ARAVIND
2019102014

Lecture 21
Image Segementation Contd

→ $R_D = 1 - \frac{1}{C_R}$
↳ Compression Ratio

→ Spatial → Interpixel → Pixels are correlated

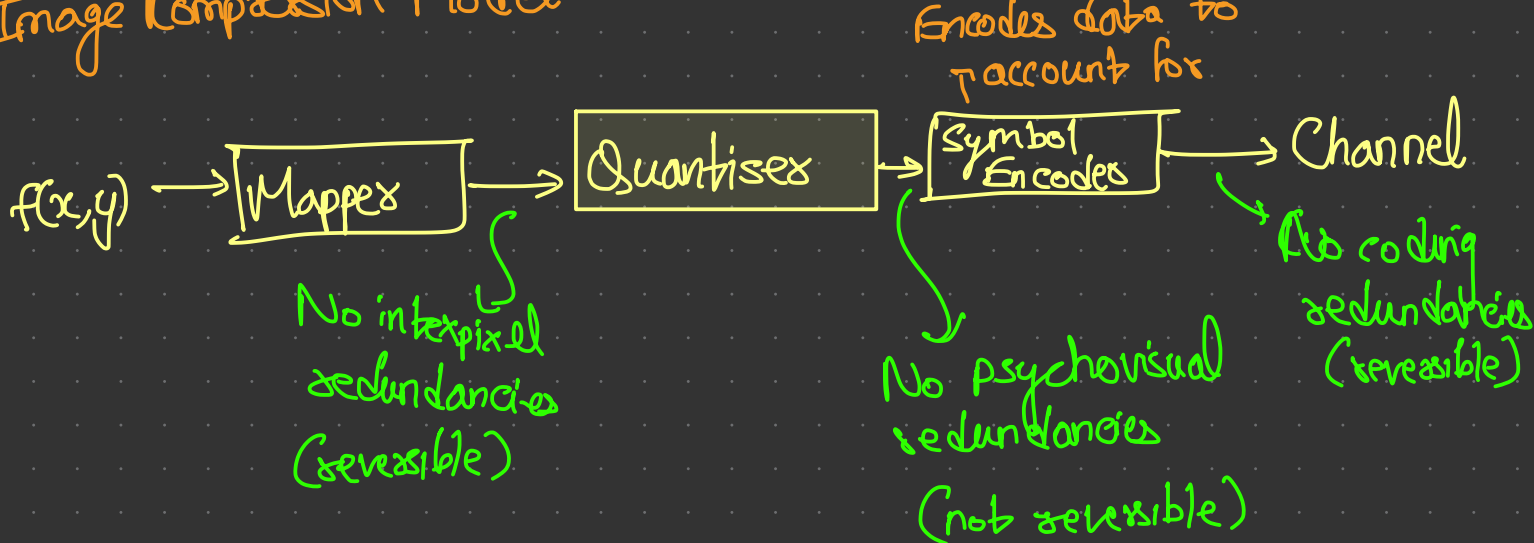
$$f \circ g = \int_{-\infty}^{\infty} f(x)\, g(x+a)\, da$$

↳ Auto correlation: $F(x) = g(x)$

→ Psychovisual → Visual / Perceptual redundancy

→ Shannon's Theorem:
- → Provides lower bound on bit rate for encoding statistically independent symbols
- → For finite markov, this may not hold as we can leverage on the redundacies

