EL6123 S09 Final Exam Solution

1. (a) From the Fig. 1(b) $\begin{cases} a = \frac{b}{2} & \text{nearest neighbor condition} \\ b = \frac{a+1}{2} & \text{centroid condition} \end{cases}$

$$\Rightarrow \begin{cases} a = \frac{1}{3} \\ b = \frac{2}{3} \end{cases}$$

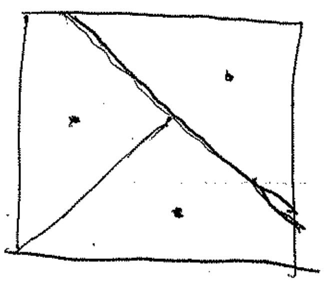
+5

$$mmse = \int_{-1}^{1}\int_{\frac{1}{3}}^{1}\left[\left(x-\frac{2}{3}\right)^2+y^2\right]\cdot 3\,dx\,dy = \frac{40}{27}$$

(b) For the codebook configuration in Fig. 1(c), it is equivalent to that in Fig 1(b) if the same value for a and b are choosen. Because the minimal mean square errors are the same between two codebooks.

+5

(c)



This codebook has less mean square error than Fig 1 (b) (c).

+5

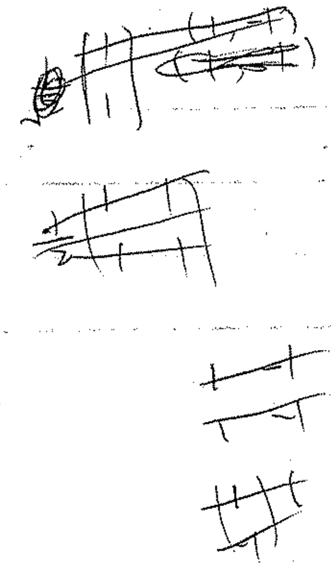
10

2.

$$\begin{bmatrix} R(A,A), & R(A,B) \\ R(B,A) & R(B,B) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} R(F,A) \\ R(F,B) \end{bmatrix}$$

$$\begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \rho \\ \frac{\rho}{2} \end{bmatrix} \Rightarrow \begin{cases} a = \dfrac{\rho - \frac{\rho^2}{2}}{1-\rho^2} \\ b = \dfrac{\frac{\rho}{2} - \rho^2}{1-\rho^2} \end{cases}$$

$$\sigma_p^2 = R(F,F) - (\rho, \tfrac{\rho}{2}) \begin{bmatrix} a \\ b \end{bmatrix} = \frac{1 - \frac{9}{4}\rho^2 + \rho^4}{1-\rho^2} \sigma^2$$



✓

3. (a). $U_{0,0} = u_0 \cdot u_0^T = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

$U_{0,1} = u_0 \cdot u_1^T = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$

$U_{1,0} = u_1 \cdot u_0^T = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$

$U_{1,1} = u_1 \cdot u_1^T = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$

(b) $U = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$

(c) $C_s = \begin{bmatrix} 1 & \rho & \rho & \rho^2 \\ \rho & 1 & \rho^2 & 0 \\ \rho & \rho^2 & 1 & 0 \\ \rho^2 & 0 & 0 & 1 \end{bmatrix}$ $\quad C_t = V C_s V^H = U C_s U^H = \begin{bmatrix} 1+\rho+\rho^2 & \frac{1}{2}\rho & \frac{1}{2}\rho & 0 \\ \frac{1}{2}\rho & 1-\rho^2 & 0 & -\frac{1}{2}\rho \\ \frac{1}{2}\rho & 0 & 1-\rho^2 & -\frac{1}{2}\rho \\ 0 & -\frac{1}{2}\rho & -\frac{1}{2}\rho & 1-\rho+\rho^2 \end{bmatrix} \sigma^2$

