Wavelets and Applications

Session 8: Compression

Pier Luigi Dragotti
EEE Dept, Imperial College London
p.dragotti@imperial.ac.uk

Outline

- What is Compression?
- Overview of Transform Coding
  - Entropy coder
  - Quantisation and Rate-Distortion function
  - Bit-allocation
  - Orthogonal and bio-orthogonal transforms
- Image Compression
- Video Compression

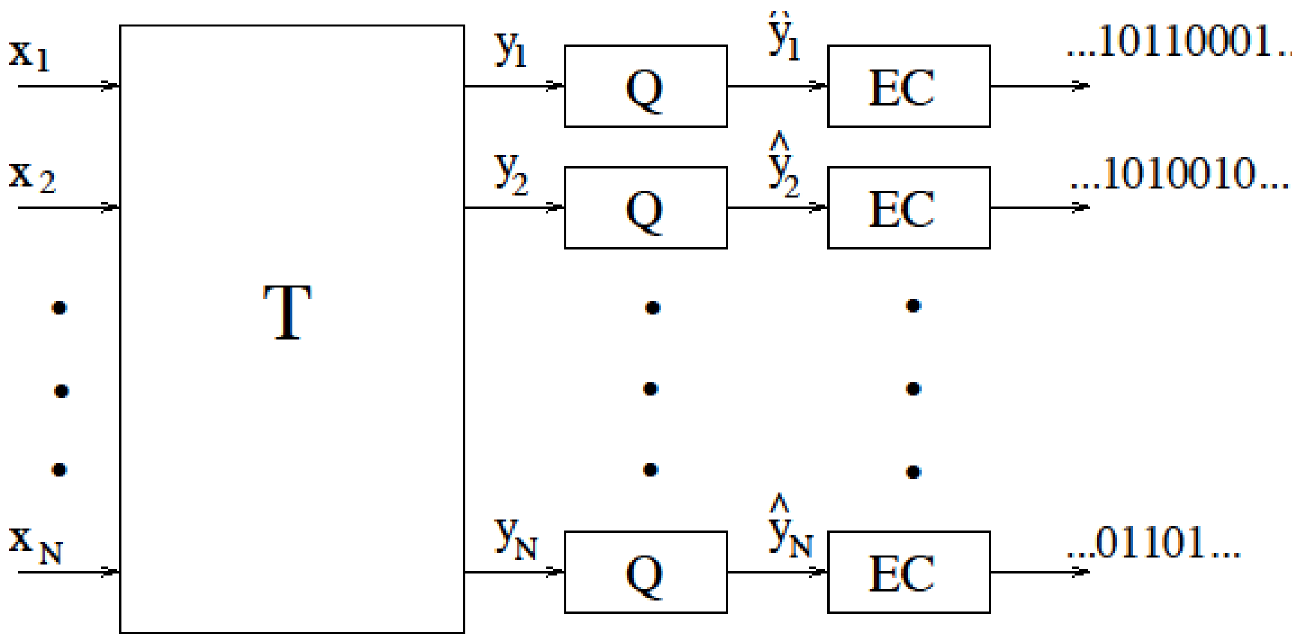
What is Compression?

- Compression deals with the problem of reducing the amount of memory necessary to store data or the problem of reducing the time and bandwidth necessary to transmit such data.

- There are two forms of compression
  - **Lossless compression** is reversible in that the original source can be reconstructed exactly
  - **Lossy compression** is not reversible and only approximate versions of the original source can be obtained
- Lossless compression is normally used for text and achieves a compression ratio of ~2
- Lossy compression is used for multimedia data (e.g., images, audio, videos) and achieves compression ratios higher than a factor 10