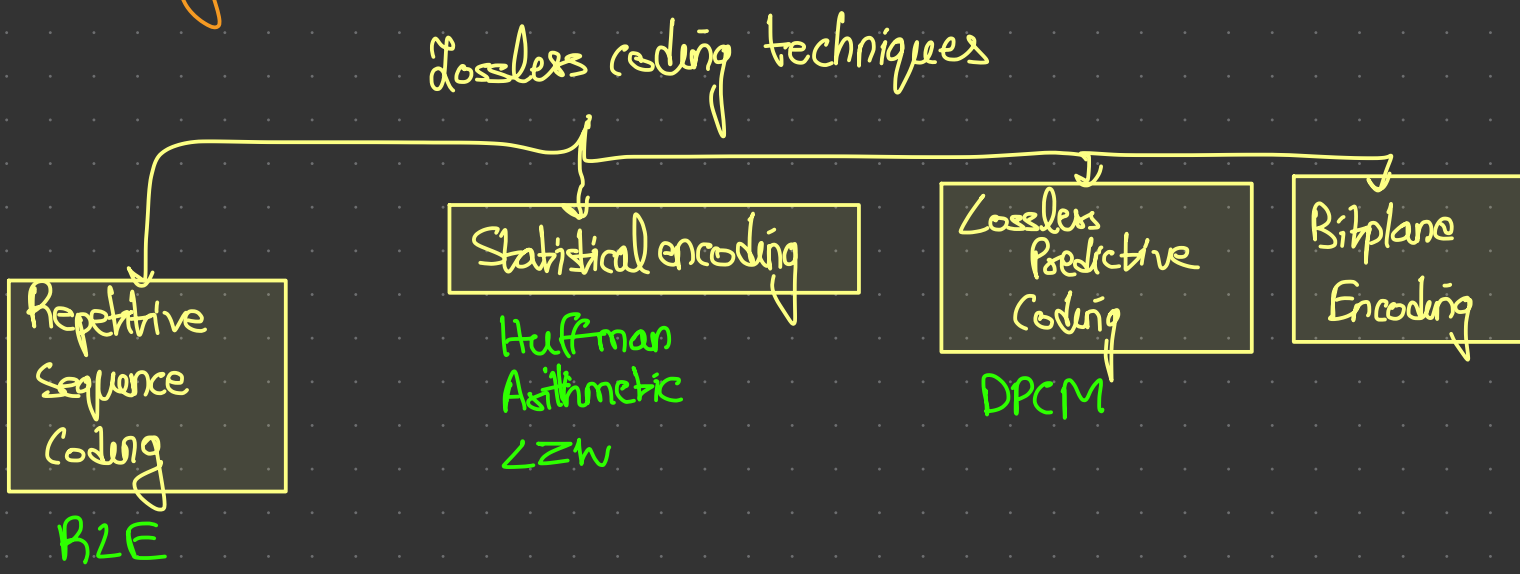
Image Compression Model



Lossless Compression

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

Taxonomy of lossless methods



Huffman Encoding [Addresses coding redundancy]

- → Variable length coding technique
- → Source symbols encoded one at a time
- → Optimal code - Minimise code word length per source symbol

Forward Pass

- → Sort prob. per symbol
- → Combine two lowest
- → Repeat step a till only 2 probs remain

Backward Pass

- → Assign code symbols going backwards

→ Implemented using lookup table to do decoding unambiguously

Arithmetic (Range) Coding

→ No one-one correspondence
→ Slower than Huffman
→ Better compression than Huffman
→ Sequence assigned to subinterval in [0,1) represented by arithmetic code
  ↳ Interval reduces with increase in number of symbols in message

LZW Coding

→ Codeblock (dictionary) need to be constructed
→ Initially first 256 entries are assigned directly

CS = CR + P
- CS: Concatenated Sequence
- CR: Currently Recognised Sequence
- P: Pixel being processed

→ Dictionary can be built as we go through sequence [real time]

CR = Empty
repeat
P = Next pixel
CS = CR + P
IF CS is found,
  (1) No output
  (2) CR = CS
Else:
  (1) Output D(CR)
  (2) Add CS to D
  (3) CR = P

Run-length coding (Interpixel redundancy)

→ Reduce size of repeating strings of symbols

→ Encode a run of symbols into 2 bytes (symbols, count)

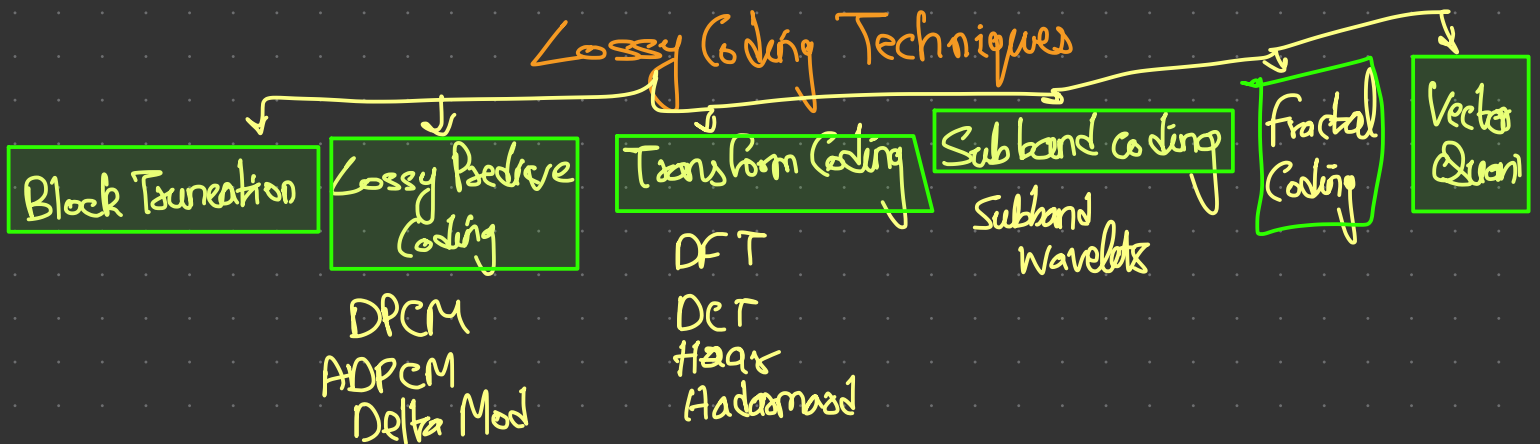
→ Can compress any type of data

Bit Plane Coding

→ Process each bit plane individually

↳ Decompose into series of binary image

↳ Compress each binary image (using eg run-length)



Transform the image into some other domain to reduce interpixel redundancy.

→ Do quant- in forward transform

→ Fourier Transform

Suppress high freq. coeff in freq. domain

Types of transforms:

→ DFT
→ DCT (Discrete Cosine Transform)
→ KLT (Karhunen-Loeve Transform)

DCT performs better and has a better RMS error in comparison to DFT

DCT minimises blocking artifacts

DCT → 2n-point periodicity
n-point periodicity

JPEG compression