Chapter 2 Entropy Coding

- Statistical modeling and entropy
- Huffman coding
- Milestone: predictive still image codec

Marginal entropy

- Definition

$$H(X) = -\sum_{x \in A_X} p_X(x) \log_2 p_X(x)$$

- Predefined Code Table
  - Average codeword length (AWL)
  - Minimum average codeword length

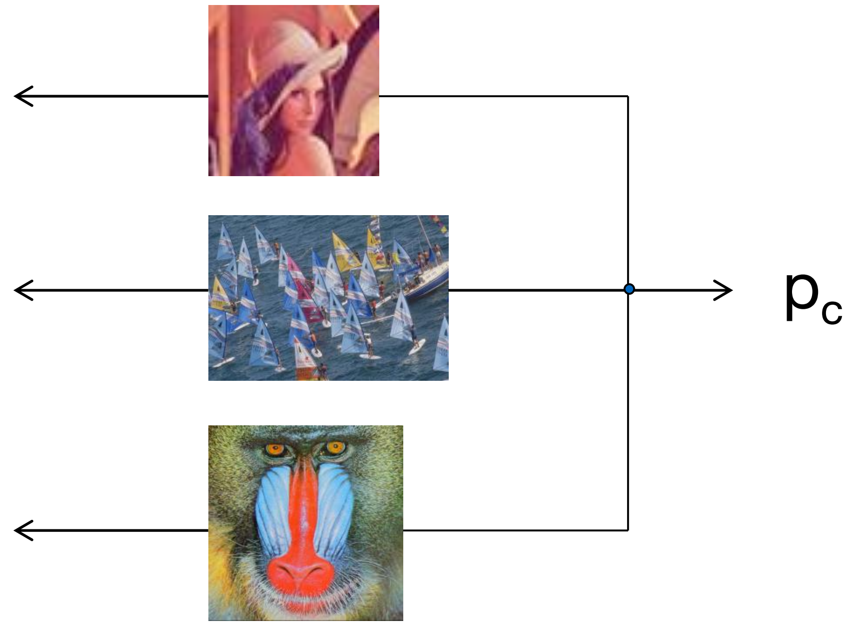
$$\min(AWL) = -\sum_{x \in A_X} \underbrace{\boxed{p_X(x)}}_{\text{Source}} \cdot \underbrace{\boxed{\log_2 p_C(x)}}_{\text{Code table}}$$

Common code table (E2-1c)

$$AWL_1 = -\sum_{x \in A_{X_1}} p_1(x) \cdot \log_2 p_C(x)$$ p1

$$AWL_2 = -\sum_{x \in A_{X2}} p_2(x) \cdot \log_2 p_C(x)$$ p2

$$AWL_3 = -\sum_{x \in A_{X31}} p_3(x) \cdot \log_2 p_C(x)$$ p3



Matlab code to get the PMF (pixel value 0~255):

PMF = hist(im(:), lower_bound:upper_bound);   PMF = PMF/sum(PMF);

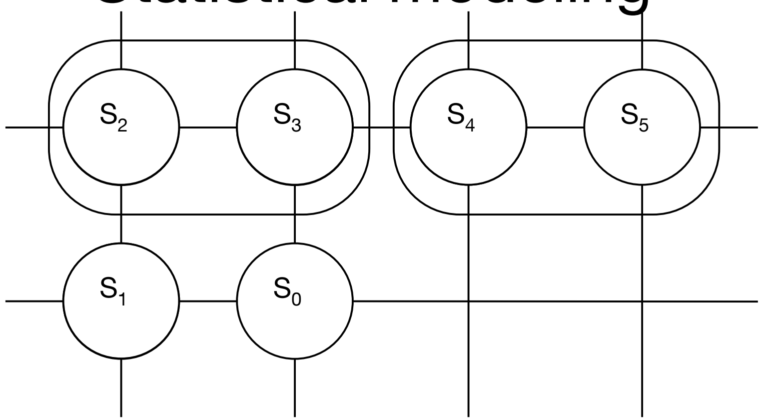
Note: the lower_bound and upper_bound are not for a specific image, but for the general case (e.g. 0~255 for RGB values, -128 ~ 255 for YCbCr values).

