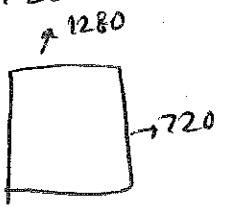
(g) Because the second method can provide an overall better spatial-temporal resolution, at only slightly higher bit rate, the second method is better.

- 720 format: progressive, 60 $\frac{\text{frame}}{\text{sec}}$, 1280 × 720



a) temporal freq = 30 cycle/s

b) vertical freq = $\frac{720}{2}$ = 360 cycle/picture height

c) $f_{h\max} = IAR \times f_{v\max} = \frac{1280}{720} \times 360 = 640$ ✓

- 1080-i format: interlaced → field rate = 60, 1920 × 540 pels/field

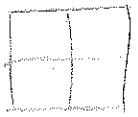
d) temporal freq = 30,

e) vertical freq = $\frac{540}{2}$ = 270

f) $f_{h\max} = IAR \times f_{v\max} = 270 \times \frac{1920}{540} = 960$ ✓

g) in this case vertical frequency for part (e) would change, and the effective $f_{v\max}$ = 540 because two fields are merged. ✓

(the human eye is less sensitive to spatial details because of motion.)

h) 

→ every 2×2 → 6×8 = 48 bits

→ $\frac{1280 \times 720}{2 \times 2} \times \frac{12}{4} \times 48 \frac{\text{bits}}{\text{block}} \times 60 =$

R_1 = bit rate = $12 \times 80 \times (720)^2$ bits/sec

i) $R_2 = \frac{1920 \times 540}{2 \times 2} \times \frac{12}{4} \times 48 \text{ bits} \times 60 = 1920 \times 540 \times 720$

→ $\frac{R_1}{R_2} = \frac{1280 \times 720 \times 720}{1920 \times 540 \times 720} = \frac{128 \times 4}{3 \times 2} = \frac{8}{9}$ → $R_1 < R_2$

(j) ✗ bit rate of second one is more than first one, so in terms of bit rate efficiency the second one is better.

2. Square error $E = \sum_{\vec{x}\in B} | f_1(\vec{x}) - f_2(\vec{x}+d(\vec{x}))|^2$

where $d(\vec{x}) = \sum_{k=1}^{q} \phi_k(x,y) \begin{bmatrix} d_{x,k} \\ d_{y,k} \end{bmatrix} = \begin{bmatrix} d_x(\vec{x}) \\ d_y(\vec{x}) \end{bmatrix}$

Suppose that we use the first order gradient, the gradient is:

$$\begin{bmatrix} d_x(\vec{x}) \\ d_y(\vec{x}) \end{bmatrix} = \frac{\partial E}{\partial \vec{d}_k} = -\sum_{\vec{x}\in B} 2(f_1(\vec{x}) - f_2(\vec{x}+d(\vec{x}))) \begin{bmatrix} \frac{\partial f_2}{\partial x} \cdot \frac{\partial d_x(\vec{x})}{\partial d_{x,k}} \\ \frac{\partial f_2}{\partial y} \cdot \frac{\partial d_x(\vec{x})}{\partial d_{y,k}} \end{bmatrix}$$

and $\frac{\partial d_x(\vec{x})}{\partial d_{x,k}} = \phi_k(\vec{x})$ $\frac{\partial d_y(\vec{x})}{\partial d_{y,k}} = \phi_k(\vec{x})$

$$\therefore \frac{\partial E}{\partial \vec{d}_k} = -\sum_{\vec{x}\in B} 2(f_1(\vec{x}) - f_2(\vec{x}+d(\vec{x}))) \phi_k(x,y) \begin{bmatrix} \frac{\partial f_2}{\partial x} \\ \frac{\partial f_2}{\partial y} \end{bmatrix}$$ evaluated at $x+d(x)$

The gradient descent algorithm: set $E_{old} = E_{max}$

$n=0$, the initial condition $\vec{d}_k^{(0)} = [0 \;\; 0]^T$

$\vec{\phi}^{(n)}(x) = \sum \phi_k(x,y) \vec{d}_k^{(n)}$, $e(\vec{x}) = f_1(\vec{x}) - f_2(\vec{x} + \vec{d}^{(n)}(x))$

$E_{new} = \sum_{\vec{x}\in B} [e(\vec{x})]^2$, if $\frac{|E_{old} - E_{new}|}{E_{old}} > \varepsilon$ (ε is the converging threshold)

