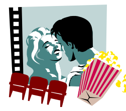
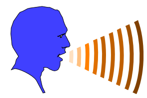
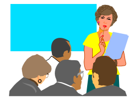
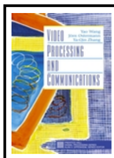
   

Video Processing & Communications

Waveform-Based Coding: Transform and Predictive Coding

Yao Wang
Polytechnic University, Brooklyn, NY11201
http://eeweb.poly.edu/~yao

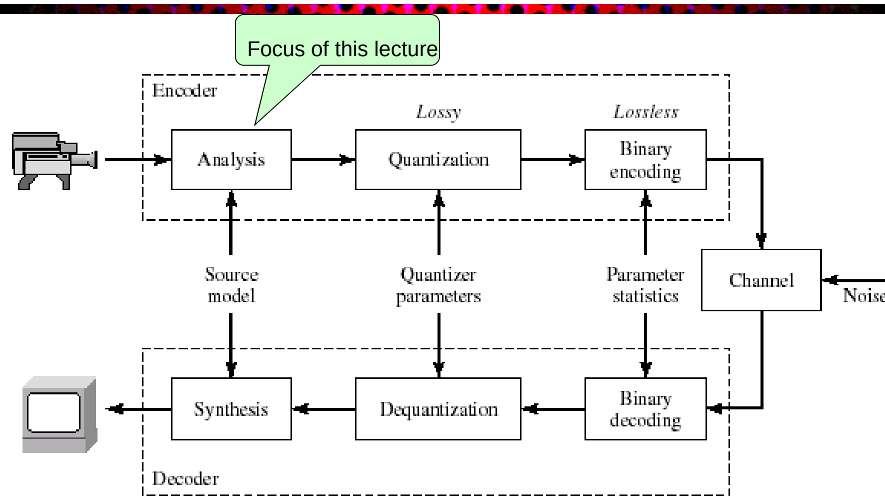
Based on: Y. Wang, J. Ostermann, and Y.-Q. Zhang, Video Processing and Communications, Prentice Hall, 2002.



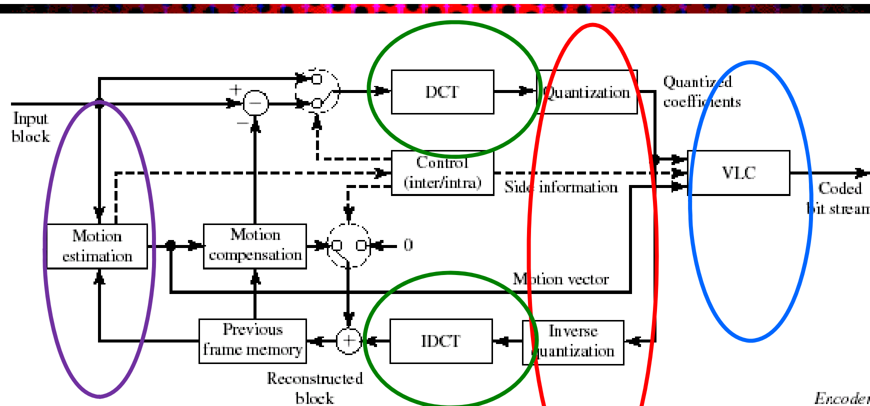
Outline

- Overview of video coding systems
- Transform coding
- Predictive coding

Components in a Coding System



Encoder Block Diagram of a Typical Block-Based Video Coder (Assuming No Intra Prediction)



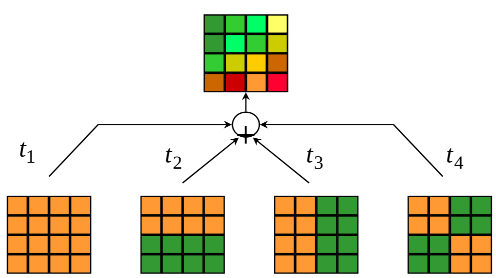
Lectures 3&4: Motion estimation
Lecture 5: Variable Length Coding
Last lecture: Scalar and Vector Quantization
This lecture: DCT, wavelet and predictive coding

