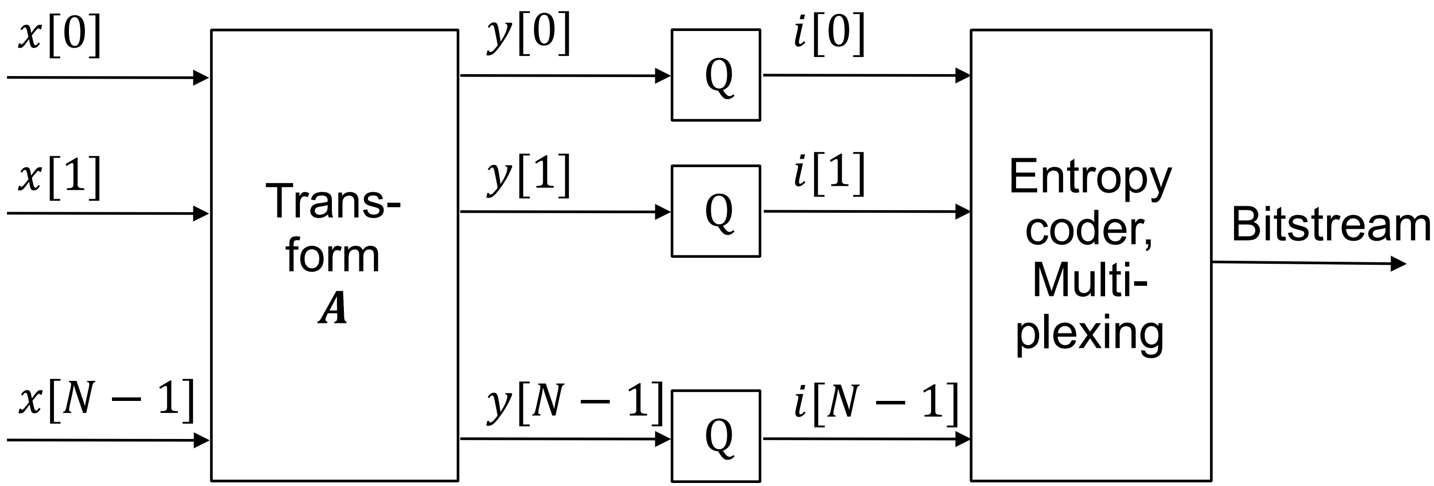
7 Transform Coding

7.1 Principle of Transform Coding

Goal: transform input signal such that the output signal has following properties

- Reduced statistical dependencies between its samples
- Energy is packed into only a few transform coefficients

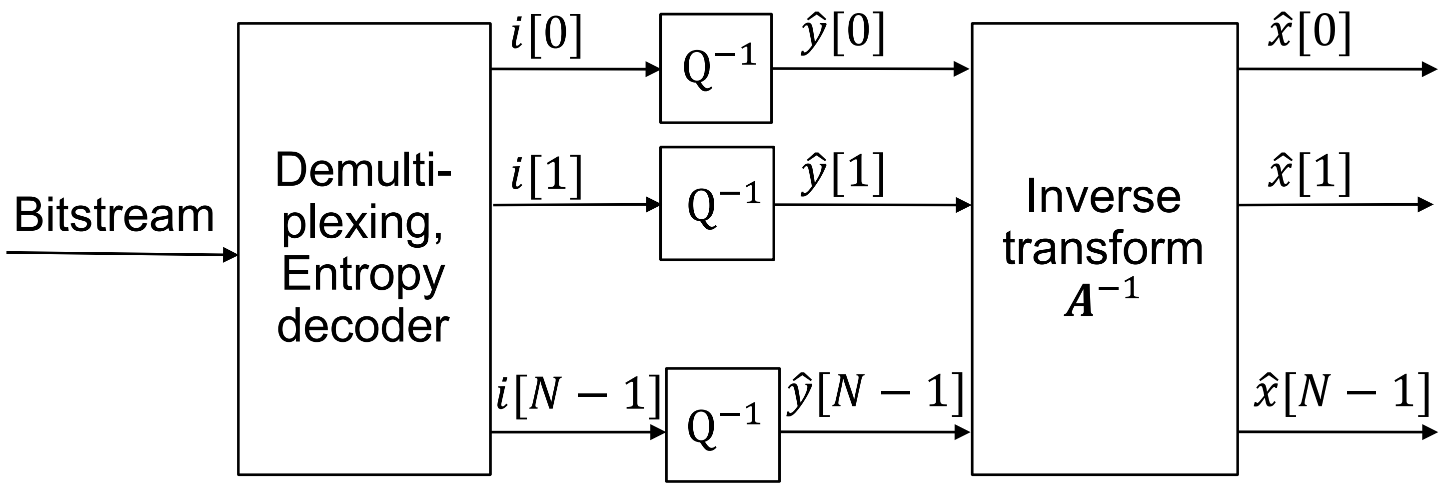
Block diagram of transform coder



- Decompose signal $x[n]$ into vectors or blocks of N input samples
- Transform each block individually into transform coefficients $y[n]$
- Quantize transform coefficients individually
- Code indices $i[n]$ by fixed length or entropy code

Transform Decoder

Block diagram of transform decoder: invert steps of coding process



- Reconvert incoming bitstream and decode quantization indices
- Replace indices for transform coefficient by representative value of quantization interval
- Apply inverse transform to calculate reconstructed signal

⇨ Lossy compression technique due to quantization, i.e. $\hat{x}[n] \neq x[n]$ in general

7.2 Orthonormal Transforms

Forward transform

- Reorder signal samples $x[n]$ as a column vector $\boldsymbol{x}$ of length N
- Apply transform matrix $\boldsymbol{A} = (\boldsymbol{a_0}, \boldsymbol{a_1}, \dots, \boldsymbol{a_{N-1}})^{\mathrm{T}}$ with size $N \times N$

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \quad \Leftrightarrow \quad y[k] = \sum_{n=0}^{N-1} a_k[n]x[n], \qquad 0 \le k \le N-1$$

- $y[k]$ are called transform coefficients

Orthonormal (unitary) transform

- any two different rows of transform matrix $\boldsymbol{A}$ are orthogonal to each other

$$\sum_{n=0}^{N-1} a_k[n]a_l[n] = \delta[k-l] = \begin{cases} 1 & \text{for} \quad k = l \\ 0 & \text{for} \quad k \ne l \end{cases} \quad \Rightarrow \quad \boldsymbol{A}^{-1} = \boldsymbol{A}^{\mathrm{T}}$$

Inverse transform simplifies to

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{y} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} \quad \Rightarrow \quad x[n] = \sum_{k=0}^{N-1} y[k]a_k[n], \qquad 0 \le n \le N-1$$