Then $E_{old} = E_{new}$

$$\vec{g}^{(n)} = \frac{\partial E}{\partial \vec{d}_k}\Big|_{\vec{d}_k^{(n)}} = -2 \sum e(\vec{x}) \phi_k(\vec{x}) \begin{bmatrix} \frac{\partial f_2}{\partial x} \\ \frac{\partial f_2}{\partial y} \end{bmatrix}\Big|_{\vec{x}+d^{(n)}(x)}$$ ✓

$\vec{d}_k^{(n+1)} = \vec{d}_k^{(n)} - \alpha \vec{g}^{(n)}$, $n = n+1$, and goto

else return $\vec{d}(\vec{x}) = \vec{d}^{(n)}(\vec{x})$

✓

3. (a) bit rate = $\log_2(L)$ bit/sample ✓

(b) bit rate = $\log_2(L_1) + \log_2(L_2)$ bit/sample. ✓

$\log_2(L_1) + \log_2(L_2) = \log_2(L_1 L_2) = \log_2(L)$ ✓

therefore, the two methods have the same bit rate.

(c). For every sample: (N is the number of dimensions)

operations: $\underbrace{N \cdot L}_{\text{Substraction}} + \underbrace{N \cdot L}_{\text{square}} + \underbrace{(N-1)L}_{\text{Sum}} = L(3N-1)$ ✓

You can also just say NL

(d.) For each sample vector: $\rightarrow N$

operations = $\underbrace{N + (N-1) + 1}_{\text{Norm}}$ + $\underbrace{L_1}_{\text{gain quantization}}$ + $\underbrace{N \cdot L_2 + N \cdot L_2 + (N-1)L_2}_{\text{shape-quantization}}$ ✓ ✓ ✓

$= L_2(3N-1) + 2N + L_1$ $\quad N + L_1 + NL_2$

When N is large $\frac{Op(VQ)}{Op(gsVQ)} = \frac{L(3N-1)}{L_2(3N-1)+2N+L_1} = \frac{3L - \frac{L}{N}}{3L_2 - \frac{L_2}{N} + 2 + \frac{L_1}{N}} \approx \frac{3L_1L_2}{3L_2+2} > 1$

$\therefore$ gain-shape VQ requires less computation. ✓ $\approx$ 4

(e) The conventional VQ is likely to yield less quantization distortion. ✓

Since the vector might not uniformly distributed in shape and gain.

(f) pro: less computation

con: yields bigger distortion. ✓

4. (a,b) Both codebook satisfies the necessary condition for minimizing the MSE.

Since the vectors are uniformly distributed.

$\frac{d\,MSE}{d\,\vec{g}_i} = \vec{2} \sum (\vec{x}_k - \vec{g}_i) = 0 \rightarrow \vec{g}_i = p(x) \sum_k \vec{x}_k \rightarrow g_{ix} = p(\vec{x}) \sum x$

$g_{iy} = p(\vec{x}) \sum y$

for (b). $MSE = 4 \cdot \frac{1}{4} \cdot \int_0^1 \int_0^1 (x-a)^2 + (y-a)^2 \, dx\,dy$

$= \int_0^1 (y-a)^2 + \frac{1}{3}\left[(1-a)^3 + a^3\right] dy$

$= \frac{2}{3}\left[(1-a)^3 + a^3\right]$

$\frac{d\,MSE}{da} = \frac{2}{3}\left[-3(1-a)^2 + 3a^2\right] = 0$

$\therefore a = \frac{1}{2} \qquad MSE = \frac{1}{6}$

for (c) $MSE = 4 \cdot \frac{1}{4} \int_0^1 \int_{-x}^{x} (x-a)^2 + y^2 \, dy\,dx$

$\frac{d\,MSE}{da} = \int_0^1 \int_{-x}^{x} -2(x-a) \cdot \, dy\,dx$

$= \int_0^1 -4x^2 + 4ax \, dx$

$= -\frac{4}{3} + 2a = 0$

$\therefore a = \frac{2}{3}$

$MSE = \int_0^1 \int_{-x}^{x} (x-\frac{2}{3})^2 + y^2 \, dy\,dx$

$= \int_0^1 2x(x-\frac{2}{3})^2 + \frac{2}{3}x^3 \, dx$

$= \frac{1}{3} - \frac{8}{9} + \frac{4}{9} = \frac{2}{9}$

For this part, you could just say $a = \frac{1}{2}$ based on centroid condition

(c) Codebook in Figure (b) is better.

The necessary condition is not sufficient to guarantee quantizer optimality. because the result could be a local minimum.

c) as we see $mse_b < mse_c \rightarrow$

codebook in figure (b) is better in terms of mse.

→ so the necessary condition are satisfied, but for figure (c) it is an local optimum.

so necessary condition can not guarantee. lonely.