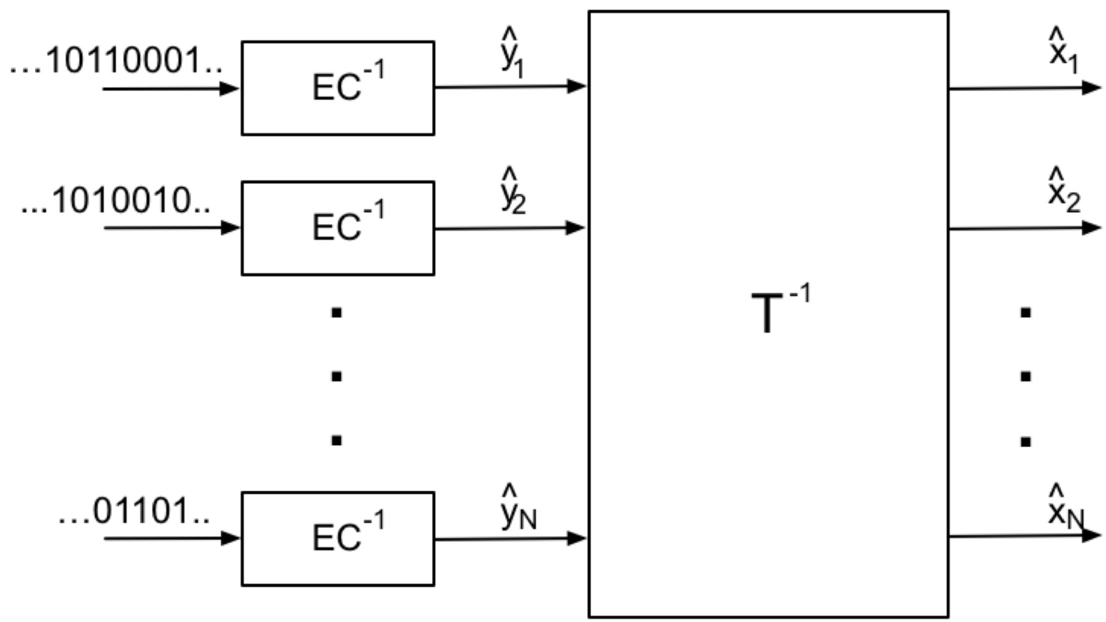
Overview of Transform Coding

- Typical multimedia compression systems have three main elements:
  - A block implementing a linear transform
  - A quantiser
  - An entropy encoder



Overview of Transform Coding (cont'd)

- Reconstruction is achieved by inverting the entropy coder (lossless compression) and the linear transform
- The reconstruction error is normally measured as: $D = \|x - \hat{x}\|^2$
- The goal is to minimize *D* for a given bit-budget *R* or to achieve a target distortion *D* with the minimum number of bits



Entropy Coder (lossless coder)

- Consider a source X that produces symbols a_i with probability of occurrence *p*(a_i), *i=0,1,..,M-1*.
- The aim is to find a binary representation per symbol with length l_i

  such that the expected length $E[(x)] = \sum_{i=0}^{M-1} p(a_i) l_i$ is minimized.
- *The binary representation must be:*
  - *Invertible*
  - *Uniquely decodable (no codeword is allowed to be the prefix of another coderword)*
- The lower bound on the expected length of a prefix code is given by the entropy of X:

$$H(x) = -\sum_{i=0}^{M-1} p(a_i) \log_2(p(a_i)) \text{ bits per symbol}$$

- *This is achieved when* $l_i = -\log_2 p(a_i).$

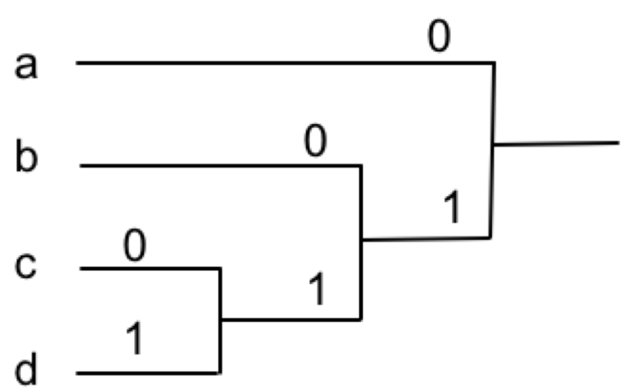
Entropy Coder Example (Huffman Code)

- Consider a source with the four symbols {a,b,c,d}
- p(a)=0.5; p(b)=0.25, p(c)=p(d)=0.125.
- Fixed length binary representation a → 00

  b → 01

  c →10

  d →11
- Variable length Huffman code:



a → 0
b→10
c→110
d→111

Entropy Coder Example (cont'd)

- Assume the source produces b,a,c,a,b,d,a,a
- Fixed length binary stream 0100100001110000 (average length 2bits/symbol)
- Variable length binary stream 10011001011100 (average length H(x)=1.75bits/symbols)

Entropy Coder Example (cont'd)