$\sigma_{t,k}^2 = \{1+\rho+\rho^2, 1-\rho^2, 1-\rho^2, 1-\rho+\rho^2\}\sigma^2$

(d) $R_k = R + \frac{1}{2}\log_2 \frac{\epsilon_{t,k}^2 \sigma_{t,k}^2}{(\Pi_k \epsilon_{t,k}^2 \sigma_{t,k}^2)^{\frac{1}{4}}} = R + \frac{1}{2}\log_2 \frac{\sigma_{t,k}^2}{([(1+\rho^2)^2-\rho^2](1-\rho^2)^2)^{\frac{1}{4}} \sigma^2}$

where $\sigma_{t,k}^2 = \{1+\rho+\rho^2, 1-\rho^2, 1-\rho^2, 1-\rho+\rho^2\}\sigma^2$

15

4. Solution: ~~Since~~ we will use parallel cameras configuration.

$$x_l = \frac{F}{Z}\left(X+\frac{B}{2}\right) \quad x_r = \frac{F}{Z}\left(X-\frac{B}{2}\right)$$

$$d_x = x_l - x_r = \frac{FB}{Z}$$

$$\Rightarrow Z = \frac{FB}{d_x}$$

So $\Delta Z = Z_2 - Z_1$

$$= FB\left(\frac{1}{dx_2} - \frac{1}{dx_1}\right)$$

It gives us one way to solve the problem.

Assume we can do automatic segmentation on each image to identify pixels in the two regions.

~~Since the two~~

So step 1° use segmentation to identify pixels in two regions on each image

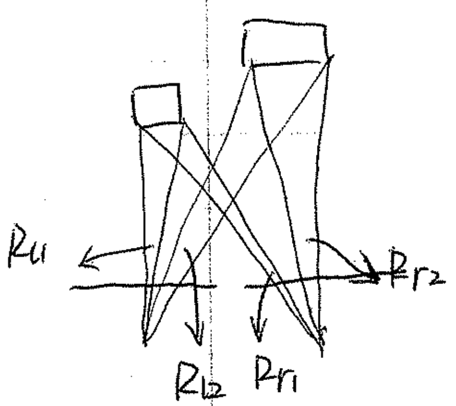
~~2° we can use the idea about Mean Square Error~~

And call ~~them~~ $R_{l1}, R_{l2}, R_{r1}, R_{r2}$

2° since the two foreground objects have flat front surface.

So dx_1, dx_2 can consider as constant

~~3° we can use the idea about Mean Square Error~~



3° ~~we can us~~ for R_{l1}, R_{r1}

we can use minimize $\left\{ \sum_{R_{l1}} \left| f_{R_{r1}}(x+dx_1, y) - f_{R_{l1}}(x, y) \right|^2 \right\}$ to predict dx_1

4° same method for R_{l2}, R_{r2}

minimize $\left\{ \sum_{f=R_{l2}} \left| f_{R_{r2}}(x+dx_2, y) - f_{R_{l2}}(x, y) \right|^2 \right\}$ to predict dx_2.

5° solve the function: $\Delta Z = FB\left(\frac{1}{dx_2} - \frac{1}{dx_1}\right)$ to get ΔZ.

11

5. (a) ① Intra-mode prediction

② ~~Integer~~ Integer DCT transform with variable block sizes

③ Adaptive deblocking filtering

④ Multiple reference frame prediction

+5

(b) Layered coding: A coding method to divide bit stream into ~~several~~ base Layer and enhancement layers. Decodering the base layer can achieve a low quality video, while combining the enhancement layers, higher quality video ~~can~~ will be reconstructed.

Compared with single layer coding, layered coding method could be adopted to scalable coding, which enable customers with different access bandwidth and decoding capability ~~to~~ receive the same ~~bit stream~~ video with different quality. ~~LC~~ Different Layers of LC can be transmit through different ~~channels~~ networks to reduce the effect of transmission error. Note that the base layer should be transmi on a most reliable ~~channel~~ network.

transmission control for LC MDC ?