A Review of Vector Quantization

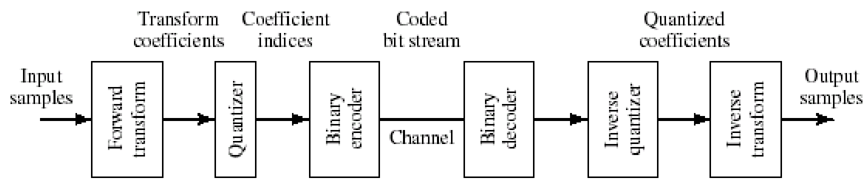
- Motivation: quantize a group of samples (a vector) together, to exploit the correlation between samples
- Each sample vector is replaced by one of the representative vectors (or patterns) that often occur in the signal
- Typically a block of 4x4 pixels
- Design is limited by ability to obtain training samples
- Implementation is limited by large number of nearest neighbor comparisons – exponential in the block size

Transform Coding

- Motivation:
  - Represent a vector (e.g. a block of image samples) as the superposition of some typical vectors (block patterns)
  - Quantize and code the coefficients
  - Can be thought of as a constrained vector quantizer



Block Diagram



General Linear Transform

- Basis vectors (or blocks):

$$[\mathbf{U}] = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N]$$

- Inverse transform represents a vector or block as the superposition of basis vectors or blocks

$$\text{inverse transform:} \quad \mathbf{s} = \sum_{k \in \mathcal{N}} t_k \mathbf{u}_k = [\mathbf{U}]\mathbf{t}$$

- Forward transform determines the contribution (weight) of each basis vector

$$\text{forward transform:} \quad \mathbf{t} = [\mathbf{U}]^{-1}\mathbf{s} = [\mathbf{V}]\mathbf{s}$$

Unitary Transform

- Unitary (orthonormal) basis:
  – Basis vectors are orthogonal to each other and each has length 1

$$\langle \mathbf{u}_k, \mathbf{u}_l \rangle = \sum_{n \in \mathcal{N}} u^*_{k;n} u_{l;n} = \delta_{k,l} = \begin{cases} 1 & \text{if } k = l, \\ 0 & \text{if } k \neq l, \end{cases}$$

$$[\mathbf{U}]^H [\mathbf{U}] = [\mathbf{U}][\mathbf{U}]^H = [\mathbf{I}]_N$$

- Transform coefficient associated with a basis vector is simply the projection of the input vector onto the basis vector

$$\text{forward transform:} \quad t_k = \langle \mathbf{u}_k, \mathbf{s} \rangle \quad \text{or} \quad \mathbf{t} = [\mathbf{U}]^H \mathbf{s} = [\mathbf{V}]\mathbf{s}$$

$$\text{inverse transform:} \quad \mathbf{s} = \sum_{k \in \mathcal{N}} t_k \mathbf{u}_k = [\mathbf{U}]\mathbf{t} = [\mathbf{V}]^H \mathbf{t}.$$

Transform design

- What are desirable properties of a transform for image and video?
  – Nearly decorrelating – improves efficiency of scalar quantizer
  – High energy compaction – a few large coefficients to send
  – Easy to compute (few operations)
  – Separable – compute 1-D transform first on rows, then on columns
- What size transform should we use?
  – Entire image? Small?
  – 2-D (on an image) or 3-D (incorporating time also)?
- From Amy Reibman

Karhunen Loève Transform (KLT)

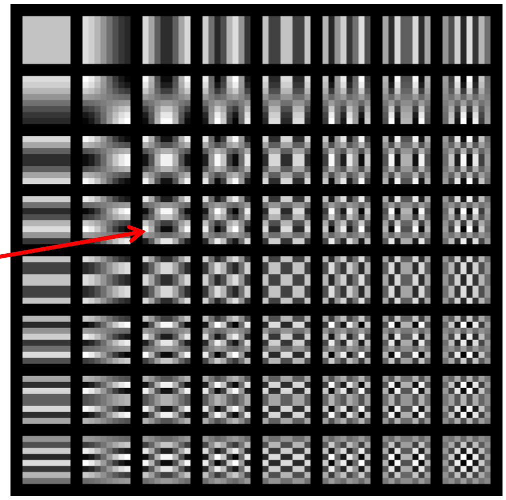
- Optimal transform
- Requires statistics of the input source
  - Known covariance function
- Coefficients are completely uncorrelated
- The best energy compaction
  - Sort coefficients from largest to smallest expected squared magnitude; then the sum of the energies of the first M coefficients is as large as possible
- No computationally efficient algorithm
- We'll derive it later