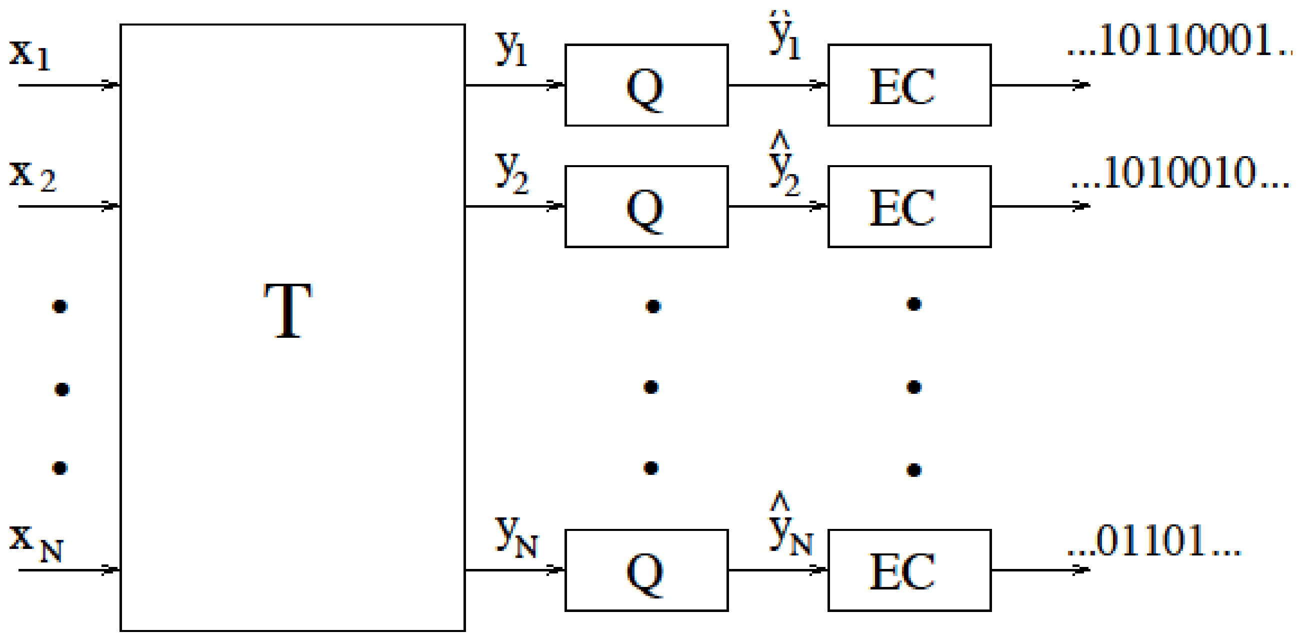
- Assume the source produces b,a,c,a,b,d,a,a
- Fixed length binary stream 0100100001110000 (average length 2bits/symbol)
- Variable length binary stream 10011001011100 (average length H(x)=1.75bits/symbols)

- Note that in both cases we can separate the symbols in the stream:
- Fixed length (easy) 01,00,10,00,01,11,00,00
- Huffman code is uniquely decodable:

  10,0,110,0,10,111,0,0

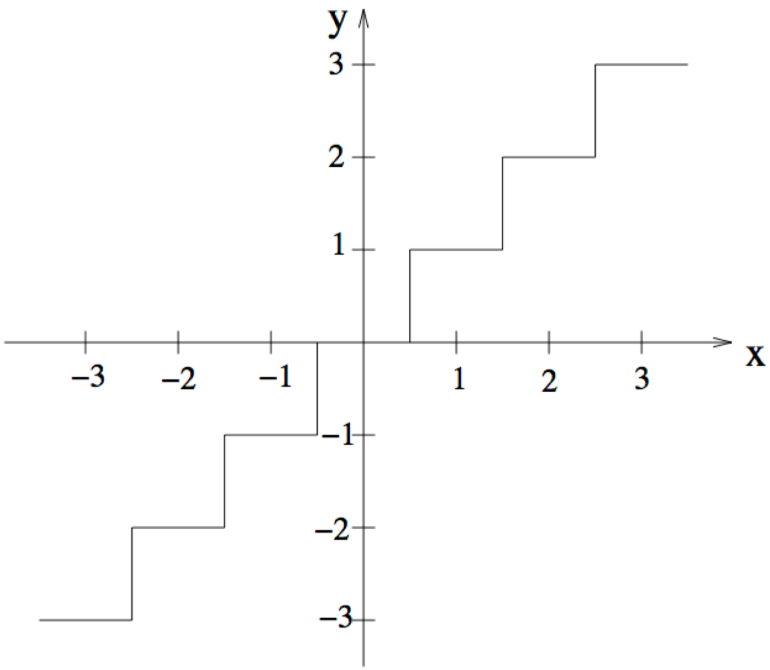
Scalar Quantisation

- Typically each sample y_i is quantized independently (scalar quantisation)
- Quantization is irreversible. This means that quantization introduces approximation errors (lossy compression).
- Usually the number of quantisation levels N=2^R so that each symbol can be expressed using a stream of bits.



Scalar Quantisation

- When the quantisation intervals have equal width we say that quantisation is ‘*uniform*’.



