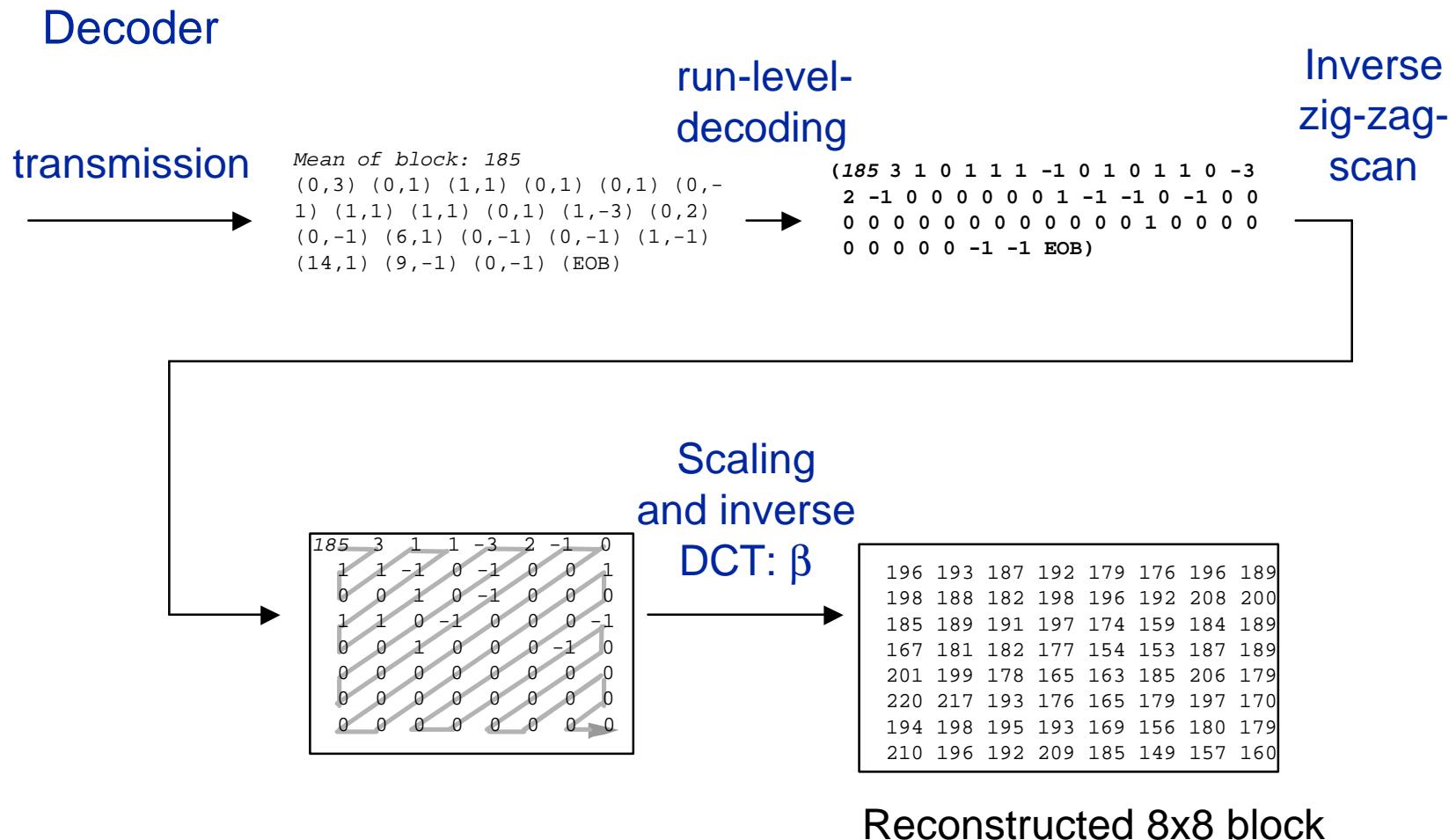


# Entropy Coding of Transform Coefficients II

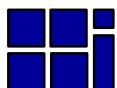
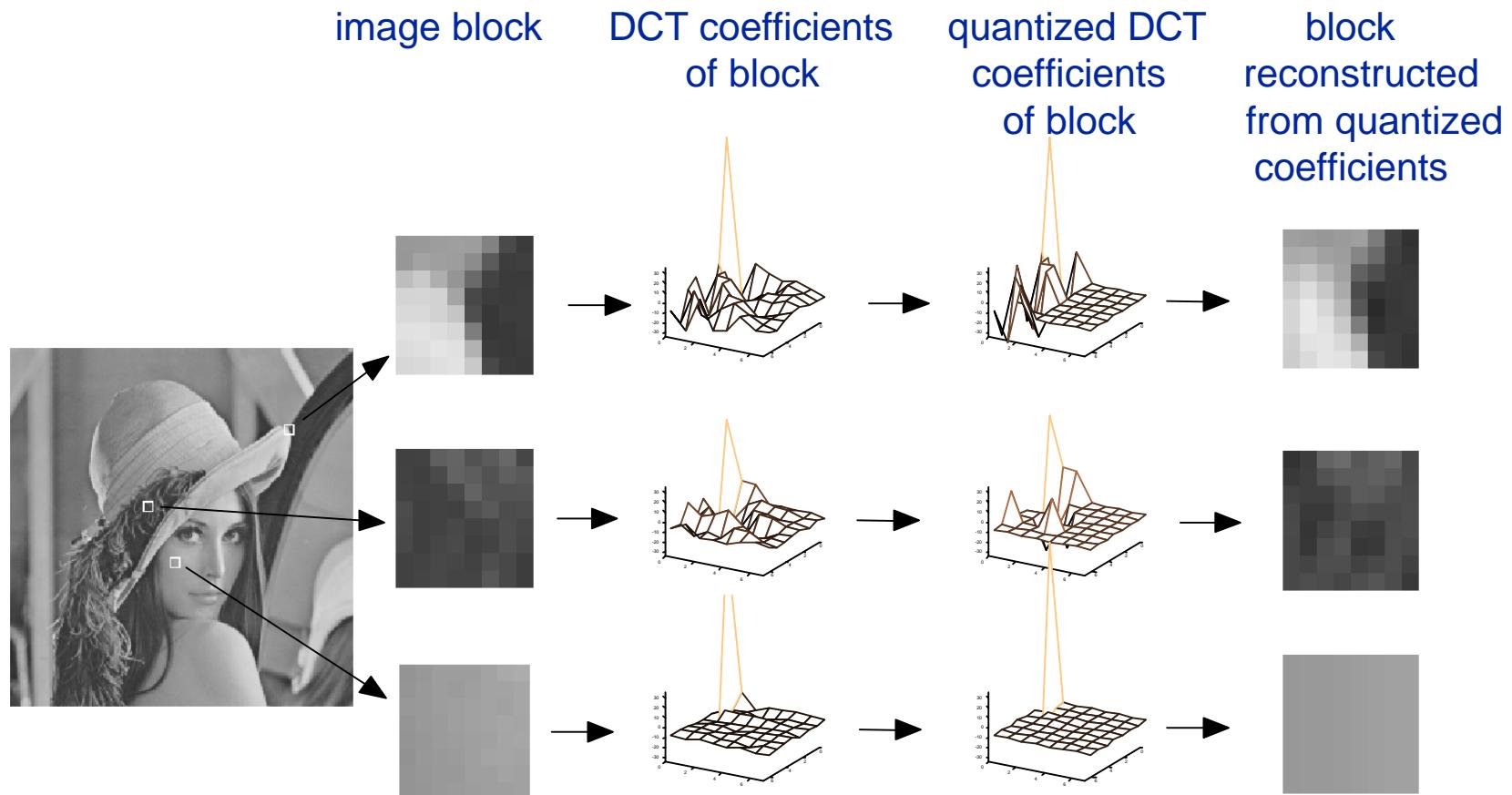
## Encoder



# Entropy Coding of Transform Coefficients III

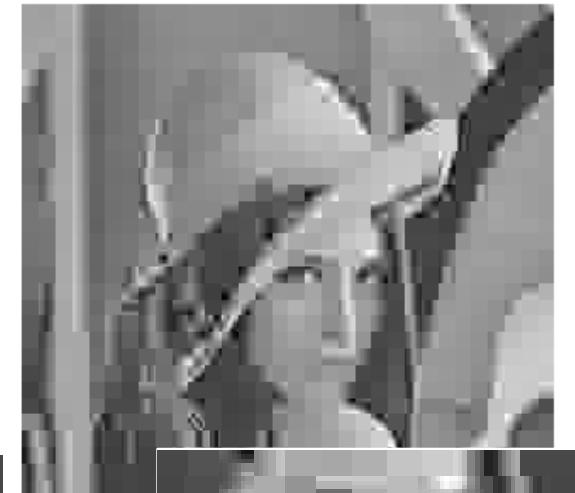


# Detail in a Block vs. DCT Coefficients Transmitted



# Typical DCT Coding Artifacts

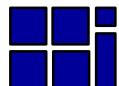
DCT coding with increasingly coarse quantization, block size 8x8