- From Amy Reibman

Other Transform Bases

- Suboptimal transforms – many available!
  - Discrete Fourier Transform (DFT): complex values; discontinuities
  - Discrete Cosine transform (DCT): nearly as good as KLT for common image signals
  - Hadamard and Haar: basis functions contain only +1,0,-1

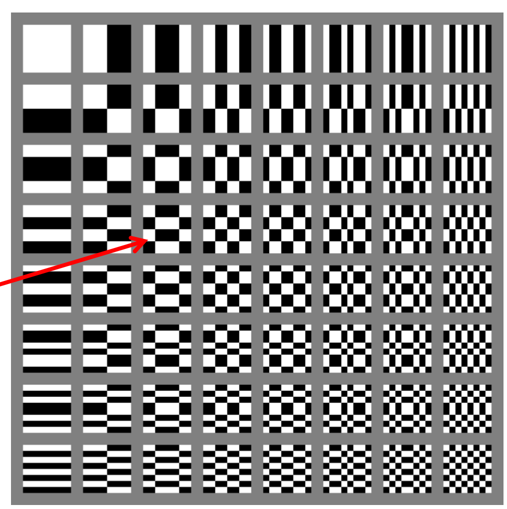
Discrete Cosine Transform: Basis Images



Example:

```
D=dctmtx(8);
X=zeros(8);
X(4,3)=1;
Basis=D'*X*D;
```


Hadamard Transform: Basis images

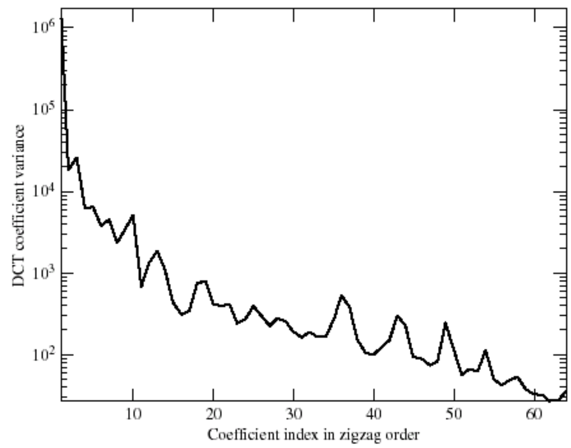


Example:

```
D=hadamard(8);
reindex=[1,8,4,5,2,7,3,6];
D(reindex,:)=D;
X=zeros(8);
X(4,3)=1;
Basis=D'*X*D;
```

From Amy Reibman

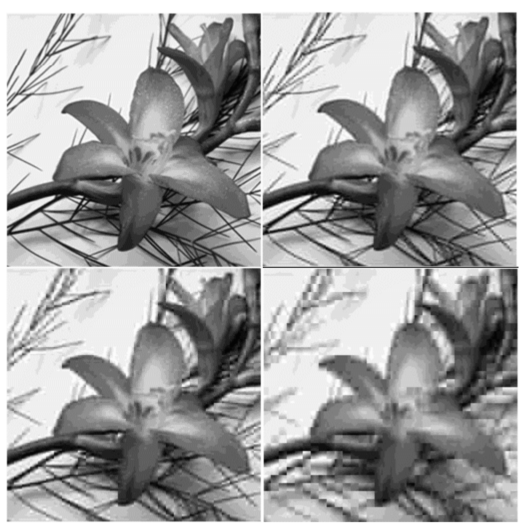
Energy Distribution of DCT Coefficients in Typical Images



Images Approximated by Different Number of DCT Coefficients

Original

With 16/64 Coefficients



With 8/64 Coefficients

With 4/64 Coefficients

Demos

- Use matlab demo to demonstrate approximation using different number of DCT coefficients

(dctdemo.m)