Distortion due to Quantisation

- Assume y_i is uniformly distributed over [-A,A] and i.i.d.
- Number of quantization levels $N=2^R$
- We have that quantisation interval $\Delta=2A/N=2A2^{-R}$
- Power of y_i : $\sigma^2=4A^2/12$.
- Quantization error: $E[(y_i - \hat{y}_i)^2] = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} y^2 dy = \frac{\Delta^2}{12} = \sigma^2 2^{-2R}$
- Which yields the operational distortion-rate function $D(R) = \sigma^2 2^{-2R}$
- For arbitrary sources and high rate:

$$D(R) = c\sigma^2 2^{-2R}$$

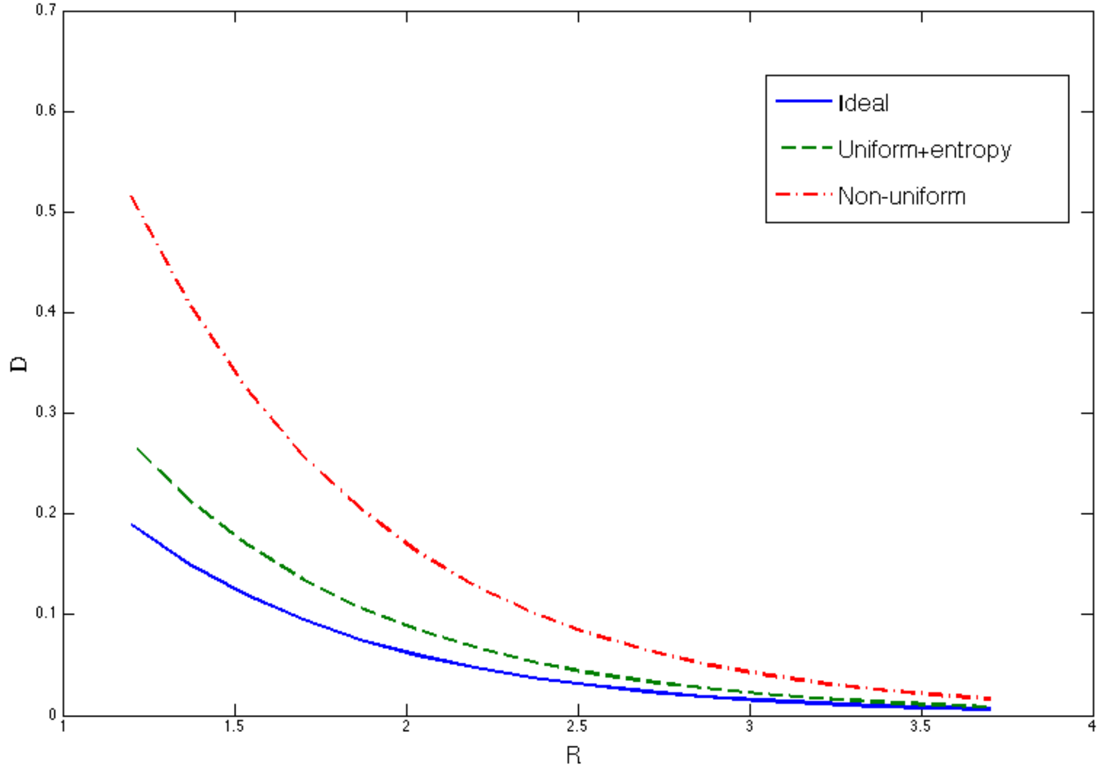
Entropy-Constrained Quantisation

- When y_i is not uniformly distributed, a non-uniform quantiser might be more appropriate.
- At the same time, an entropy encoder may compensate any sub-optimality
- Fundamental result in high-rate quantization theory:

'At high rate (large R), a uniform quantiser followed by an entropy encoder is optimal'