- Inverse transform can be regarded as series expansion of input signal
- Rows of $\boldsymbol{A}$ are called basis vectors $\boldsymbol{a}_k$

Two-Dimensional Orthonormal Transforms

Image processing: generalize 1D transform to two dimensions

- Define $N{\times}N$ image $x[m,n]$ over a square block with $m,n = 0,\dots,N-1$
- Replace basis vectors by *basis images* $\boldsymbol{a}_{kl}$

$$y[k,l] = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x[m,n]a_{kl}[m,n] \qquad 0 \le k,l \le N-1$$

- Elements $y[k,l]$ are called transform coefficients
- Matrix $\boldsymbol{Y} = \{y[k,l]\}\ \ 0 \le k,l \le N-1$ is the transformed image

Orthogonality: two-dimensional basis images satisfy

$$\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} a_{kl}[m,n]a_{k'l'}[m,n] = \delta[k-k',l-l']$$

Inverse transform given by $x[m,n] = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} y[k,l]a_{kl}[m,n] \qquad 0 \le m,n \le N-1$

- Series expansion of an image $x[m,n]$ into real valued basis images $\boldsymbol{a}_{kl}$

Separable Two-Dimensional Orthonormal Transforms

Problem: two-dimensional transform requires huge computation time

- $N \times N$ multiplications for each 2D transform coefficient ⇨ $O(N^4)$

Separable orthonormal transform

- Restrict basis images to be separable

$$a_{kl}[m,n] = a_k[m]a_l[n] \quad 0 \le k,l \le N-1$$

- Forward and inverse transform simplify to

$$y[k,l] = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} a_k[m]x[m,n]a_l[n] \quad \Leftrightarrow \quad \boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}\boldsymbol{A}^{\mathrm{T}}$$

$$x[m,n] = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} a_k[m]y[k,l]a_l[n] \quad \Leftrightarrow \quad \boldsymbol{X} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{A}$$

- Two matrix multiplications with $(N \times N)$ elements and N multiplications for calculating each element

⇨ Computational complexity reduces to $O(2N^3)$

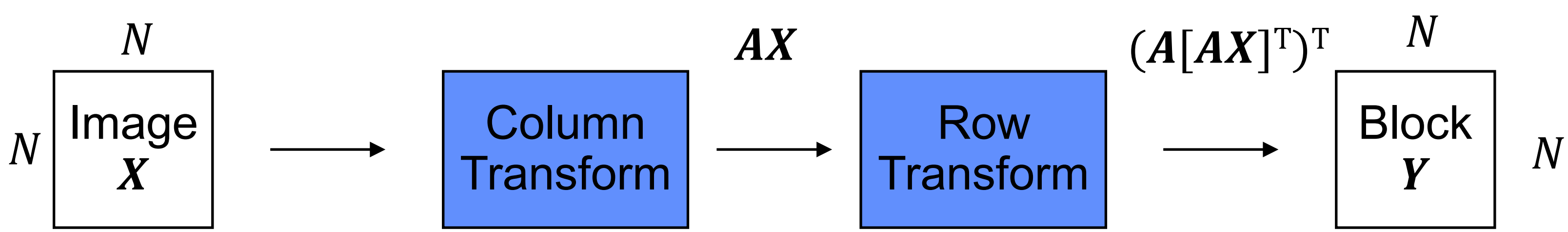
Implementation of Separable 2D Transforms

Realization: consecutive transform of image columns and rows

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}\boldsymbol{A}^{\mathrm{T}} \Rightarrow \boldsymbol{Y}^{\mathrm{T}} = \boldsymbol{A}[\boldsymbol{A}\boldsymbol{X}]^{\mathrm{T}}$$

- First transform each column of $\boldsymbol{X}$, then transform each row of the result

Block diagram



Inverse transform

$$\boldsymbol{X} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{A} \Rightarrow \boldsymbol{X}^{\mathrm{T}} = \boldsymbol{A}^{\mathrm{T}}[\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Y}]^{\mathrm{T}}$$

- First inverse transform each column of $\boldsymbol{Y}$, then inverse transform each row of the result

Tensor Notation for 2D Transforms

- Order samples of image $\boldsymbol{X} = \{x[m,n]\}$ into a column vector of length N^2 by scanning the image row by row

$$\mathbf{x} = \{x[0{,}0], x[0{,}1], \ldots, x[0, N-1], x[1{,}0], \ldots, x[N-1, N-1]\}^{\mathrm{T}}$$

- Entries of basis images $\boldsymbol{a}_{kl}$ are ordered accordingly into $N^2 \times N^2$ matrix $\boldsymbol{B}$

- Two-dimensional transform can be expressed as

$$\boldsymbol{y} = \boldsymbol{B}\boldsymbol{x}$$

- If transform is separable, then transform $\boldsymbol{B}$ can be constructed from $\boldsymbol{A}$ via

$$\boldsymbol{B} = \boldsymbol{A} \otimes \boldsymbol{A} = \begin{bmatrix} a_{0,0}\boldsymbol{A} & a_{0,1}\boldsymbol{A} & \cdots & a_{0,N-1}\boldsymbol{A} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N-1,0}\boldsymbol{A} & a_{N-1,1}\boldsymbol{A} & \cdots & a_{N-1,N-1}\boldsymbol{A} \end{bmatrix}$$

Generalization: any arbitrary one-dimensional transform is called separable if it can be decomposed into a Kronecker product of two transforms

Properties of Orthonormal Transforms

Energy conservation

- Orthonormal transforms conserve signal energy, i.e. the length of the input vector in the N-dimensional space does not change by transform

$$\|\boldsymbol{y}\|_2^2 = \sum_{k=0}^{N-1} |y[k]|^2 = \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} = (\boldsymbol{A}\boldsymbol{x})^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}$$

$$= \boldsymbol{x}^{\mathrm{T}} \underbrace{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}}_{=\boldsymbol{I}} \boldsymbol{x} = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = \sum_{n=0}^{N-1} |x[n]|^2 = \|\boldsymbol{x}\|_2^2$$

- For two-dimensional image transforms we have

$$\sum_{m=0}^{N-1} \sum_{n=0}^{N-1} |x[m,n]|^2 = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} |y[k,l]|^2$$

Rotation

- Orthonormal transforms can be regarded as rotation of input vector $\boldsymbol{x}$ in an N-dimensional space
- Equivalently, the transform is a rotation of the basis coordinates and the components of $\boldsymbol{y}$ are the projections of $\boldsymbol{x}$ on the new basis vectors

7.3 Karhunen Loève Transform (KLT)

Optimum transform introduced as series expansion for continuous random processes by Karhunen and Loeve

- Discrete equivalent for random sequences, also called *method of principal components*, developed by Hotelling

