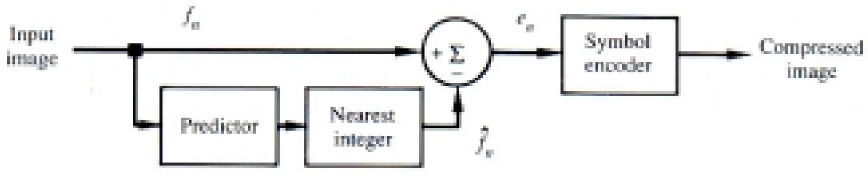
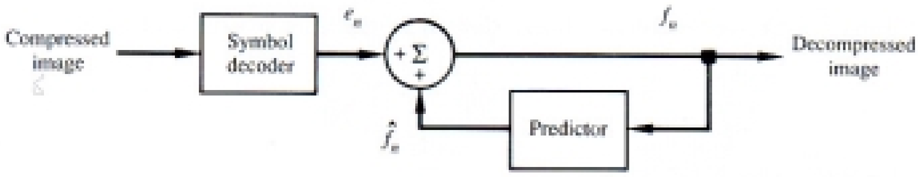
6.4.3 Lossless Predictive Coding

- Elimation of interpixel redundancies
- Extracting and coding only the new information in each pixel
  - new information = actual value – predictive value

    encoder, decoder



(a)



(b)

- Encoder
  - prediction error

$$e_n = f_n - \hat{f}_n \rightarrow \text{ coded using variable-length code}$$

- Decoder

$$f_n = e_n + \hat{f}_n$$

- Linear prediction

$$\hat{f}_n = round\left[\sum_{i=1}^{m} \alpha_i f_{n-i}\right]$$

- 1D prediction can be written as

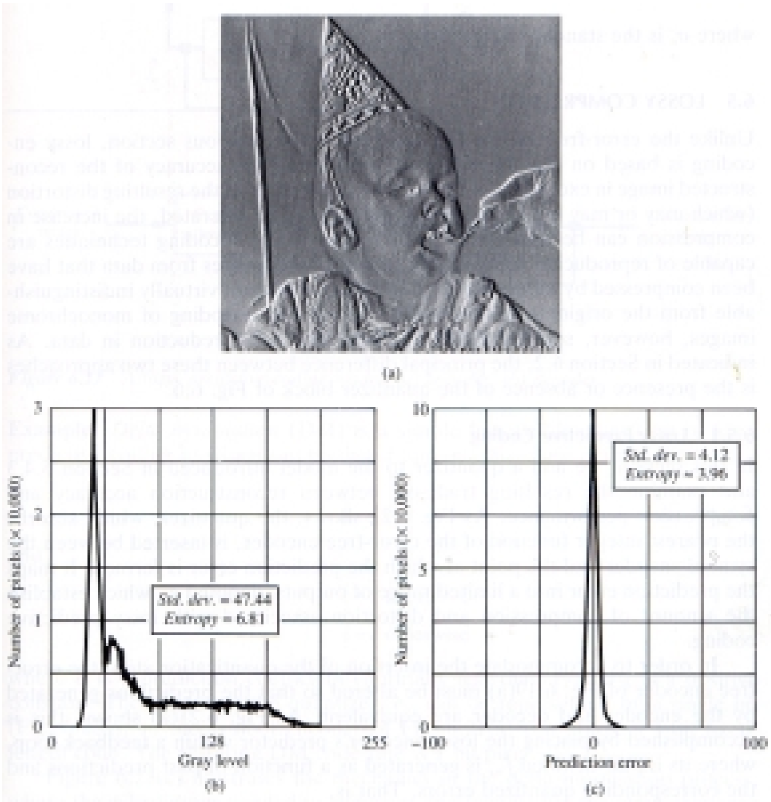
$$\hat{f}_n(x, y) = \text{round}\left[\sum_{i=1}^{m} \alpha_i f(x, y - i)\right]$$