quantizer step size  
for AC coefficients: 25

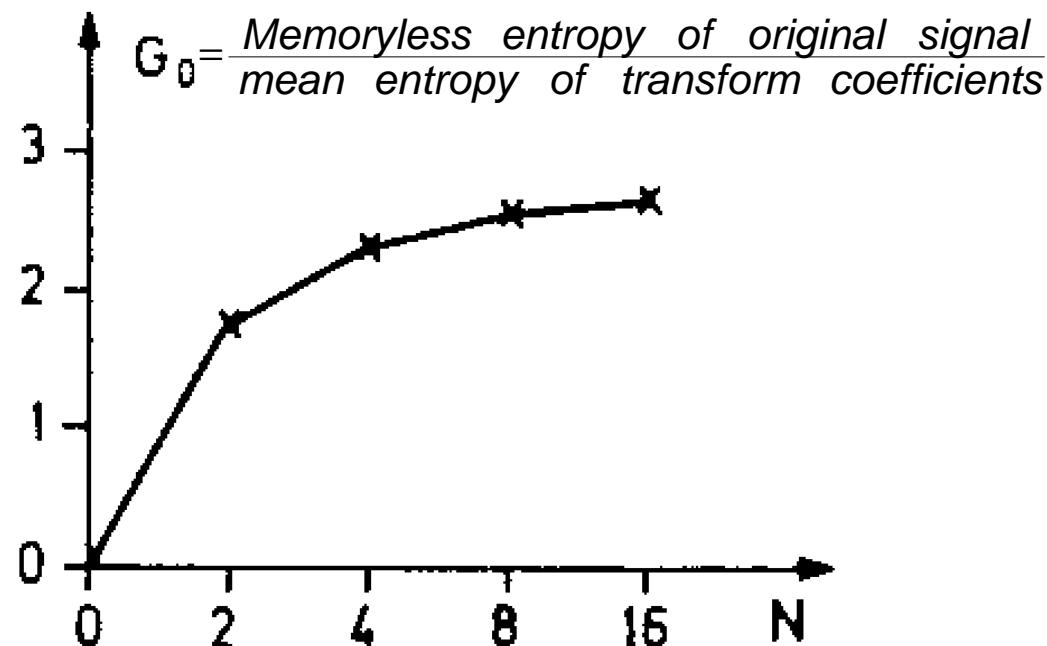
quantizer step size  
for AC coefficients: 100

quantizer step size  
for AC coefficients: 200

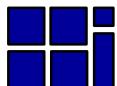


# Influence of DCT Block Size

- Efficiency as a function of blocksize  $N \times N$ , measured for 8 bit Quantization in the original domain and equivalent quantization in the transform domain.



- Block size  $8 \times 8$  is a good compromise.

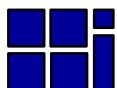


# Fast DCT Algorithm I

DCT matrix factored into sparse matrices (Arai, Agui, and Nakajima; 1988):