(for convex problem local min = global min but here is not convex)

$\sigma_1^2 = \sigma_2^2$ ← 5- $C_x = E\left(\begin{bmatrix} A \\ B \end{bmatrix}[A \;\; B]\right) = \begin{vmatrix} C_{AA} & C_{AB} \\ C_{BA} & C_{BB} \end{vmatrix} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\sigma^2$

$\Rightarrow$ eigenvalues → $\det(C - \lambda I) = 0 \rightarrow \begin{vmatrix} \sigma^2 - \lambda I & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 - \lambda I \end{vmatrix} =$

$\Rightarrow \lambda_1 = (1-\rho)\sigma^2, \; \lambda_2 = (1+\rho)\sigma^2$

$\Rightarrow$ now eigen vectors → $C\phi_1 = \lambda_1\phi_1, \; C\phi_2 = \lambda_2\phi_2$

$\Rightarrow \begin{bmatrix} \sigma^2 & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 \end{bmatrix}\begin{bmatrix} \phi_{11} \\ \phi_{12} \end{bmatrix} = \begin{bmatrix} \lambda_1\phi_{11} \\ \lambda_1\phi_{12} \end{bmatrix} \rightarrow$

$\phi_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \; \phi_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix} \rightarrow U = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ ← transform

b) equal to ~~eigenvector~~ eigenvalues → $\sigma_{t_1}^2 = (1-\rho)\sigma^2$

$\sigma_{t_2}^2 = (1+\rho)\sigma^2$

$R_1 = R + \frac{1}{2}\log\frac{\varepsilon_1^2\sigma_{t_1}^2}{\sqrt{\varepsilon_1^2\varepsilon_2^2\sigma_{t_1}^2\sigma_{t_2}^2}}, \quad R_2 = R + \frac{1}{2}\log\frac{\varepsilon_2^2\sigma_{t_2}^2}{\sqrt{\varepsilon_1^2\varepsilon_2^2\sigma_{t_1}^2\sigma_{t_2}^2}}$

$\propto$ $\propto$

→

$$\Rightarrow \sigma_t^2 = \sqrt{\sigma_{t1}^2 \sigma_{t2}^2} = \sqrt{1-\rho^2}\,\sigma^2$$

$$D(R) = \varepsilon^2 \sqrt{1-\rho^2}\,\sigma^2\, 2^{-\alpha R}$$

6- $B=1$, $A=0$ $\quad [R_{BB}]\,a = [R_{AB}]$

a) $\Rightarrow [\sigma^2]\,a = \rho\sigma^2 \rightarrow a = \rho$

b) $\sigma_p^2 = R(0,0) - \sum a_i R(i,0) = \sigma^2 - \rho(\rho\sigma^2) = \sigma^2(1-\rho^2)$

d) $D(R) = \varepsilon^2 \sigma^2 (1-\rho^2)\, 2^{-\alpha R}$

d) $\dfrac{D_6(R)}{D_5(R)} = \sqrt{1-\rho^2} < 1 \rightarrow D_6(R) < D_5(R)$

so distortion of problem (6) is less than problem (5)
so problem (6) gives lower reconstruction error.
and efficiency in terms of use for problem 6 is better
for the same bit rate
but because problem 5 had the KLT → so it decorrelates them and maximize energy compaction
so for problem 5, the coefficient would be decorrelated and have maximum energy compaction.

Not relevant to this prob.

The reason predictive coding is better is because, even if coding of a pixel involves only two pixels, its "effective" block length is ∞ because of the recursive nature of predictive coding. It however is more sensitive to transmission error.

7. (a) I-frame: It can be decoded independently. } what about coding

P-frame: It can only be decoded when the previous I-frame is known.

B-frame: It's coded using both previous frame and future frame.

Coding Efficiency: B > P > I

Complexity: B > P > I ✓

(b) (a) encoding order:

$f_0 \to f_4 \to f_2 \to f_1 \to f_3 \to f_8 \to f_6 \to f_5 \to f_7$ ✓

(c) Pro: Better compression, scalability, random access

Con: There will be delays. ✓

(d) Base layer: All the 2 frames.

Enhanced layers: P layers and B layers. ✓

We can actually generate 4 layers.

Base layer: f_0 f_8 ...

Enhanced layer 1: f_4 ...

2: f_2, f_6

3: f_1, f_3, f_5, f_7, ...

(a) I-frame: a frame that is coded without referencing other frames. Only intra prediction is used.

P-frame: a frame that is coded with prediction from a previous frame (which could be I- or P-frame)

B-frame: a frame that is coded with prediction from a previous and a following frame.