- 2-D predictive coding
  - input : left-to-right

    top-to-bottom

- ex.) predictor

$$\hat{f}(x,y) = round[\alpha\ f(x,y-1)],\alpha = 1$$: differential coding



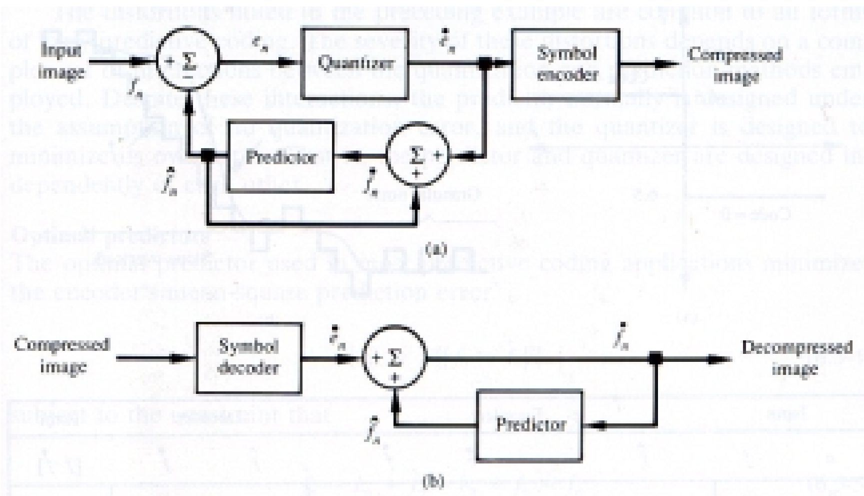
- prediction error often is modeled as Laplacian pdf

$$p_e(e) = \frac{1}{\sqrt{2}\sigma_e}\exp\left(\frac{-\sqrt{2}\,|e|}{\sigma_e}\right)$$

6.5 Lossy Compression

- monochorom image
  - commonly compressed by more than 30:1
  - indistinguishable from the originals at 10:1 to 20:1
- error free coding : seldom result in more than 3:1 reduction
- principal difference
  - presence or absence of the <u>quantizer block</u>

6.5.1 Lossy Predictive Coding



- input of predictor is a function of past predictions and the corresponding quantized error

$$\dot{f}_n = \dot{e}_n + \hat{f}_n$$

ex.) Delta Modulation (DM)

$\hat{f}_n = \alpha \, \dot{f}_n$ where $\alpha < 1$ prediction coeffs.

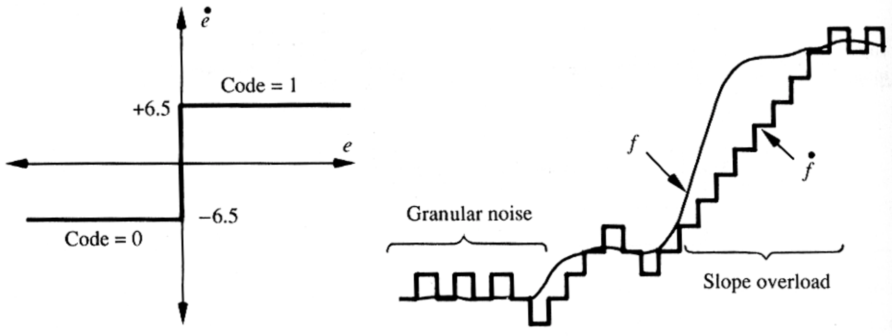
$$\dot{e}_n = \begin{cases} +\zeta & for \quad e_n > 0 \\ -\zeta & otherwise \end{cases}$$

Positive constant

: single bit

→ 1-bit fired length code

- DM code rate : 1 bpp



(a) (b)