Distortion in Transform Coding

- Distortion in sample (image) domain

$$D_s = \frac{1}{N} E\{\|\mathcal{S} - \hat{\mathcal{S}}\|^2\} = \frac{1}{N} \sum_{n \in \mathcal{N}} D_{s,n} \quad D_{s,n} = E\{(\mathcal{S}_n - \hat{\mathcal{S}}_n)^2\}.$$

- Distortion in coefficient (transform) domain

$$D_t = \frac{1}{N} E\{\|\mathcal{T} - \hat{\mathcal{T}}\|^2\} = \frac{1}{N} \sum_{k \in \mathcal{N}} D_{t,k} \quad D_{t,k} = E\{(\mathcal{T}_k - \hat{\mathcal{T}}_k)^2\}.$$

- With a unitary transform, the two distortions are equal

$$D_s = \frac{1}{N} E\{\|\mathcal{S} - \hat{\mathcal{S}}\|^2\} = \frac{1}{N} E\{\|[\mathbf{V}]^H (\mathcal{T} - \hat{\mathcal{T}})\|^2\}$$

$$= \frac{1}{N} E\{(\mathcal{T} - \hat{\mathcal{T}})^H [\mathbf{V}][\mathbf{V}]^H (\mathcal{T} - \hat{\mathcal{T}})\} = \frac{1}{N} E\{\|\mathcal{T} - \hat{\mathcal{T}}\|^2\} = D_t,$$

Modeling of Distortion Due to Coefficient Quantization

- High Resolution Approximation of Scalar Quantization
  - With the MMSE quantizer, when each coefficient is scalar quantized with sufficient high rates, so that the pdf in each quantization bin is approximately flat

One coefficient
$$D_{t,k}(R_k) = \epsilon^2_{t,k}\sigma^2_{t,k}2^{-2R_k}$$

Average over all coefficients
$$D_{\mathrm{TC}} = D_s = D_t = \frac{1}{N}\sum_{k\in\mathcal{N}} \epsilon^2_{t,k}\sigma^2_{t,k}2^{-2R_k}.$$

$\epsilon^2_{t,k}$ Depends on the pdf of the k-th coefficient.

Optimal Bit Allocation Among Coefficients

- How Many Bits to Use For Each Coefficient?
  - Can be formulated as an constrained optimization problem:

Minimize:
$$D_{\mathrm{TC}} = D_s = D_t = \frac{1}{N}\sum_{k\in\mathcal{N}} \epsilon^2_{t,k}\sigma^2_{t,k}2^{-2R_k}.$$

Subject to:
$$\sum_{k\in\mathcal{N}} R_k = RN$$

  - The constrained problem can be converted to unconstrained one using the Lagrange multiplier method

Minimize:
$$J(R_k, \forall k\in\mathcal{N}) = \sum_{k\in\mathcal{N}} \epsilon^2_{t,k}\sigma^2_{t,k}2^{-2R_k} + \lambda\left(\sum_{k\in\mathcal{N}} R_k - RN\right)$$

Derivation and Result

If we let $(\partial J / \partial R_k) = 0$, we obtain

$$\frac{\partial D_{t,k}}{\partial R_k} = -2 \ln 2 D_{t,k} = -(2 \ln 2)\epsilon_{t,k}^2 \sigma_{t,k}^2 2^{-2R_k} = -\lambda, \quad \forall k \in \mathcal{N}$$

Multiply to obtain:

$$\lambda^N = (2 \ln 2)^N \left(\prod_k \epsilon_{t,k}^2 \sigma_{t,k}^2\right) 2^{-2\sum_k R_k} = (2 \ln 2)^N \left(\prod_k \epsilon_{t,k}^2 \sigma_{t,k}^2\right) 2^{-2NR}$$

$$\lambda = (2 \ln 2) \left(\prod_k \epsilon_{t,k}^2 \sigma_{t,k}^2\right)^{1/N} 2^{-2R}.$$

Substitute into first equation:

$$R_k = R + \frac{1}{2} \log_2 \frac{\epsilon_{t,k}^2 \sigma_{t,k}^2}{\left(\prod_k \epsilon_{t,k}^2 \sigma_{t,k}^2\right)^{1/N}}.$$