D(R) Curves examples

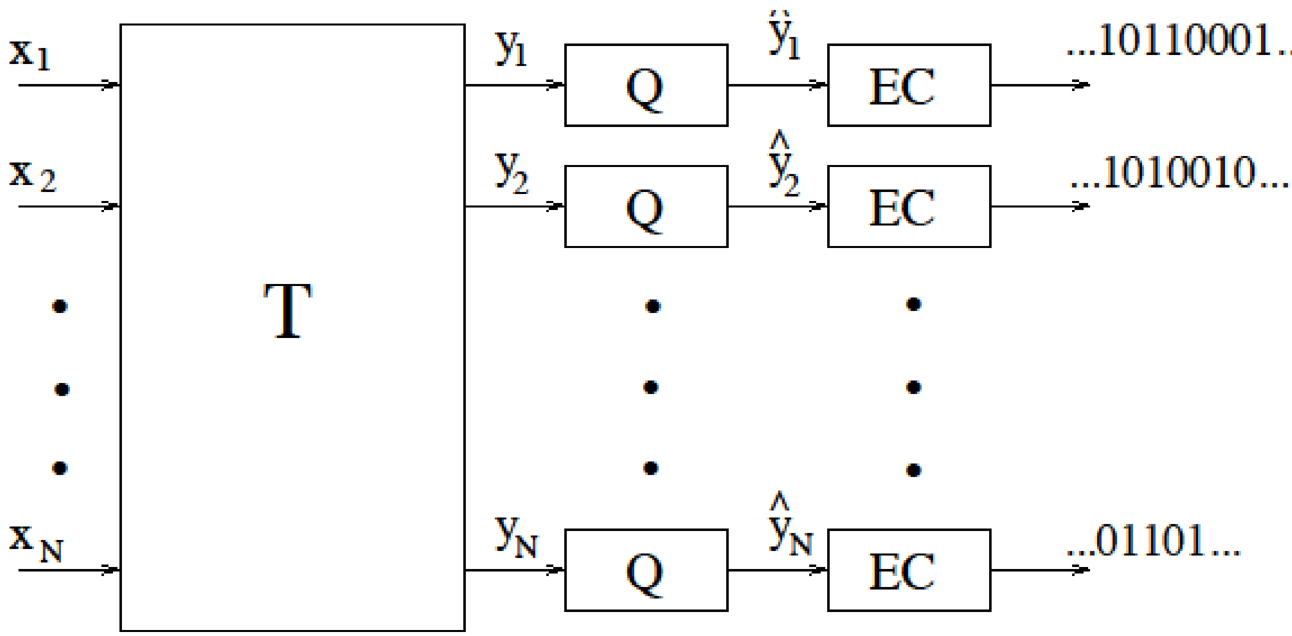
- Gaussian source with $\sigma^2=1$. Note: $$D = \left\| x - \hat{x} \right\|^2$$



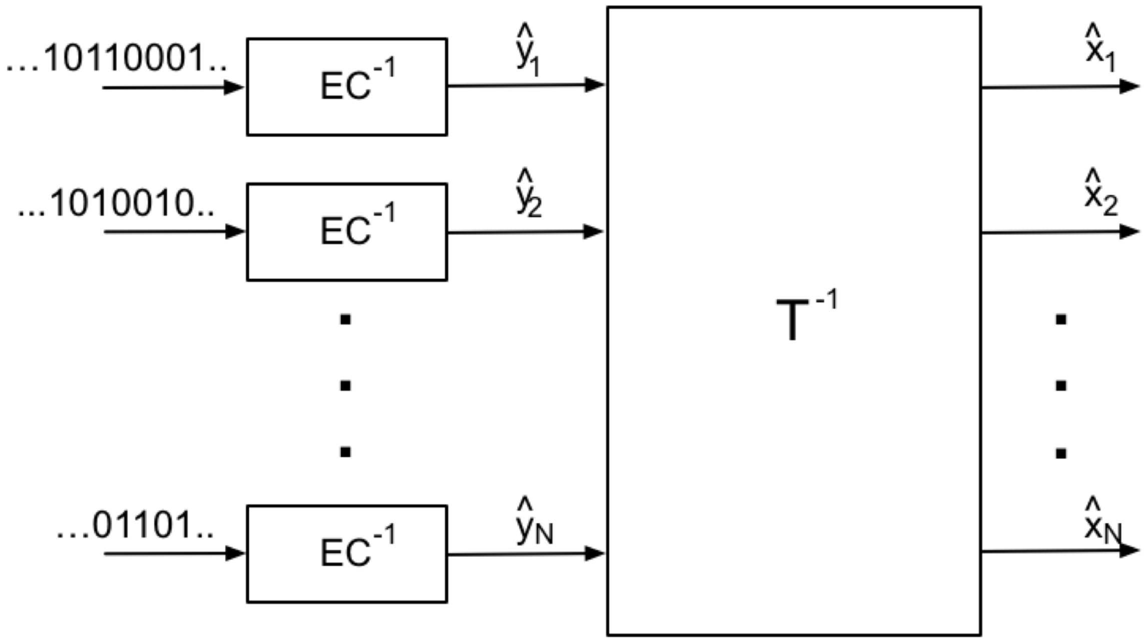
Bit allocation

- Given y_i i=1,2,…,N and a total bit budget R, how should we allocate the bits to the components in order to minimize the global distortion given by $D(R) = \sum_{i=1}^{N} D_i(R_i)$?
- Find $\min \sum_{i=1}^{N} D_i(R_i) + \lambda \sum_{i=1}^{N} R_i$
- Since $D_i(R_i) = c_i \sigma_i^2 2^{-2R_i}$
- We have that $R_i = \frac{R}{N} + \log_2 \sigma_i - \frac{1}{N} \sum_{k=1}^{N} \log_2 \sigma_i$