Input		Encoder				Decoder		Error
n	f	$\hat{f}$	e	$\dot{e}$	$\dot{f}$	$\hat{f}$	$\dot{f}$	$[f-\dot{f}]$
0	14	—	—	—	14.0	—	14.0	0.0
1	15	14.0	1.0	6.5	20.5	14.0	20.5	−5.5
2	14	20.5	−6.5	−6.5	14.0	20.5	14.0	0.0
3	15	14.0	1.0	6.5	20.5	14.0	20.5	−5.5
•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•
14	29	20.5	8.5	6.5	27.0	20.5	27.0	2.0
15	37	27.0	10.0	6.5	33.5	27.0	33.5	3.5
16	47	33.5	13.5	6.5	40.0	33.5	40.0	7.0
17	62	40.0	22.0	6.5	46.5	40.0	46.5	15.5
18	75	46.5	28.5	6.5	53.0	46.5	53.0	22.0
19	77	53.0	24.0	6.5	59.6	53.0	59.6	17.5
•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•

- slope overload
  - $n = 14 \ to \ 19$
  - input : large change
  - ζ : too small, to represent input's large change
- granular noise
  - $n = 0 \ to \ 7$
  - smooth region
- predictor, quantizer
  - designed independently of each other

optimal predictors

- encoder's mean square prediction error → minimization

$$E\{e_n^2\} = E\left\{ \left[f_n - \hat{f}_n \right]^2 \right\}$$

constraints

$$\dot{f}_n = \dot{e}_n + \hat{f}_n \approx e_n + \hat{f}_n = f_n \quad \text{: negligible quantization error}$$

$$\hat{f}_n = \sum_{i=1}^{m} \alpha_i \ f_{n-i} \quad \text{: linear combination of m preview pixel}$$

: simplify the analysis considerably

⇒ DPCM (Differential PCM)

$$E\{e_n^2\} = E\left\{\left[f_n - \sum_{i=1}^{m}\alpha_i f_{n-i}\right]\right\}$$

→ minimization

$$E\left[e^2(k)\right] = E\left[\left\{x(k) - \sum_{i=1}^{N} a_i\, x(k-i)\right\}^2\right]$$

differentiating with respect to a_i and equating with zero

$$\sum_{i=1}^{p} a_i\, E[x(k-i)\cdot x(k-j)] = E[x(k)x(k-j)]$$

$$\sum_{i=1}^{p} a_i R_{i-j} = R_j \qquad j = 1, 2, \cdots, p$$

assuming $E[x(k)] = 0 \Longrightarrow$ in matrix form

$$\underbrace{\begin{bmatrix} R_o & R_1 & R_2 & \cdots & R_{p-1} \\ R_1 & R_0 & & & R_{p-2} \\ & R_1 & & & \\ \vdots & & \ddots & & \vdots \\ & \vdots & & & \\ R_{p-1} & R_{p-2} & & & R_0 \end{bmatrix}}_{\underline{\underline{R}}} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_p \end{bmatrix}$$

-differentiated value = 0

if f_n : zero mean, variance σ^2

$$\alpha = \mathbf{R}^{-1}\mathbf{r}$$

where

$$\mathbf{R} = \begin{bmatrix} E\{f_{n-1} f_{n-1}\} & E\{f_{n-1} f_{n-2}\} & \cdots & E\{f_{n-1} f_{n-m}\} \\ E\{f_{n-2} f_{n-1}\} & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & & & \\ E\{f_{n-m} f_{n-1}\} & E\{f_{n-m} f_{n-2}\} & \cdots & E\{f_{n-m} f_{n-m}\} \end{bmatrix}$$

$$\mathbf{r} = \begin{bmatrix} E\{f_n f_{n-1}\} \\ \vdots \\ E\{f_n f_{n-m}\} \end{bmatrix} \quad \text{and} \quad \alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix}$$

- coeffs. : depend only on the autocorrelation of the pixels

- variance of prediction error

$$\sigma_e^2 = \sigma^2 - \alpha^T \mathbf{r} = \sigma^2 - \sum_{i=1}^{m} E\{f_n f_{n-1}\}\alpha_i$$