Result: all distortions are equal!

$$D_{\mathrm{TC}} = D_t = D_{t,k} = \left(\prod_k \epsilon_{t,k}^2 \sigma_{t,k}^2\right)^{1/N} 2^{-2R}.$$

Implication of Optimal Bit Allocation

- Bit rate for a coefficient proportional to its variance (energy)

$$R_k = R + \frac{1}{2} \log_2 \frac{\epsilon_{t,k}^2 \sigma_{t,k}^2}{\left(\prod_k \epsilon_{t,k}^2 \sigma_{t,k}^2\right)^{1/N}}.$$

Geometric mean

- Distortion is equalized among all coefficients and depends on the geometric mean of the coefficient variances

$$D_{\mathrm{TC}} = D_t = D_{t,k} = \left(\prod_k \epsilon_{t,k}^2 \sigma_{t,k}^2\right)^{1/N} 2^{-2R}.$$

Transform Coding Gain Over PCM

- PCM: quantize each sample in the image domain directly
- Distortion for PCM if each sampled is quantized to R bit:

$$D_{\mathrm{PCM}} = D_{s,n} = \epsilon_s^2 \sigma_s^2 2^{-2R}$$

- Gain over PCM: $G_{\mathrm{TC}} = \frac{D_{\mathrm{PCM}}}{D_{\mathrm{TC}}}.$

$$G_{\mathrm{TC}} = \frac{\epsilon_s^2 \sigma_s^2}{\left(\prod_k \epsilon_{t,k}^2 \sigma_{t,k}^2\right)^{1/N}} = \frac{\epsilon_s^2}{\left(\prod_k \epsilon_{t,k}^2\right)^{1/N}} \frac{\frac{1}{N}\sum \sigma_{t,k}^2}{\left(\prod_k \sigma_{t,k}^2\right)^{1/N}},$$

Arithmetic mean

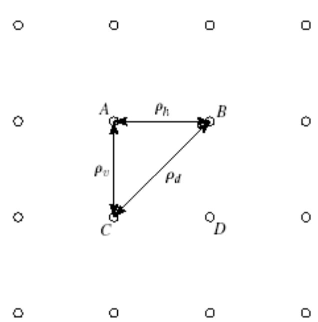
Geometric mean

- For Gaussian source
  - each sample is Gaussian, so that coefficients are also Gaussian, $\epsilon_{t,k}^2$ are all the same

$$G_{\mathrm{TC,Gaussian}} = \frac{\sigma_s^2}{\left(\prod_k \sigma_{t,k}^2\right)^{1/N}} = \frac{\frac{1}{N}\sum \sigma_{t,k}^2}{\left(\prod_k \sigma_{t,k}^2\right)^{1/N}}.$$

Example

- Determine the optimal bit allocation and corresponding TC gain for coding 2x2 image block using 2x2 DCT. Assuming the image is a Gaussian process with inter-sample correlation as shown below.



$$\rho_h = \rho_d = \rho,\ \rho_d = \rho^2.$$

Example Continued
(Convert 2x2 into 4x1)

- Correlation matrix

$$[\mathbf{C}]_s = E\left\{\begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix}[A \quad B \quad C \quad D]\right\} = \begin{bmatrix} C_{AA} & C_{AB} & C_{AC} & C_{AD} \\ C_{BA} & C_{BB} & C_{BC} & C_{BD} \\ C_{CA} & C_{CB} & C_{CC} & C_{CD} \\ C_{DA} & C_{DB} & C_{DC} & C_{DD} \end{bmatrix}$$

$$= \sigma_s^2 \begin{bmatrix} 1 & \rho_h & \rho_v & \rho_d \\ \rho_h & 1 & \rho_d & \rho_v \\ \rho_v & \rho_d & 1 & \rho_h \\ \rho_d & \rho_v & \rho_h & 1 \end{bmatrix}.$$

- DCT basis images

$$\frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \frac{1}{2}\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}, \quad \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}, \quad \frac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