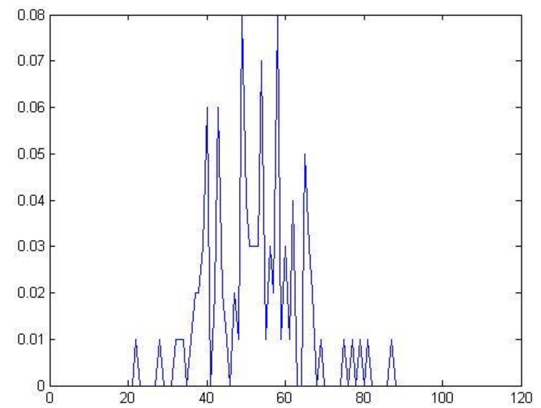
Joint entropy

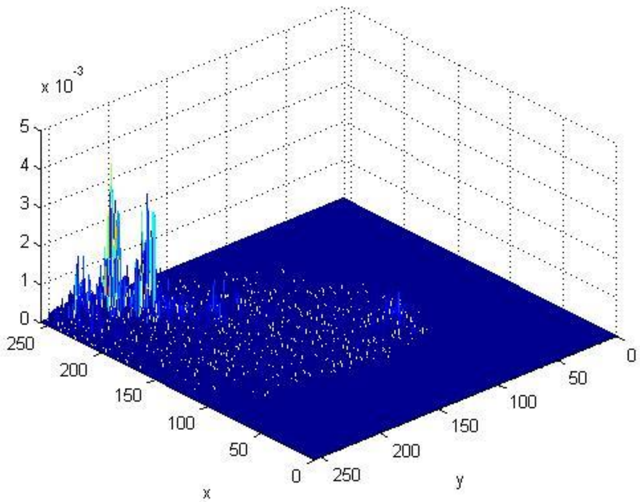
- Definition

$$H(X,Y)$$
$$= E\left[-\log_2 p_{X,Y}(x,y)\right]$$
$$= -\sum_y \sum_x p_{X,Y}(x,y) \log_2 p_{X,Y}(x,y)$$

- Statistical modeling







PMF(S2, S3) = PMF(S2, S3)+1;

Conditional entropy

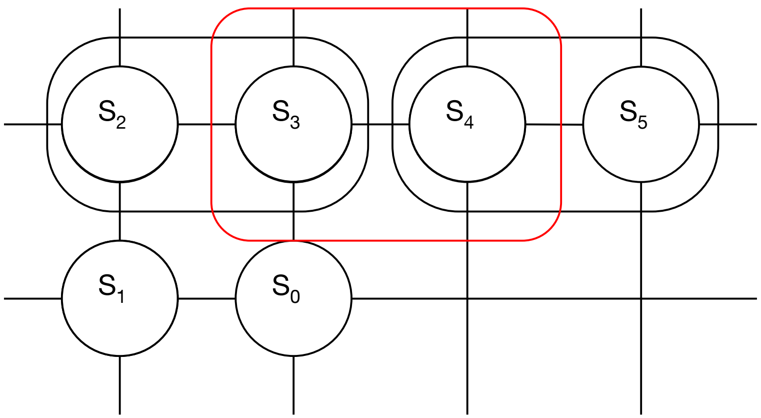
- Definition

$$H(X\,|\,Y) = E\left[-\log_2 p_{X|Y}(x,y)\right] = -\sum_y \sum_x p_{X,Y}(x,y)\log_2 p_{X|Y}(x,y)$$

$$p_{X|Y}(x,y) = \frac{p_{X,Y}(x,y)}{p_Y(y)} = \frac{p_{X,Y}(x,y)}{\sum_{x_0 \in X} p_{X,Y}(x_0,y)}$$

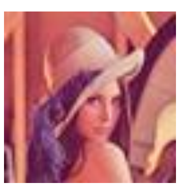
Normalize the $p_{X,Y}(x,y)$ in every column

- Statistical modeling



Huffman coding

- Example:
  - hist(im(:), lower_bound:upper_bound) → PMF
  - buildHuffman(PMF)
  - Use enc_huffman_new() and dec_huffman_new()
  - Bytestream = enc_huffman_new(data-lower_bound+1, BinCode, Codelengths)
  - Bitrate = length(bytestream)*8 / (image_width * image_height)
- Code table training