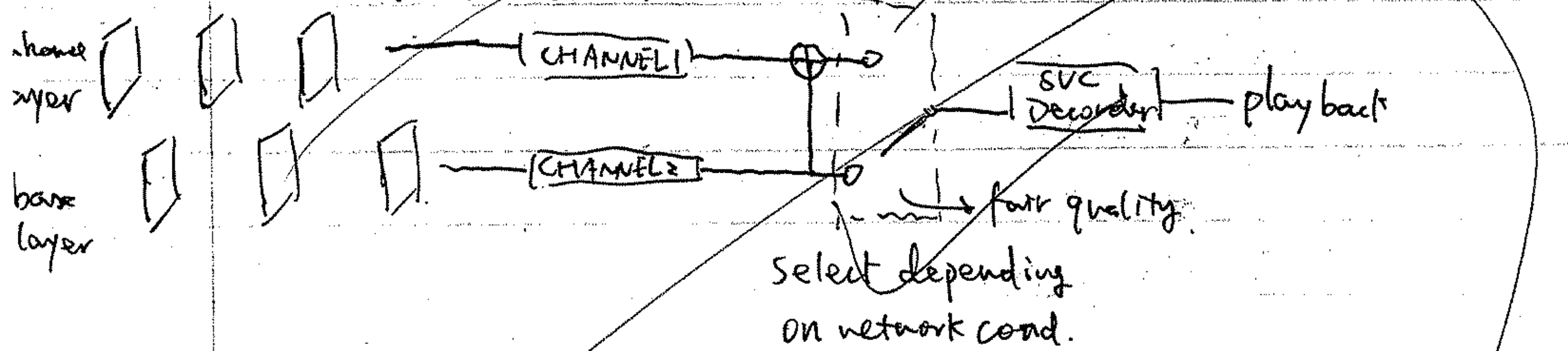
+6

MDC is a coding method that ~~divide~~ divide bit stream into multiple channels. Acceptable video ~~can~~ can be decodered from ~~one~~ any channel. Decoding using multiple channels will obtain video with higher quality.

Different channels of MDC should be transmitted ~~through~~ different networks to redu transmission error.

~~~~ The coding effeciency of LC is higher than that of MDC, because dividing video into different channels reduce the correlation between frames thus need more bits to code. MDC is more error ~~~~ resilient than LC, because the lost of one channel will not lead to serious distortion while the lost of base layer in LC will corrupt the video.
~~~~