- Equivalent transform matrix

$$[\mathbf{U}] = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Example Continued

$$[\mathbf{C}]_t = [\mathbf{V}][\mathbf{C}]_s[\mathbf{V}]^H \quad \longrightarrow$$

$$\sigma_{t,k}^2 = \{(1+\rho)^2, (1-\rho^2), (1-\rho^2), (1-\rho)^2\}\sigma_s^2$$

$$\sigma_t^2 = \left(\prod_k \sigma_{t,k}^2\right)^{1/4} = (1-\rho^2)\sigma_s^2; \qquad G_{\mathrm{TC}} = \frac{\sigma_s^2}{\sigma_t^2} = \frac{1}{1-\rho^2}$$

$$R_k = R + \frac{1}{2}\log_2 \frac{\epsilon_{t,k}^2 \sigma_{t,k}^2}{\left(\prod_k \epsilon_{t,k}^2 \sigma_{t,k}^2\right)^{1/N}}. \quad \longrightarrow$$

$R_k = \{4.64, 2, 2, -0.64\}$. (for R=2)

Optimal Transform

- Optimal transform
  - Should minimize the distortion for a given average bit rate
  - Equivalent to minimize the geometric mean of the coefficient variances
- When the source is Gaussian, the optimal transform is the Karhunen-Loeve transfrom, which depends on the covariance matrix between samples
  - Basis vectors are the eigen vectors of the covariance matrix, the coefficient variances are the eigen values

$$[\mathbf{C}]_s \phi_k = \lambda_k \phi_k, \quad \text{with} \quad \langle \phi_k, \phi_l \rangle = \delta_{k,l}. \qquad \sigma_k^2 = \lambda_k.$$

$$\prod_{k \in \mathcal{N}} \sigma_{t,k}^2 = \det[\mathbf{C}]_t = \det[\mathbf{C}]_s.$$

$$D_{\mathrm{TC}} = \epsilon_{\mathrm{Gaussian}}^2 (\det[\mathbf{C}]_s)^{1/N} 2^{-2R}.$$

$$G_{\mathrm{TC,KLT}} = \frac{\epsilon_s^2}{(\prod_k \epsilon_k^2)^{1/N}} \frac{\sigma_s^2}{(\prod_k \lambda_k)^{1/N}} = \frac{\epsilon_s^2}{(\prod_k \epsilon_k^2)^{1/N}} \frac{\sigma_s^2}{(\det[\mathbf{C}]_s)^{1/N}}$$

Example

- Determine the KLT for the 2x2 image block in the previous example

$$[\mathbf{C}]_s = \sigma_s^2 \begin{bmatrix} 1 & \rho_h & \rho_v & \rho_d \\ \rho_h & 1 & \rho_d & \rho_v \\ \rho_v & \rho_d & 1 & \rho_h \\ \rho_d & \rho_v & \rho_h & 1 \end{bmatrix}$$

Determine the eigenvalues by solving: $\det([\mathbf{C}]_s - \lambda[\mathbf{I}]) = 0$

$$\lambda_k = \{(1+\rho)^2, (1-\rho^2), (1-\rho^2), (1-\rho)^2\}\sigma_s^2.$$

(same as the coefficient variances with DCT)

Determine the eigenvectors by solving $([\mathbf{C}]_s - \lambda[\mathbf{I}])\phi_k = \mathbf{0}$

Resulting transform is the DCT

Properties of KLT

- Optimal transform for Gaussian sources
- Nearly optimal transform for non-Gaussian sources

- Minimal approximation error for K<N coefficients among all unitary transforms
- KLT has highest energy compaction
- Coefficients are uncorrelated

- Requires a stationary source with known covariance matrix – most sources vary spatially and temporally
- No fast algorithms – and not signal independent

JPEG Image Coder

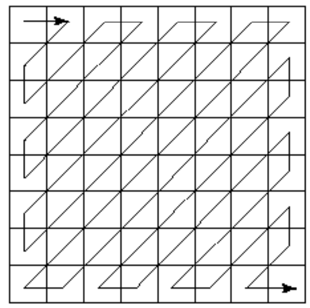
- Uses 8x8 DCT
- Each coefficient is quantized using a uniform quantizer, but the step sizes vary based on coefficient variances and their visual importance
- Quantized coefficients are converted into binary bitstreams using runlength coding plus Huffman coding