Huffman code table	
0	00000
1	0000110000
2	00001101
…	
N	00001100011111 0000110

Lossless predictive coding (E2-6)

- Encoder
  - Predictor P(S)=F(S1',S2',S3')
  - Residual E = S – P
  - Huffman coding E
- Decoder
  - Huffman decoding for E
  - Reproduce predictor P
  - Reconstruct S' = P + E

Pseudo code for encoder :

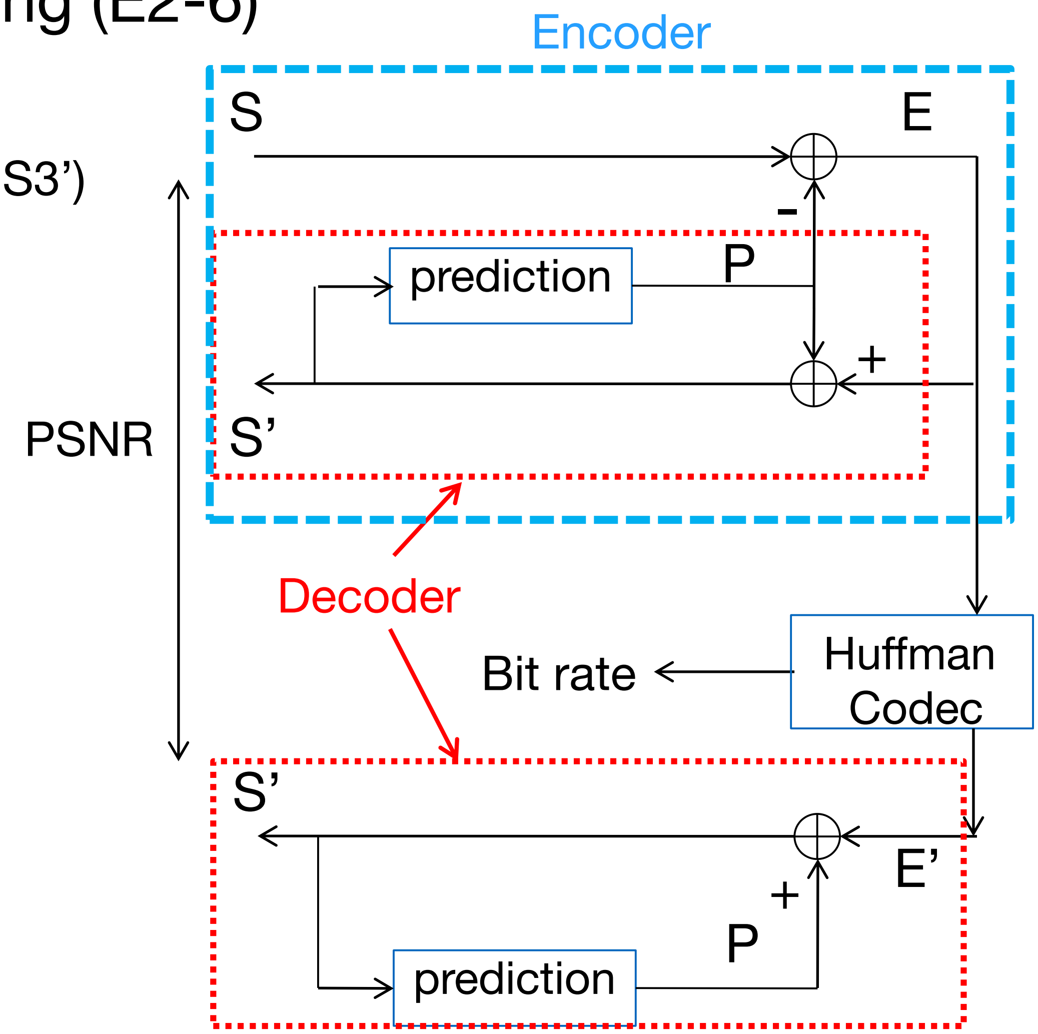
Loop: for each pixel

P = getPrediction(S')

E = S – P

E' = Encode(E)

S' = P + E'