Given: random vector $\boldsymbol{x}$ and its covariance matrix $\boldsymbol{\Psi}_{xx}$ with

$$\boldsymbol{\Psi}_{xx} = \mathrm{E}\{(\boldsymbol{x}-\boldsymbol{\mu}_x)(\boldsymbol{x}-\boldsymbol{\mu}_x)^{\mathrm{T}})\} = \begin{bmatrix} \sigma_x^2 & \psi_{xx}[1] & \cdots & \psi_{xx}[N-1] \\ \psi_{xx}[1] & \sigma_x^2 & \cdots & \psi_{xx}[N-2] \\ \vdots & \vdots & \ddots & \vdots \\ \psi_{xx}[N-1] & \psi_{xx}[N-2] & \cdots & \sigma_x^2 \end{bmatrix}$$

- Basis vectors of KLT are given by the orthonormalized eigenvectors of $\boldsymbol{\Psi}_{xx}$

$$\boldsymbol{\Psi}_{xx}\boldsymbol{v}_k = \lambda_k \boldsymbol{v}_k, \quad 0 \le k \le N-1$$

- KLT of $\boldsymbol{x}$ is defined as

$$\boldsymbol{y} = \boldsymbol{V}\boldsymbol{x}, \quad \text{with} \quad \boldsymbol{V} = [\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{N-1}]^{\mathrm{T}}$$

Properties of KL Transform

Orthogonality

- Covariance matrix $\boldsymbol{\Psi}_{xx}$ is symmetric and positive semidefinite
- Eigenvalues λ_k are real and non-negative, eigenvectors are orthogonal

Inverse KLT: due to orthogonality we have

$$\boldsymbol{x} = \boldsymbol{V}^{\mathrm{T}}\boldsymbol{y} = \sum_{k=0}^{N-1} y[k]\boldsymbol{v}_k$$

Decorrelation: since $\boldsymbol{\Psi}_{xx}$ is Hermitian (complex symmetric) we have

$$\boldsymbol{\Psi}_{yy} = \mathrm{E}\{(\boldsymbol{y}-\boldsymbol{\mu}_{\boldsymbol{y}})(\boldsymbol{y}-\boldsymbol{\mu}_{\boldsymbol{y}})^{\mathrm{T}}\} = \boldsymbol{V}\boldsymbol{\Psi}_{xx}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{\Lambda} = \mathrm{Diag}\{\lambda_k\}$$

- Off-diagonal elements of $\boldsymbol{\Psi}_{yy}$ are zero, the diagonal elements represent the coefficient variances

$$\boldsymbol{\Psi}_{yy} = \boldsymbol{\Lambda} = \mathrm{Diag}\left\{\sigma^2_{y[0]}, \sigma^2_{y[1]}, \ldots, \sigma^2_{y[N-1]}\right\}$$

- KLT coefficients $\{y[k], k = 0, \ldots, N-1\}$ are uncorrelated

Basis Restriction Error of KL Transform

Restriction: vector $\boldsymbol{x}$ is transformed to $\boldsymbol{y}$, elements of $\hat{\boldsymbol{y}}$ are chosen to be the first L elements of $y[k]$ and zeros elsewhere

$$\hat{y}[k] = \begin{cases} y[k], & 0 \leq k \leq L-1 \\ 0, & \text{else} \end{cases}$$

Mean square error between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ is called *basis restriction error*

$$J_L = \frac{1}{N}\sum_{n=0}^{N-1} |x[n] - \hat{x}[n]|^2 = \frac{1}{N}\mathrm{Tr}\left\{(\boldsymbol{x} - \hat{\boldsymbol{x}})(\boldsymbol{x} - \hat{\boldsymbol{x}})^{\mathrm{T}}\right\}$$

KL transform: expected value of error J_L is minimum for every value of $L \in [1, N]$ when the transform is selected to

$$\boldsymbol{A} = \boldsymbol{V}, \qquad \boldsymbol{B} = \boldsymbol{V}^{\mathrm{T}}$$

where rows of $\boldsymbol{V}$ are arranged in decreasing order of the eigenvalues of $\boldsymbol{\Psi}_{xx}$

Equivalence: KLT packs the maximum average energy in any $L \leq N$ samples of $\boldsymbol{y}$

7.4 Discrete Cosine Transform (DCT)

Definition of DCT Type II for block of $N \times N$ pixels

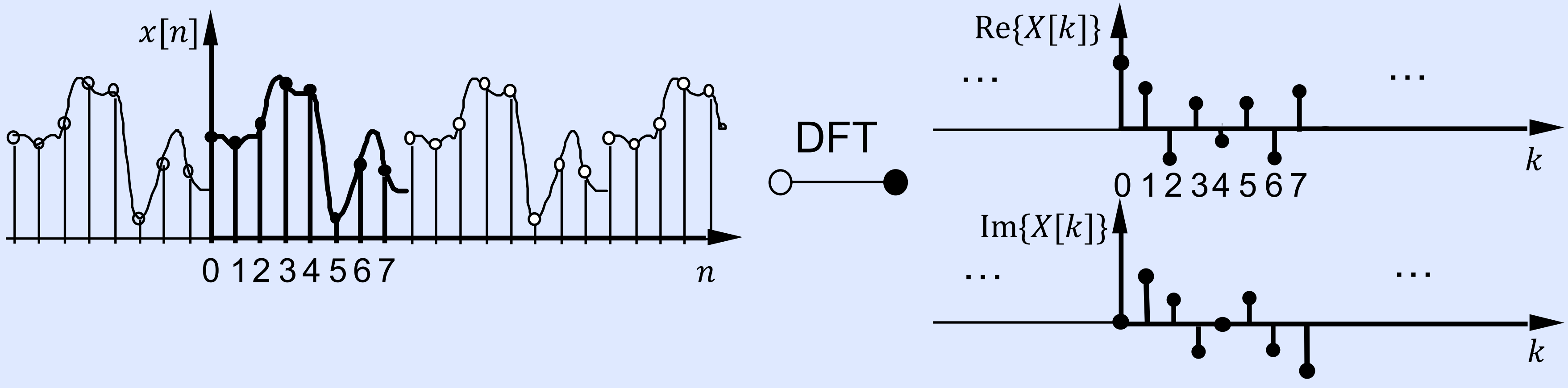
Transform matrix $\boldsymbol{A} = (\boldsymbol{a}_{\boldsymbol{0}}, \ldots, \boldsymbol{a}_{\boldsymbol{N-1}})^{\mathrm{T}}$ with $a_k[n] = \gamma_k \cos \frac{\pi(2n+1)k}{2N}$