JPEG: a bit more detail

Perceptual based quantization matrix:

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

Zig-zag ordering of DCT coefficients:



Runlength coding example:

DCT coefficients: [5 0 0 2 3 0 0 4 0 0 0 0 0 0 1 0 0 0 0 0 ... 0]

Coding symbols: 5, (2,2), (0,3), (2,4), (6,1), EOB

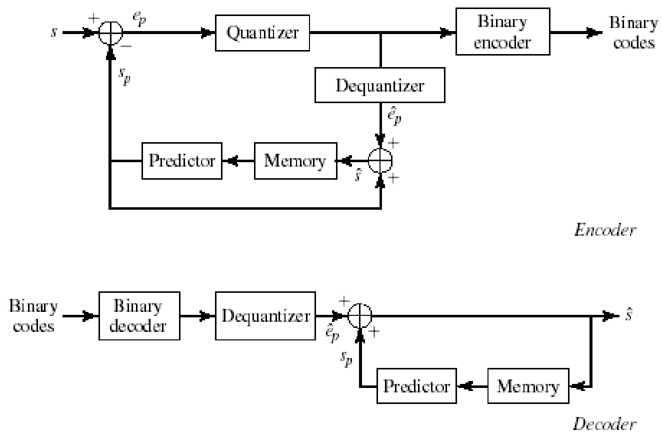
Predictive Coding

- Motivation: Predicts a sample from past samples and quantize and code the error only
- If the prediction error is typically small, then it can be represented with a lower average bit rate
- Optimal predictor: minimize the prediction error

A	B	C	D
E	F	G	H
I	J	K	L

$$\hat{f}_K = a f_F + b f_G + c f_H + d f_J$$

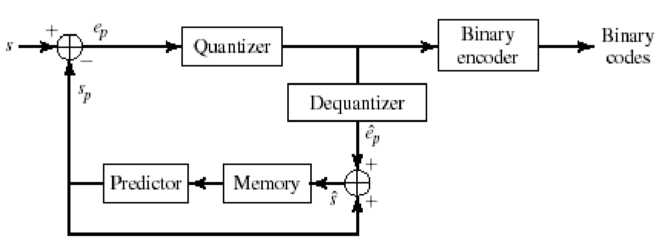
Encoder and Decoder Block Diagram (Closed Loop Prediction)



Distortion in Predictive Coder

- With closed-loop prediction, reconstruction error in a sample is equal to the quantization error for the prediction error.

$$\hat{s} = s_p + \hat{e}_p = s_p + e_p - e_q = s - e_q.$$



Optimal Predictor

- Question: what predictor should we use?
  - Minimize the bit rate for coding the prediction error
  - Because quantization error with a given bit rate depends on the variance of the signal, minimizing the quantization error = minimizing the prediction error variance.
  - We will limit our consideration to linear predictor only

$$s_p = \sum_{k=1}^{K} a_k s_k$$

Linear Minimal MSE Predictor

- Prediction error:

$$\sigma_p^2 = E\{|\mathcal{S}_0 - \mathcal{S}_p|^2\} = E\left\{\left|\mathcal{S}_0 - \sum_{k=1}^{K} a_k \mathcal{S}_k\right|^2\right\}.$$

- Optimal coefficients must satisfy:

$$E\left\{\left(\mathcal{S}_0 - \sum_{k=1}^{K} a_k \mathcal{S}_k\right)\mathcal{S}_l\right\} = 0, \quad l = 1, 2 \ldots, K. \qquad (*)$$

$$\sum_{k=1}^{K} a_k R(k, l) = R(0, l), \quad l = 1, 2 \ldots, K,$$

Note (*) is also known as the orthogonality principle in estimation theory

Matrix Form

- The previous equation can be rewritten as:

$$\begin{bmatrix} R(1,1) & R(2,1) & \cdots & R(K,1) \\ R(1,2) & R(2,2) & \cdots & R(K,2) \\ \cdots & \cdots & \cdots & \cdots \\ R(1,K) & R(2,K) & \cdots & R(K,K) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_K \end{bmatrix} = \begin{bmatrix} R(0,1) \\ R(0,2) \\ \cdots \\ R(0,K) \end{bmatrix}$$