Choice of the Right Transform



Choice of the Right Transform



Choice of the Right Transform

- Ideally T should be an orthogonal transform (i.e. $T^T T=I$), because

$$\|x - \hat{x}\|^2 = \|y - \hat{y}\|^2$$

- Discrete Cosine Transform (DCT), Karhunen-Loeve Transform (KLT) and some wavelet transforms are examples of orthogonal transforms

- When $T^T T \neq I$ the transform is said to be biorthogonal and

$$A\|x - \hat{x}\|^2 \leq \|y - \hat{y}\|^2 \leq B\|x - \hat{x}\|^2$$

where A is the minimum eigenvalue of $T^T T \neq I$ *and B is the maximum eigenvalue.*

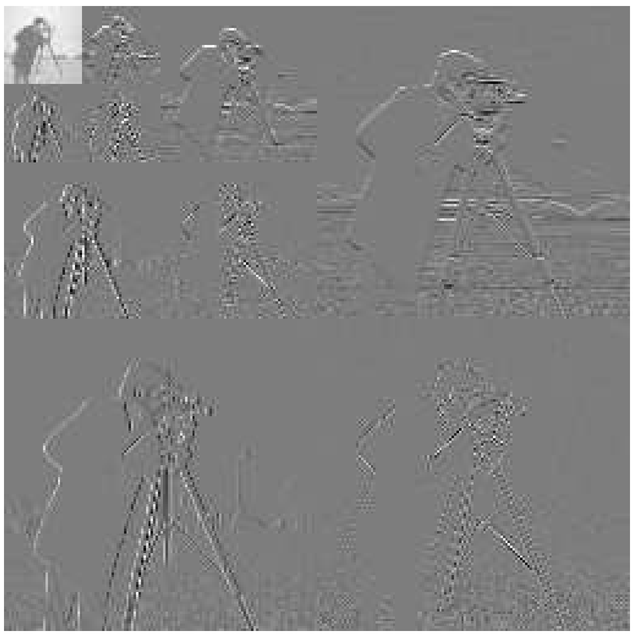
Choice of the Right Transform

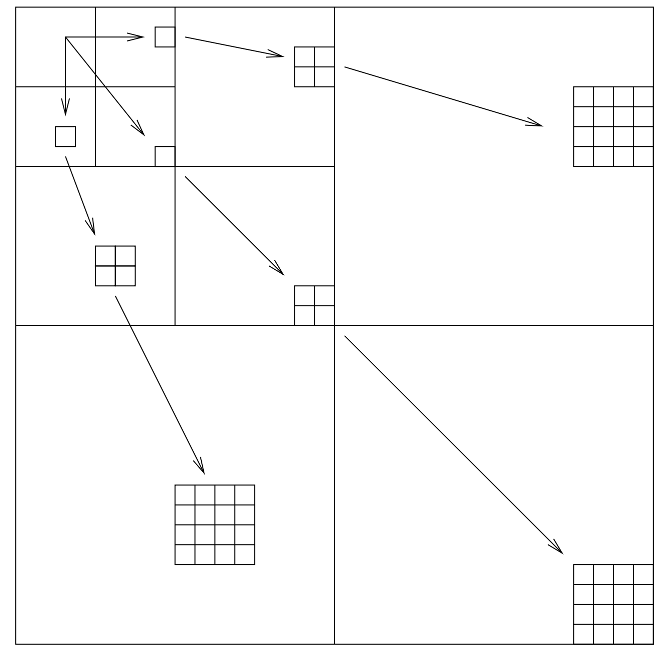
- T is chosen so that most of the energy of the incoming signal x is compacted into a small number of coefficients y (compact or sparse representation).
- KLT good for Gaussian sources
- Wavelets good for images
- DCT (DST) good for video

Image Compression

- Wavelets provide a sparse representation of images

Image Compression





Zerotree algorithm.