$$n, k = 0, \ldots, N-1$$

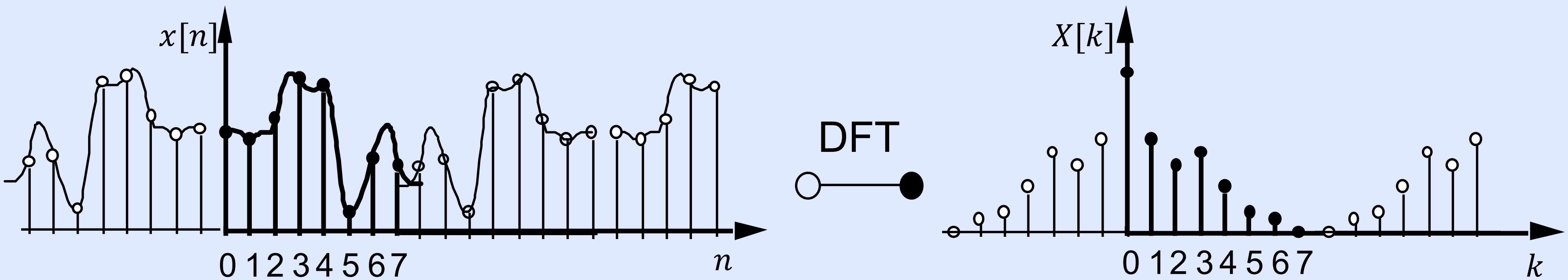
Scale factor $\gamma_i = \begin{cases} \sqrt{\dfrac{1}{N}} & \text{for } k = 0 \\ \sqrt{\dfrac{2}{N}} & \text{for } k \neq 0 \end{cases}$

- The equation gives the ideal functional definition of the DCT
- For practical implementations accuracy requirements are specified
- Compression standards don't specify any special DCT algorithm

Discrete Cosine Transform Versus DFT



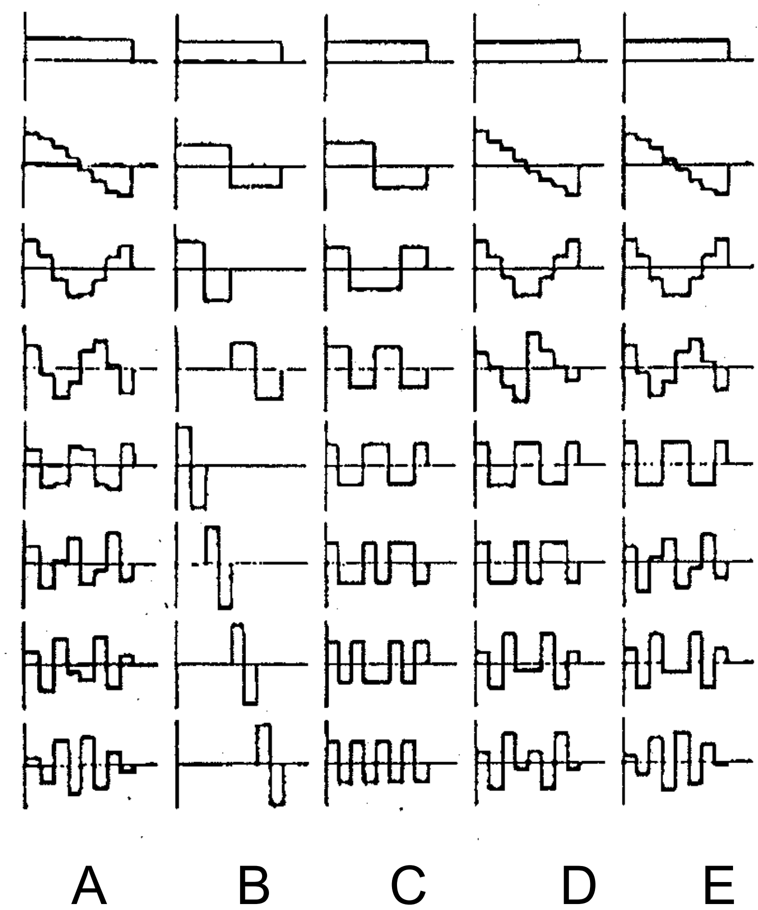
Discrete Fourier Transform: Repeat sampled data infinitely and perform Fourier Transform



Discrete Cosine Transform: Repeat *mirrored* data to force an even image function and perform Fourier Transform: imaginary coefficients (sine) disappear.

Basis Functions of Various 1D Transforms

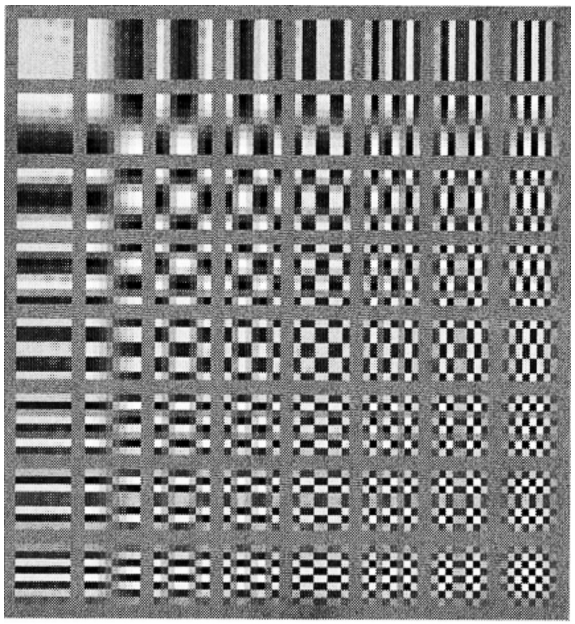
One-dimensional basis functions for block size $N = 8$



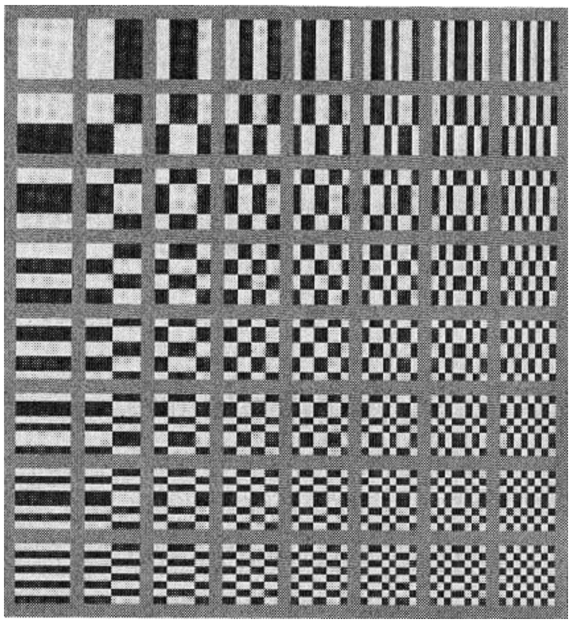
A - Karhunen Loève Transform (KLT)
B - Haar Transform (HT)
C - Walsh Hadamard Transform (WHT)
D - Slant Transform
E - Discrete Cosine Transform (DCT)