$$[\mathbf{R}]\mathbf{a} = \mathbf{r}.$$

- Optimal solution:

$$\mathbf{a} = [\mathbf{R}]^{-1}\mathbf{r}.$$

$$\sigma_p^2 = E\{(\mathcal{S}_0 - \mathcal{S}_p)\mathcal{S}_0\} = R(0,0) - \sum_{k=0}^{K} a_k R(k,0)$$

$$= R(0,0) - \mathbf{r}^T\mathbf{a} = R(0,0) - \mathbf{r}^T R^{-1}\mathbf{r}.$$

Predictive Coding Gain

$$D_{\text{DPCM}} = \epsilon_p^2 \sigma_p^2 2^{-2R} \qquad G_{\text{DPCM}} = \frac{D_{\text{PCM}}}{D_{\text{DPCM}}} = \frac{\epsilon_s^2}{\epsilon_p^2}\frac{\sigma_s^2}{\sigma_p^2}$$

$$\sigma_{p,\min}^2 = \lim_{K\to\infty} \sigma_p^2 = \exp\left(\frac{1}{2\pi}\int_{-\pi}^{\pi} \log_e S(e^{j\omega})\, d\omega\right) \qquad \sigma_{p,\min}^2 = \lim_{K\to\infty}\left(\prod_k \lambda_k\right)^{1/N},$$

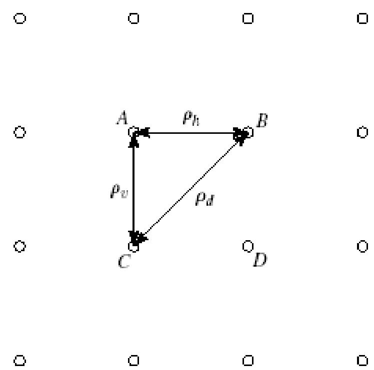
$$\sigma_s^2 = \lim_{K\to\infty} \frac{1}{K}\sum_k \lambda_k$$

$$\lim_{K\to\infty} G_{\text{DPCM}} = \frac{\epsilon_s^2}{\epsilon_p^2} \frac{\lim_{K\to\infty} \frac{1}{K}\sum_k \lambda_k}{\lim_{K\to\infty} \left(\prod_k \lambda_k\right)^K}.$$

K -> 1/K

TC=PC if the block length in TC and the predictive order in PC both go to infinity
PC is better for any finite length

Example



$$\hat{D} = a_1 C + a_2 B + a_3 A$$

Example Continued

$$\begin{bmatrix} R(C,C) & R(C,B) & R(C,A) \\ R(B,C) & R(B,B) & R(B,A) \\ R(A,C) & R(A,B) & R(A,A) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} R(D,C) \\ R(D,B) \\ R(D,A) \end{bmatrix}$$

$$\begin{bmatrix} 1 & \rho_d & \rho_v \\ \rho_d & 1 & \rho_h \\ \rho_v & \rho_h & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \rho_h \\ \rho_v \\ \rho_d \end{bmatrix}.$$

In the special case of $\rho_h = \rho_v = \rho$, $\rho_d = \rho^2$, the optimal predictor is

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \rho \\ \rho \\ -\rho^2 \end{bmatrix}.$$

The MSE of this predictor, using Equation (9.2.10), is

$$\sigma_p^2 = R(0,0) - [R(0,1) \quad R(0,2) \quad R(0,3)] \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = (1-\rho^2)^2 \sigma_s^2.$$

$$G_{\text{DPCM}} = \frac{\sigma_s^2}{\sigma_p^2} = \frac{1}{(1-\rho^2)^2}$$

(DPCM is better than TC for this case!)

Predictive Coding for Video

- For video, we apply prediction both among pixels in the same frame (intra-prediction or spatial prediction), and also among pixels in adjacent frames (inter-prediction or temporal prediction)
- Temporal prediction is done with motion compensation
- More on this subject in the next lecture.

Homework

- Reading assignment: Sec. 9.1,9.2
- Written assignment:
  - Prob. 9.3,9.4,9.5, 9.6, 9.7
- Computer assignment
  - Prob. 9.8,9.9