Image Compression

63	−34	49	10	7	13	−12	7
−31	23	14	−13	3	4	6	−1
15	14	3	−12	5	−7	3	9
−9	−7	−14	8	4	−2	3	2
−5	9	−1	47	4	6	−2	2
3	0	−3	2	3	−2	0	4
2	−3	6	−4	3	6	3	6
5	11	5	6	0	3	−4	4

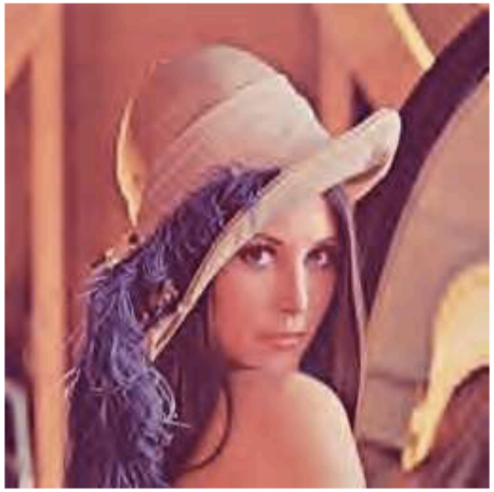
Coefficient	Symbol	Reconstruction
63	POS	48
-34	NEG	-48
-31	IZ	0
23	ZTR	0
49	POS	48
10	ZTR	0
14	ZTR	0
-13	ZTR	0
15	ZTR	0
14	IZ	0
-9	ZTR	0
-7	ZTR	0
7	Z	0
13	Z	0
3	Z	0
4	Z	0
-1	Z	0
47	POS	48

Image Compression

Wavelets are in the new image compression standard (JPEG2000)

Original Lena Image
(256×256 pixels)

JPEG (Compression Ratio
43:1)

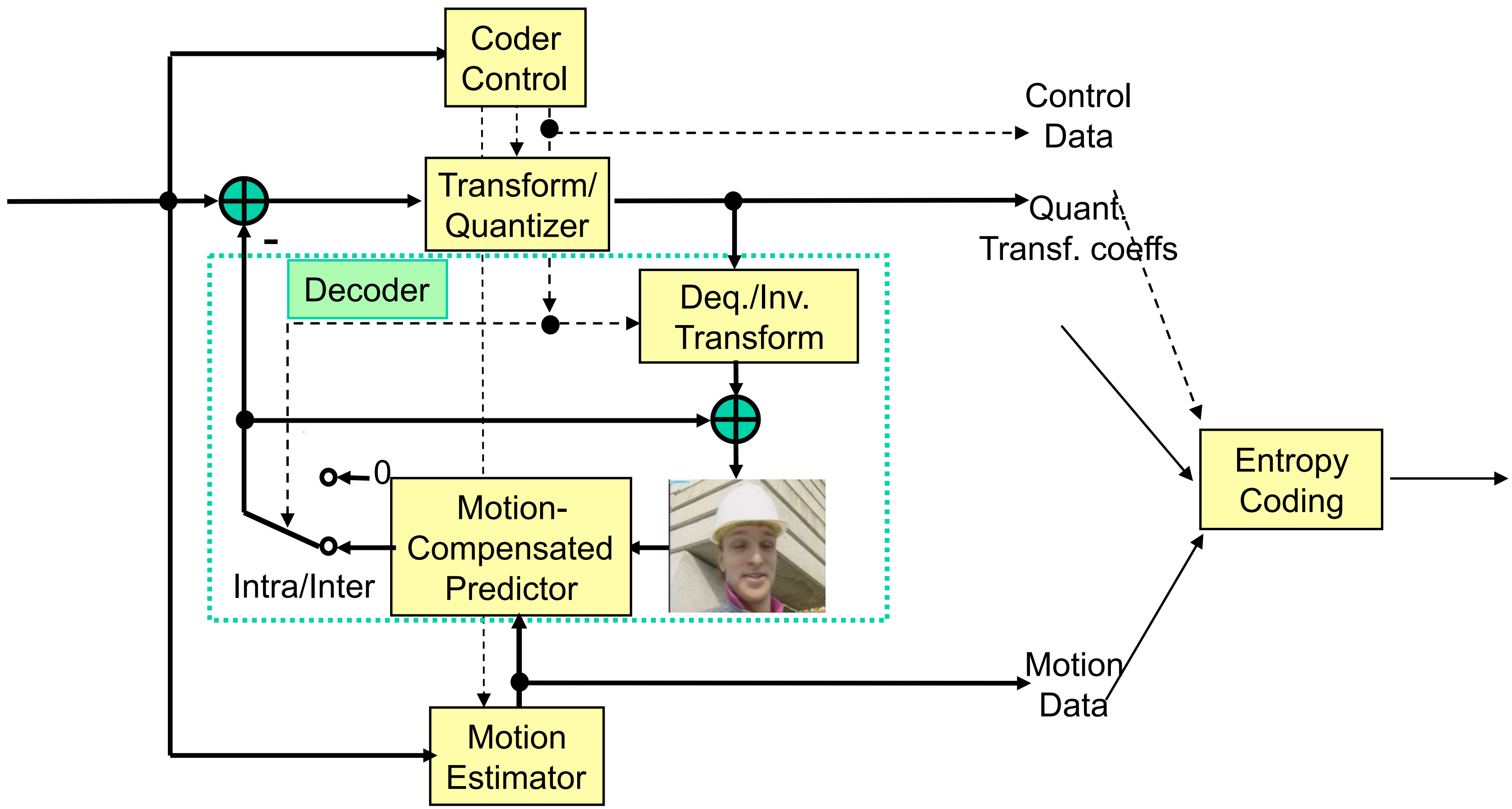


JPEG2000 (Compression
Ratio 43:1)