Basis Functions of Various 2D Transforms

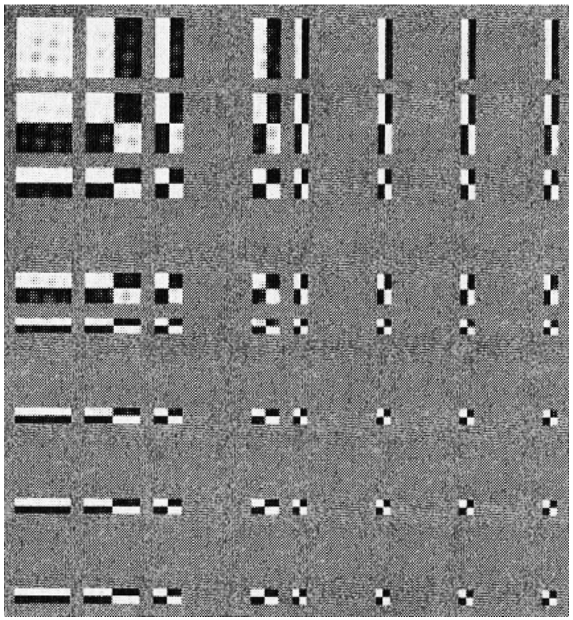
Two-dimensional basis functions for block size of 8×8 pixels



Discrete Cosine
Transform



Walsh Hadamard
Transform

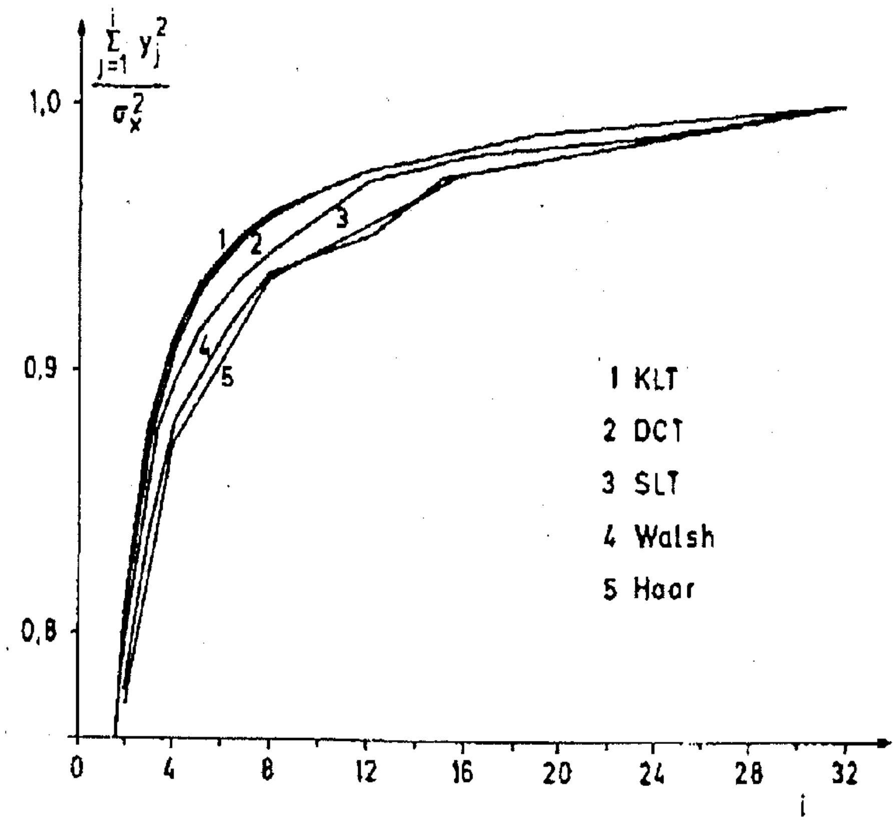


Haar
Transform

Energy Compaction of Various Transforms

Criteria for selecting a particular transform:

- decorrelation, energy concentration
- perceptual pleasant basis functions
- low computational complexity



Energy concentration for typical natural images (block size 1×32 pixels)

- KLT is optimum
- DCT performs slightly worse than KLT

Properties of Some Image Transforms

Karhunen Loève Transform (KLT): yields decorrelated coefficients by

- decomposition of signal into eigenvectors of covariance matrix
- has optimum energy concentration
- is dependent on signal statistics, computationally complex

Discrete Cosine Transform (DCT): achieves decorrelation close to KLT

- DCT is symmetrical extended DFT, has only real coefficients
- fast algorithms and special hardware available

Walsh Hadamard Transform (WHT): basis functions have form of squares

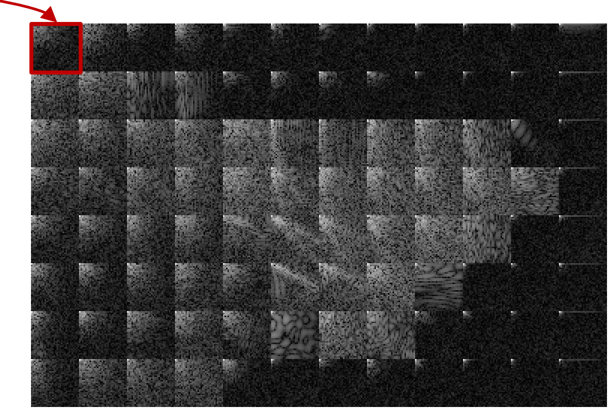
- low computational complexity as only additions are used