- computation of all autocorrelation
  - difficult in practice
  - local prediction (autocorrelations are computed image by image)
    : almost never used
  - global coeffs.
    : computed from a simple image model

ex) 2-D Markov process with separable autocorrelation function

$$E\{f(x,y)f(x-i,y-1)\} = \sigma^2 \rho_v^i \rho_h^j$$

generalized fourth-order linear predictor

$$\hat{f}(x,y) = \alpha_1 f(x,y-1) + \alpha_2 f(x-1,y-1)$$
$$+\alpha_3 f(x-1,y) + \alpha_4 f(x-1,y+1)$$

where $\alpha_1 = \rho_h$, $\alpha_2 = -\rho_v \rho_h$, $\alpha_3 = \rho_v$, $\alpha_4 = 0$

where ρ_h, ρ_v : the horizontal and vertical correlation coeffs. of image

- $\sum_{i=1}^{M} \alpha_i \le 1$: negative feedback
  - predictor's output : falls within the allowed range of gray levels and to reduce the impact of transmission noise

    ※ 한 입력의 error 가 발생하면 predictor 에서 이후의 입력값을 prediction 할 때 그 영향이 계속 유지되어 계속 잘못된 prediction 값이 얻어진다. 시간이 지남에 따라 이러한 오차가 빨리 작아지도록 하기위해서 $\sum_{i=1}^{m} \alpha_i \le 1$ 이라는 조건이 필요하다.

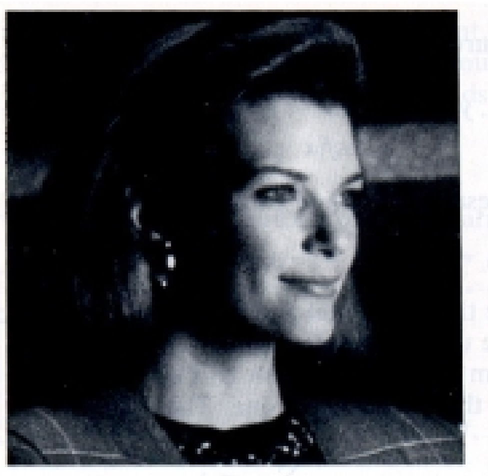
ex.)

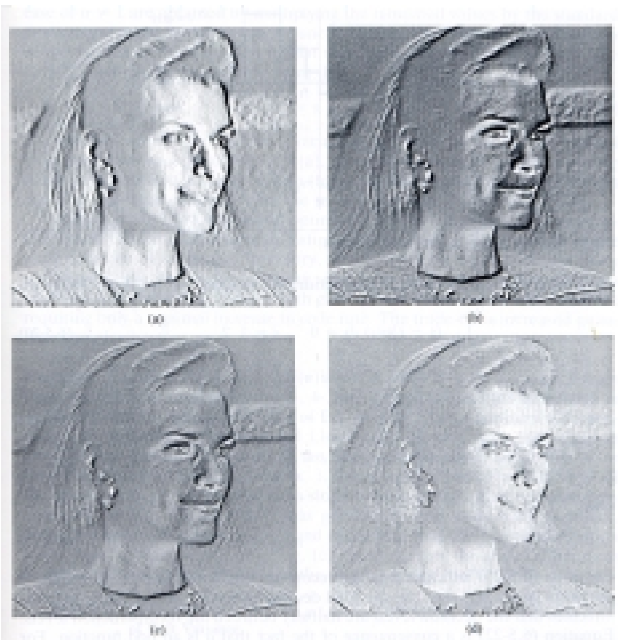
$$\hat{f}(x, y) = 0.97f(x, y - 1)$$

$$\hat{f}(x, y) = 0.5f(x, y - 1) + 0.5f(x - 1, y)$$

$$\hat{f}(x, y) = 0.75f(x, y - 1) + 0.75f(x - 1, y) - 0.5f(x - 1, y - 1)$$

$$\hat{f}(x, y) = \begin{cases} 0.97f(x, y - 1) & \text{if } \Delta h \le \Delta v \\ 0.97f(x - 1, y) & \text{otherwise} \end{cases}$$





<u>Optimal Quantization</u>