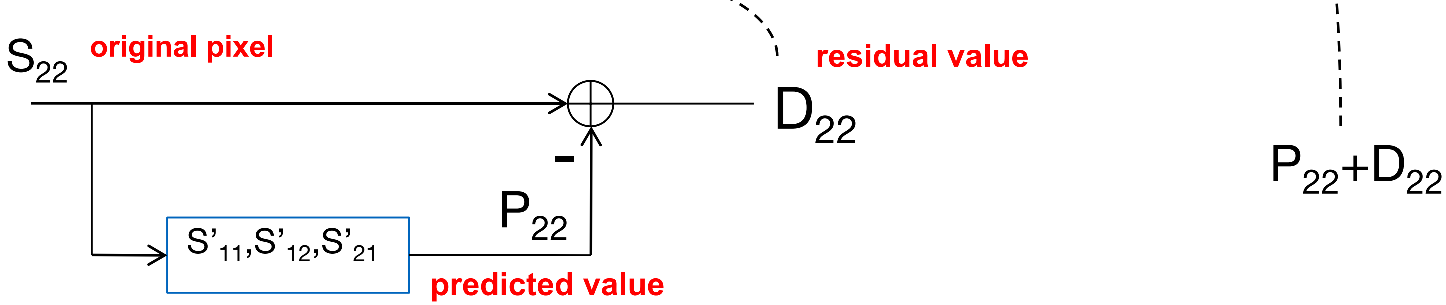


Lossless predictive coding: encoding

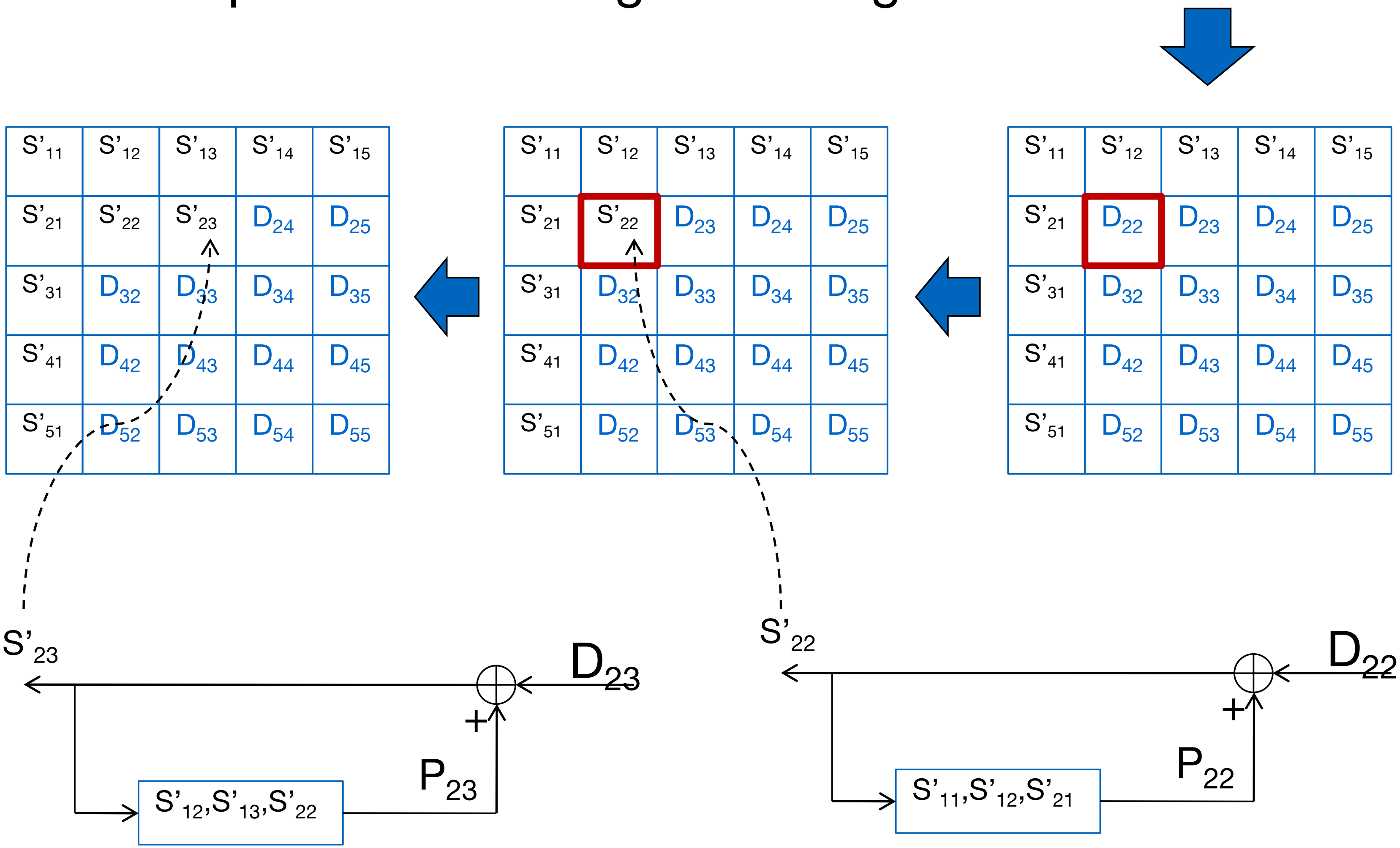
S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
S_{21}	$\boxed{S_{22}}$	S_{23}	S_{24}	S_{25}
S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
S_{21}	S_{22}	S_{23}	S_{24}	S_{25}
S_{11}	S_{12}	S_{13}	S_{14}	S_{15}