Haar Transform (HT): frequency bands have unequal width and spacing

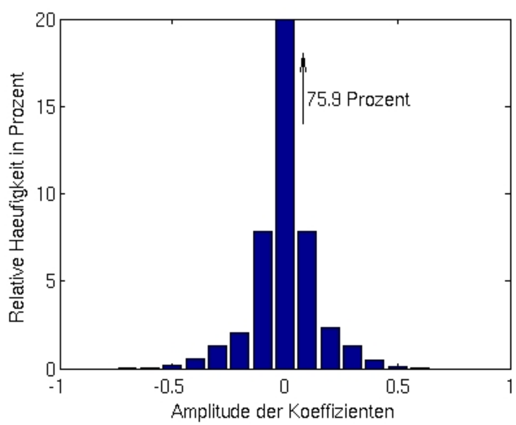
- simple version of wavelet transform

DCT Example

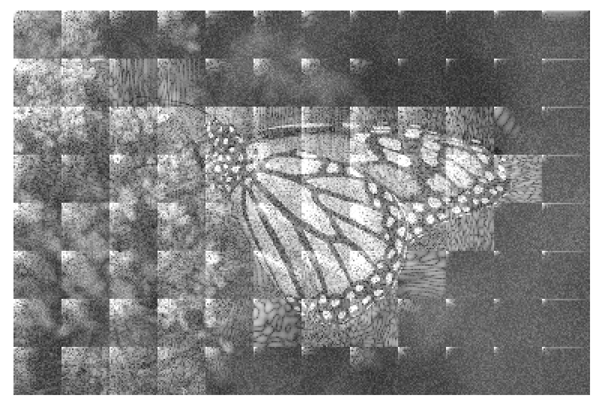
Original



DCT coefficients



Histogram of DCT coefficients



Coefficients overlaid on original

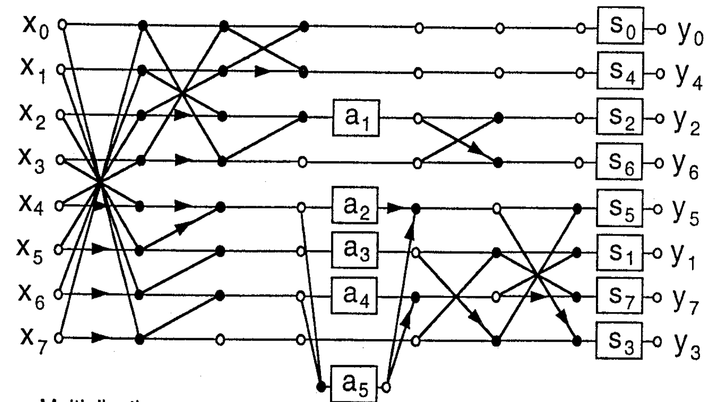
Fast Implementation of the DCT

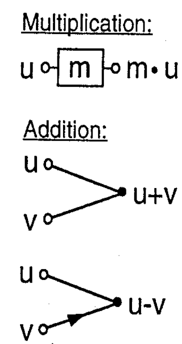
$$
\begin{aligned}
\underline{y} &= \underline{M}\,\underline{x} \\
&= \underline{S}\,\underline{P}\,\underline{M}_1\underline{M}_2\underline{M}_3\underline{M}_4\underline{M}_5\underline{M}_6\,\underline{x}
\end{aligned}
$$

$$
\underline{S} = \begin{bmatrix} s_0 &&&&&&& \\ & s_1 &&&&&& \\ && s_2 &&&&& \\ &&& s_3 &&&& \\ &&&& s_4 &&& \\ &&&&& s_5 && \\ &&&&&& s_6 & \\ &&&&&&& s_7 \end{bmatrix}
\quad
\underline{P} = \begin{bmatrix} 1 &&&&&&& \\ &&&&& 1 && \\ && 1 &&&&& \\ &&&&&&& 1 \\ & 1 &&&&&& \\ &&&& 1 &&& \\ &&& 1 &&&& \\ &&&&&& 1 & \end{bmatrix}
$$

$$
\underline{M}_1 = \begin{bmatrix} 1 &&&&&&& \\ & 1 &&&&&& \\ && 1 &&&&& \\ &&& 1 &&&& \\ &&&& 1 &&& 1 \\ &&&&& 1 & 1 & \\ &&&&& 1 & -1 & \\ &&&& -1 &&& 1 \end{bmatrix}
\quad
\underline{M}_2 = \begin{bmatrix} 1 &&&&&&& \\ & 1 &&&&&& \\ && 1 & 1 &&&& \\ && -1 & 1 &&&& \\ &&&& 1 &&& \\ &&&&& 1 && 1 \\ &&&&&& 1 & \\ &&&&& -1 && 1 \end{bmatrix}
\quad
\underline{M}_3 = \begin{bmatrix} 1 &&&&&&& \\ & 1 &&&&&& \\ && C_4 &&&&& \\ &&& 1 &&&& \\ &&&& -C_2 && -C_6 & \\ &&&&& C_4 && \\ &&&& -C_6 && C_2 & \\ &&&&&&& 1 \end{bmatrix}
$$

$$
\underline{M}_4 = \begin{bmatrix} 1 & 1 &&&&&& \\ 1 & -1 &&&&&& \\ && 1 & 1 &&&& \\ &&& 1 &&&& \\ &&&& 1 &&& \\ &&&&& 1 && \\ &&&&&& 1 & \\ &&&&&&& 1 \end{bmatrix}
\quad
\underline{M}_5 = \begin{bmatrix} 1 &&& 1 &&&& \\ & 1 & 1 &&&&& \\ & 1 & -1 &&&&& \\ 1 &&& -1 &&&& \\ &&&& -1 & -1 && \\ &&&&& 1 & 1 & \\ &&&&&& 1 & 1 \\ &&&&&&& 1 \end{bmatrix}
\quad
\underline{M}_6 = \begin{bmatrix} 1 &&&&&&& 1 \\ & 1 &&&&& 1 & \\ && 1 &&& 1 && \\ &&& 1 & 1 &&& \\ &&& 1 & -1 &&& \\ && 1 &&& -1 && \\ & 1 &&&&& -1 & \\ 1 &&&&&&& -1 \end{bmatrix}
$$





$a_1 = C_4$
$a_2 = C_2 - C_6$
$a_3 = C_4$
$a_4 = C_6 + C_2$
$a_5 = C_6$

$s_0 = \dfrac{1}{2\sqrt{2}}$

$s_k = \dfrac{1}{4\,C_k}$; $k = 1 \ldots 7$

$C_k = \cos(k\pi/16)$

- Number of multiplications:

Fast algorithm:	5 + 8	
Direct matrix multiplication:	64	*Scaling*

Factorization of DCT into sparse matrices [Arai, Agui, Nakajima, 1988]

Signal flow graph for fast DCT

7.5 Bit Allocation for Transform Coding

Coding problem: Distribute a limited number of bits among N transform coefficients such that the resulting distortion d is minimized

High-rate distortion-rate function for image pixels x

$$d(R) \cong \varepsilon^2 \sigma_X^2 2^{-2R}$$

- Distortion-rate function for coding transform coefficients:

$$d_{\mathrm{XFORM}}(R) = \frac{1}{N} \sum_{n=0}^{N-1} d_n(R_n) \cong \frac{1}{N} \sum_{n=0}^{N-1} \varepsilon^2 \sigma_{Y_n}^2 2^{-2R_n}$$

⇨ Minimize $d_{\mathrm{XFORM}}(R)$ subject to rate constraint $R = \frac{1}{N} \sum_{n=0}^{N-1} R_n$

Optimum bit allocation can be found using Lagrangian formulation

$$\underset{R_0, R_1, \ldots, R_{N-1}}{\operatorname{argmin}} \; (J) \quad \text{with} \quad J = d_{\mathrm{XFORM}}(R) + \lambda R$$

and setting the partial derivatives equal to zero:

$$\frac{\partial J}{\partial R_n} = 0 \quad \forall n$$

Transform Coding Gain

Optimum bit allocation is achieved as result of equating $\mathrm{d}J/\mathrm{d}R_n = 0$ if