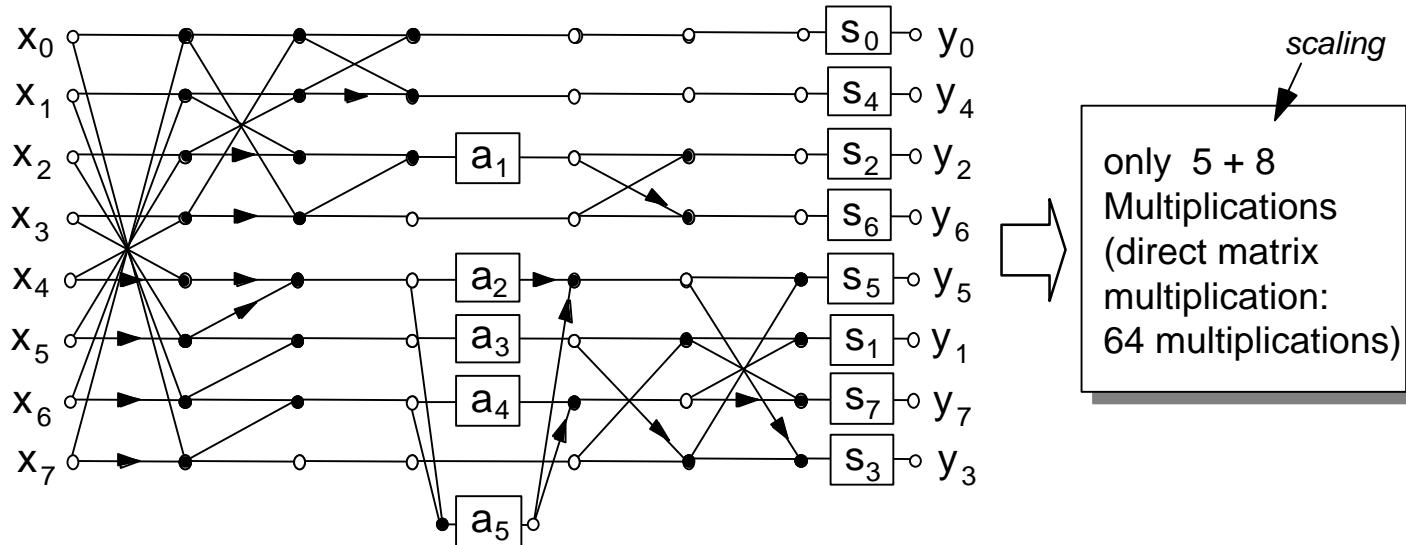
$$\begin{aligned} \underline{y} &= \underline{M} \cdot \underline{x} \\ &= \underline{S} \cdot \underline{P} \cdot \underline{M}_1 \cdot \underline{M}_2 \cdot \underline{M}_3 \cdot \underline{M}_4 \cdot \underline{M}_5 \cdot \underline{M}_6 \cdot \underline{x} \end{aligned}$$

$$\begin{aligned} \underline{S} &= \begin{bmatrix} s_0 & & & & & & & \\ s_1 & s_2 & & & & & & \\ s_2 & s_3 & 0 & & & & & \\ s_3 & s_4 & & 0 & & & & \\ s_4 & s_5 & & & 0 & & & \\ s_5 & s_6 & & & & 0 & & \\ s_6 & s_7 & & & & & 0 & \\ 0 & & & & & & & \end{bmatrix} \quad \underline{P} = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{bmatrix} \quad \underline{M}_1 = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 0 & \\ 0 & & & & & & & 1 \end{bmatrix} \quad \underline{M}_2 = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 0 & \\ 0 & & & & & & & 1 \end{bmatrix} \\ \underline{M}_3 &= \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & C_4 & 0 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 0 & \\ 0 & & & & & & & 1 \end{bmatrix} \quad \underline{M}_4 = \begin{bmatrix} 1 & 1 & & & & & & \\ & 1 & -1 & & & & & \\ & & 1 & 1 & 0 & & & \\ & & & 1 & 1 & & & \\ & & & & 1 & 1 & & \\ & & & & & 1 & 1 & \\ 0 & & & & & & & 1 \end{bmatrix} \quad \underline{M}_5 = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 0 & \\ 0 & & & & & & & 1 \end{bmatrix} \quad \underline{M}_6 = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 0 & \\ 0 & & & & & & & 1 \end{bmatrix} \end{aligned}$$



# Fast DCT Algorithm II

Signal flow graph for fast (scaled) 8-DCT according to Arai, Agui, Nakajima:



Multiplication:

$$u \circ [m] \circ m \cdot u$$

Addition:

$$u \circ \begin{cases} v \\ \end{cases} \rightarrow u+v$$

$$u \circ \begin{cases} v \\ \end{cases} \rightarrow u-v$$

$$a_1 = C_4$$

$$a_2 = C_2 - C_6$$

$$a_3 = C_4$$

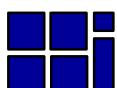
$$a_4 = C_6 + C_2$$

$$a_5 = C_6$$

$$s_0 = \frac{1}{2 \cdot \sqrt{2}}$$

$$s_k = \frac{1}{4 \cdot C_k}; k=1 \dots 7$$

$$C_k = \cos\left(\frac{ks}{16}\right)$$



---

# Transform Coding: Summary

---

- Orthonormal transform: rotation of coordinate system in signal space
- Purpose of transform: decorrelation, energy concentration
- KLT is optimum, but signal dependent and, hence, without a fast algorithm
- DCT shows reduced blocking artifacts compared to DFT
- Bit allocation proportional to logarithm of variance
- Threshold coding + zig-zag-scan + 8x8 block size is widely used today (e.g. JPEG, MPEG, ITU-T H.263)
- Fast algorithm for scaled 8-DCT: 5 multiplications, 29 additions