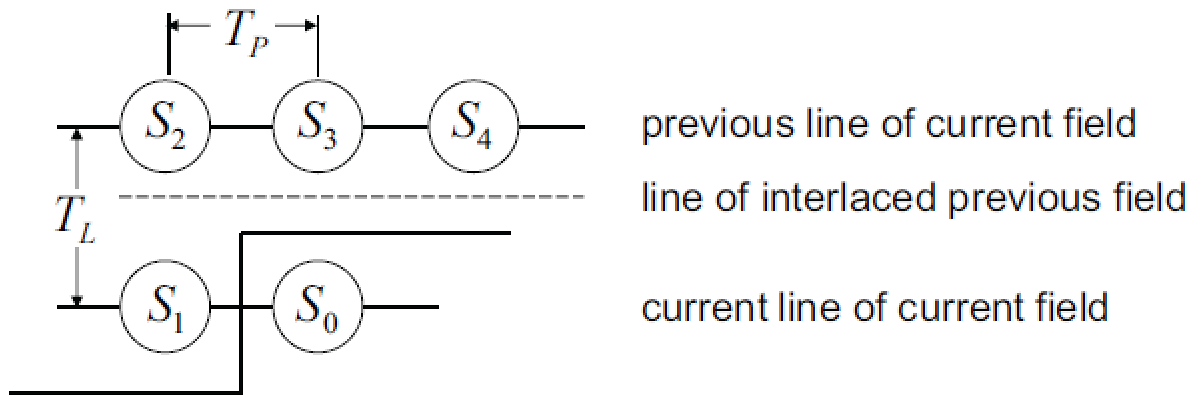
S'_{11}	S'_{12}	S'_{13}	S'_{14}	S'_{15}
S'_{21}	D_{22}	D_{23}	D_{24}	D_{25}
S'_{31}	D_{32}	D_{33}	D_{34}	D_{35}
S'_{41}	D_{42}	D_{43}	D_{44}	D_{45}
S'_{51}	D_{52}	D_{53}	D_{54}	D_{55}

S'_{11}	S'_{12}	S'_{13}	S'_{14}	S'_{15}
S'_{21}	$\boxed{S'_{22}}$			
S'_{31}				
S'_{41}				
S'_{51}				



Lossless predictive coding: decoding





Luminance signal Y

$H(S_0)$ [bit]	Predictor			$H(e)$ [bit]	Criterion
	a_1	a_2	a_3		
7.34	7/8	-5/8	3/4	4.30	minimum variance
734	7/8	-1/2	5/8	4.29	minimum entropy

Chrominance signal Cb Cr

$H(S_0)$ [bit]	Predictor			$H(e)$ [bit]	Criterion
	a_1	a_2	a_3		
5.57	5/8	-1/2	7/8	2.87	minimum variance
5.57	3/8	-1/4	7/8	2.82	minimum entropy