- distortion is equal for all transform coefficients

$$d_n(R_n) = d_{\mathrm{XFORM}}(R) \quad \forall n$$

- rate per coefficient is proportional to coefficient variance

$$R_n = \frac{1}{2}\log_2 \frac{\varepsilon^2 \sigma_{Y_n}^2}{d_{\mathrm{XFORM}}} \quad \forall n$$

Transform coding gain defined as

$$G_{\mathrm{XFORM}} = \frac{d(R)}{d_{\mathrm{XFORM}}(R)} = \frac{\sigma_X^2}{\sqrt[N]{\prod_{n=0}^{N-1} \sigma_{Y_n}^2}} = \frac{\frac{1}{N}\sum_{n=0}^{N-1} \sigma_{Y_n}^2}{\sqrt[N]{\prod_{n=0}^{N-1} \sigma_{Y_n}^2}}$$

- Coding gain increases if variances of transform coefficients are unequally distributed
- No gain in case all transform coefficient variances are (almost) equal
- Same value ε assumed for all distortion functions

Zonal Coding

- Fixed assignment of bits for each transform coefficient
- Discard predefined subset of coefficients by allocating zero bits

Example: fixed bit allocation for 8×8 DCT transformed image block

- 8 bit quantizer for DC coefficient
- Decreasing number of quantization levels (7 bit or less) for higher frequency AC coefficients

8	7	6	5	4	3	2	1
7	6	5	4	3	2	1	0
6	5	4	3	2	1	0	0
5	4	3	2	1	0	0	0
4	3	2	1	0	0	0	0
3	2	1	0	0	0	0	0
2	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0

Average bit rate for this quantization table: 1.875 bit/pixel

Disadvantage: zonal coding is not adaptive to image data, high frequencies will always be discarded

🗁 Demo 7 „Zonal DCT Coding“

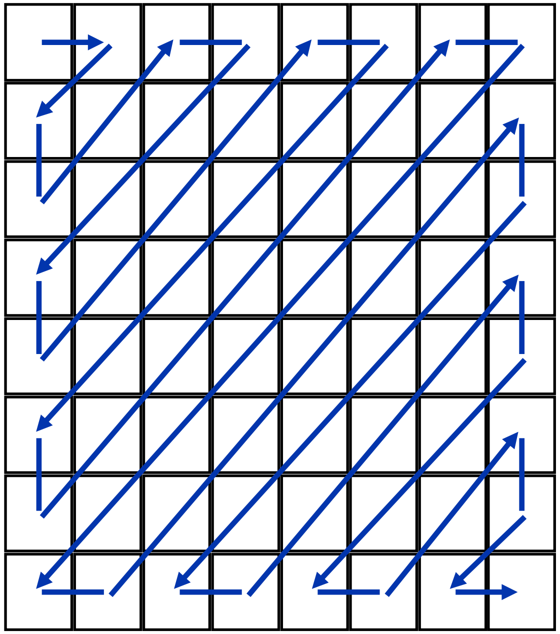
Threshold Coding

Uniform deadzone quantizer

- Transform Coefficients that fall below a threshold are discarded

Run-level coefficient coding

- Positions of non-zero transform coefficients are coded together with amplitude values using zigzag scan order



Zigzag scan ordering of coefficients

Example for Threshold Coding

Original 8×8 block

136	142	151	165	180	196	201	210
134	142	152	165	174	195	202	209
131	139	156	163	176	190	200	210
132	141	150	156	172	189	197	214
133	139	146	158	168	187	200	214
133	137	144	157	167	186	202	211
127	136	144	159	166	187	203	208
130	139	146	158	165	185	197	210

DCT transformed block

1352	-216	16	-11	-4	-1	4	-7
18	3	-10	4	0	-3	4	2
3	1	-5	3	-1	2	-2	-3
1	-2	3	5	2	2	-3	-1
0	2	3	0	3	-5	2	-1
-1	-2	-2	1	0	-2	0	2
1	2	-1	-1	2	-1	1	3
1	-1	-3	-1	0	2	-1	1

Quantization, $\Delta = 15$

90	-14	1	-1	0	0	0	0
1	0	-1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

```
Mean value: 90
(0,-14) (0,1) (2,1)
(0,-1) (0,-1) (EOB)
```

Entropy coding & transmission

```
Mean value: 90
(0,-14) (0,1) (2,1)
(0,-1) (0,-1) (EOB)
```

90	-14	1	-1	0	0	0	0
1	0	-1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Scaled decoded coefficients

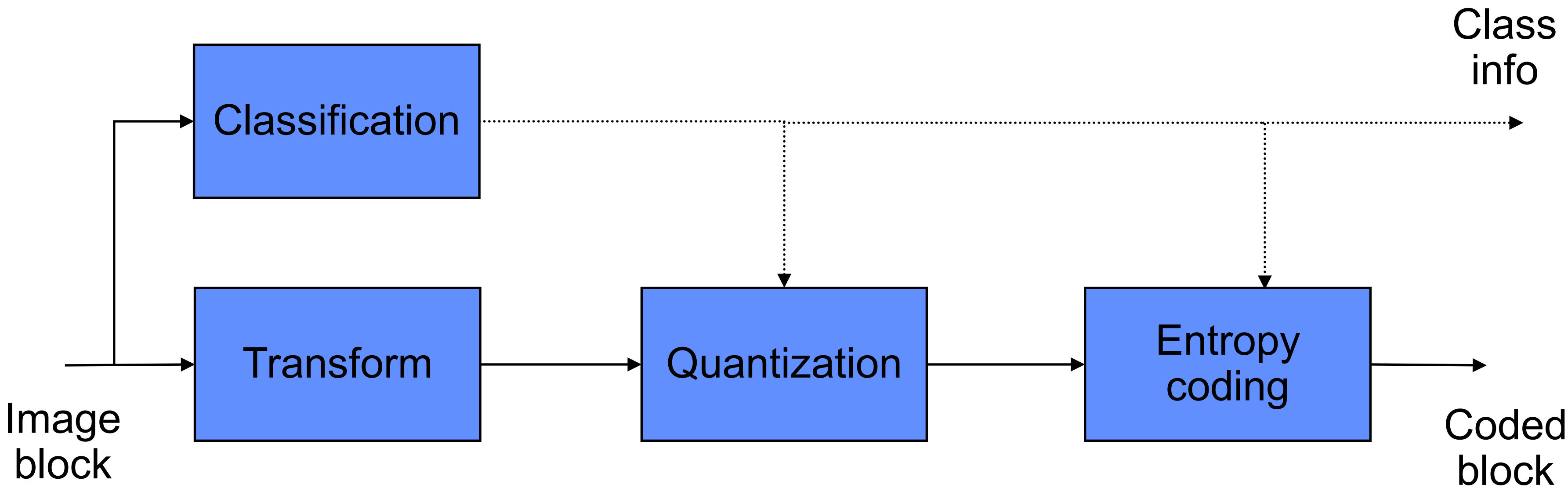
1350	-210	15	-15	0	0	0	0
15	0	-15	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Reconstructed 8×8 block