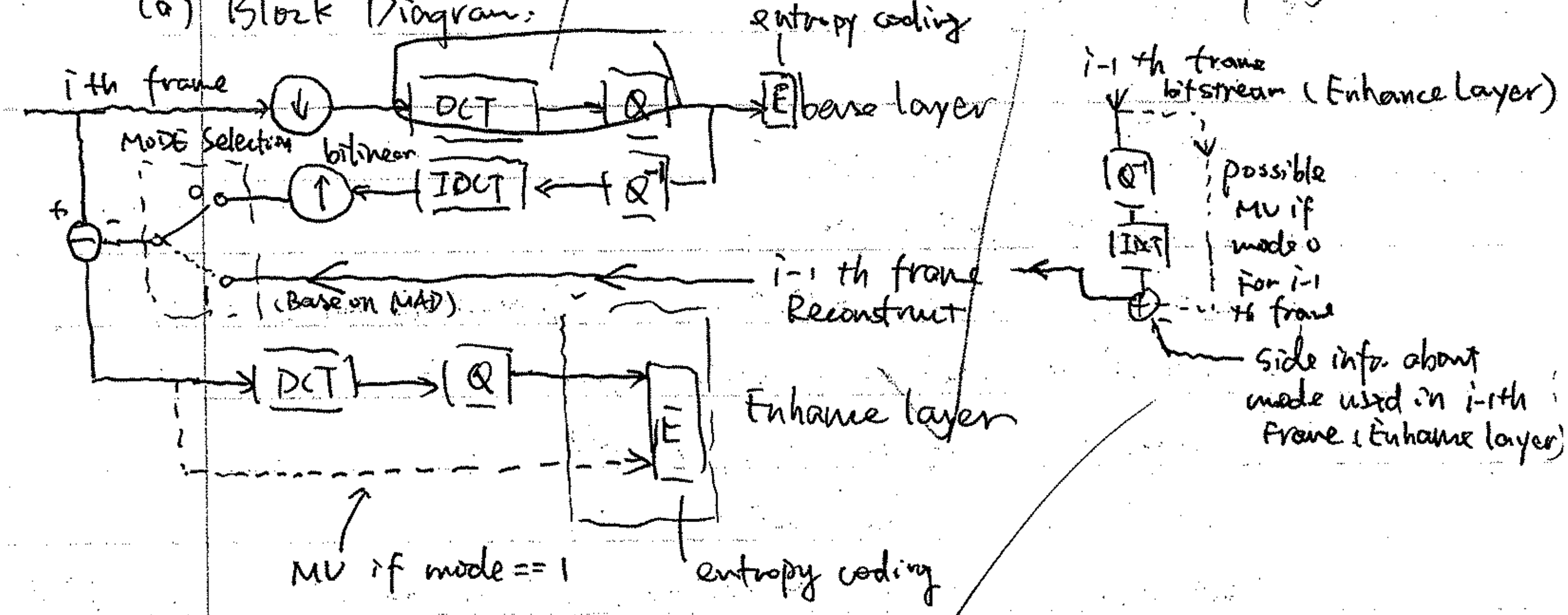
LC with transmission control



(*) Note that the above diagram omit the Layered encoder part.

6.

(a) Block Diagram:



(*) the above diagram omits some components in decoder of prediction loop

(b) (Implementation starts at the proceeding page)

20

(b)

```
Function RF = SpatialCoding(CF, PF, BCF, QP, Q.Matrix, outfile)
    [Height Width] = size(CF);
    RF = zeros(Height, Width);   IFrame = interpolate(BCF);
  for r=1:16:Height
    for c=1:16:Width
      MBlock = CF(r:r+15, c:c+15);
  [mvh_PF, mvv_PF, PredictedMBlock] = MotionEstimation(MBlock, PF);
   SAD_PF = sum(sum(abs(MBlock-predictedMBlock_PF)));
   predictedMBlock_BCF = Iframe(r:r+15, c:c+15);
   SAD_BCF = sum(sum(abs(MBlock-predictedMBlock_BCF)));
  MinErr = Min(SAD_FP, SAD_BCF);
  if minErr == SAD_PF
     Mode=0 ;   mvh=mvh_PF;  mvv=mvv_PF;
else if minErr == SAD_BCF
     Mode=1;   mvh = 0.;  mvv=0;
 end;
 if Mode=0
    predictedMBlock = predictedMBlock_PF
else if Mode=1
    predicted MBlock = predicted MBlock_BCF
end;
 Error Block = MBlock - predicted MBlock;
    Block1 = ErrorBlock(1:8, 1:8);
   DCTBlock1 = dct2(Block1)
   QDCT1 = quantizeDCT(DCTBlock1, QP, QMatrix).
```

```
Block2 = ErrorBlock(9:16 ; 1:8);
DCTBlock2 = dct2(Block2);
QDCT2 = quantizeDCT(DCTBlock2, QP, QMatrix);
Block3 = ErrorBlock(1:8 ; 9:16);
DCTBlock3 = dct2(Block3);
QDCT3 = quantizeDCT(DCTBlock3, QP, QMatrix);
Block4 = ErrorBlock(9:16 ; 9:16);
DCTBlock4 = dct2(Block4);
QDCT4 = quantizeDCT(DCTBlock4, QP, QMatrix);
EntropyCoding(mode, mvh, mvw, QDCT1, QDCT2, QDCT3, QDCT4, outfile);
QuantizedDCTBlock1 = dequantizeDCT(QDCT1, QP, QMatrix);
QuantizedDCTBlock2 = dequantizeDCT(QDCT2, QP, QMatrix);
QuantizedDCTBlock3 = dequantizeDCT(QDCT3, QP, QMatrix);
QuantizedDCTBlock4 = dequantizeDCT(QDCT4, QP, QMatrix);
QuantizedErrBlock = zeros(16, 16)
QuantizedErrBlock(1:8, 1:8) = QuantizedDCTBlock1;
QuantizedErrBlock(9:16, 1:8) = QuantizedDCTBlock2;
QuantizedErrBlock(1:8, 9:16) = QuantizedDCTBlock3;
QuantizedErrBlock(9:16, 9:16) = QuantizedDCTBlock4;
RF(r:r+15, c:c+15)
QuantizedMBlock = predictedMBlock + QuantizedErrBlock;
RF(r:r+15, c:c+15) = QuantizedMBlock;
end;
end;
```