132	141	154	167	178	190	201	209
132	140	153	166	177	189	201	209
132	140	152	164	175	188	201	209
132	140	151	162	173	187	200	210
133	139	149	159	171	185	200	210
133	139	147	157	169	183	199	210
133	138	146	155	167	182	199	210
133	138	146	155	166	182	199	211

Adaptive Transform Coding

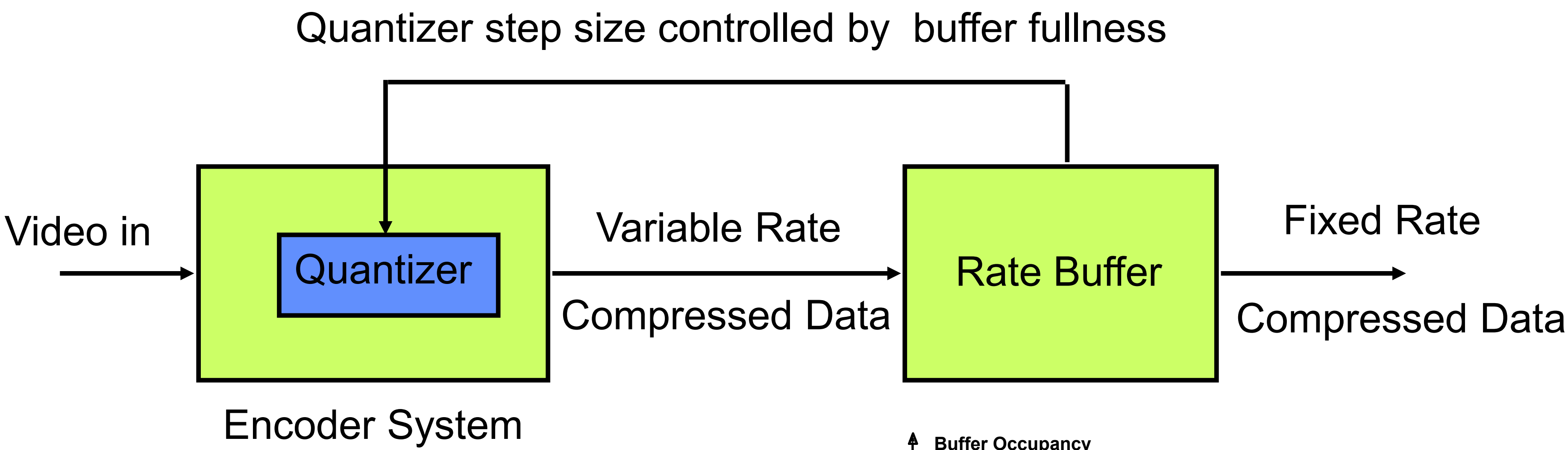
Idea: switch between different quantizers and corresponding variable length code tables depending on characteristics of image signal

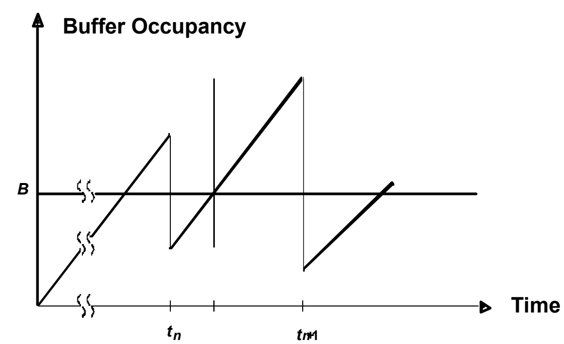


Classification considers instationarity of image signal by assigning blocks as

- little detail
- horizontal structures
- vertical structures
- highly textured areas

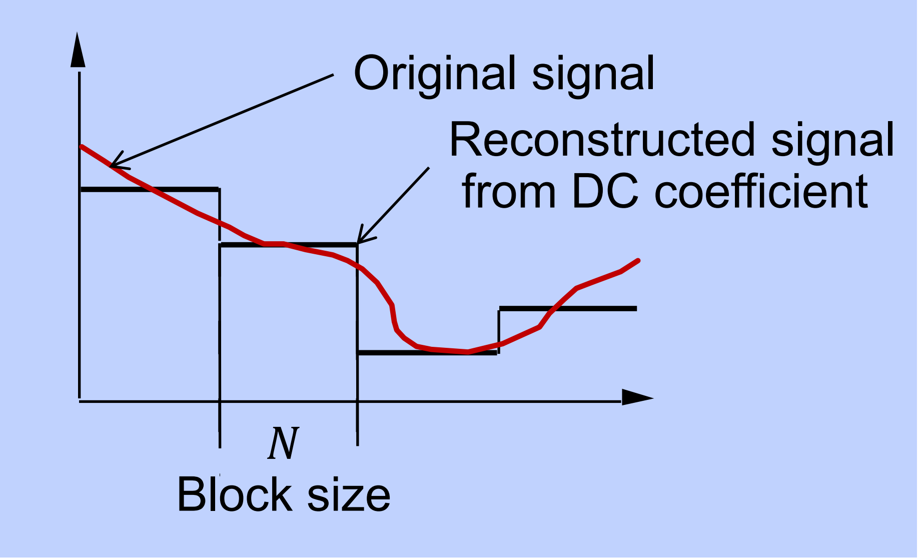
Rate Control at Encoder





7.6 Compression Artifacts

Blocking: visibility of the borders of coding blocks caused by coarse quantization of low detail regions, most common artifact of transform coding



Blurring: loss of spatial detail (sharpness) in structured image areas if high frequency components are coarsely quantized

Ringing: periodic structures (basis functions) visible at edges on flat background if only small number of frequencies are coded

Examples for Compression Artifacts

Blocking

1:40

Blurring

1:20

Ringing

1:12

Transform Coding - Summary

- Transform gives spectral decomposition of image signal
- Transform coefficients are weighting factors for basis functions
- Orthonormal transform corresponds to rotation of coordinate system
- Transform causes energy concentration on few coefficients
- KLT is optimum transform but signal dependent
- DCT performs close to KLT and has fast implementation
- Typical block sizes for image transforms are from 8×8 to 32×32 pixels
- Bit allocation proportional to logarithm of variance
- Threshold coding and zigzag scanning widely used
- Quantization of coefficients use psychovisual properties