Note: images courtesy of dspworx.com

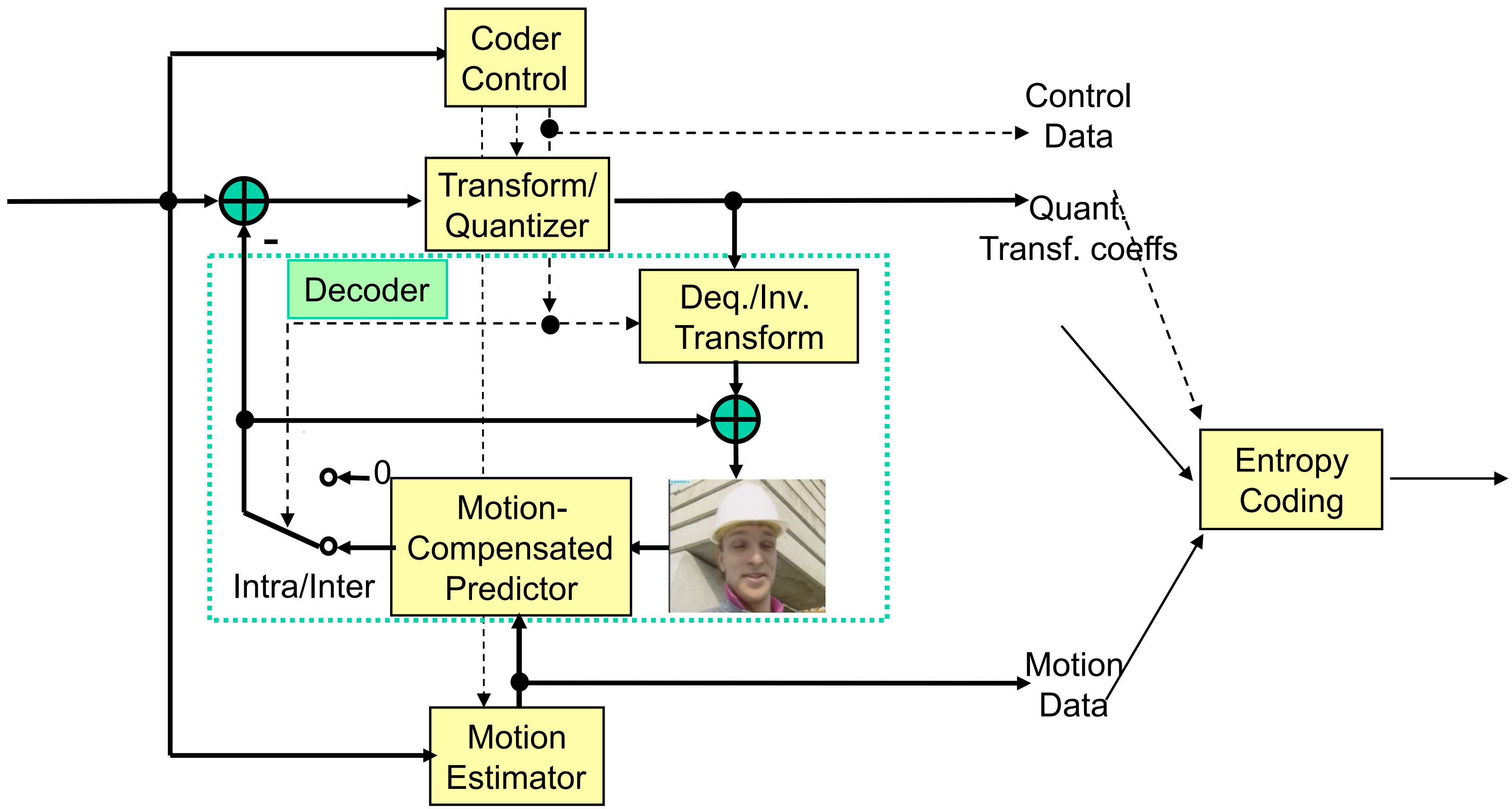
Video Compression

- Redundancy in space and time → Sparse representation in both space and time
- Redundancy in time removed through motion compensation



Video Compression

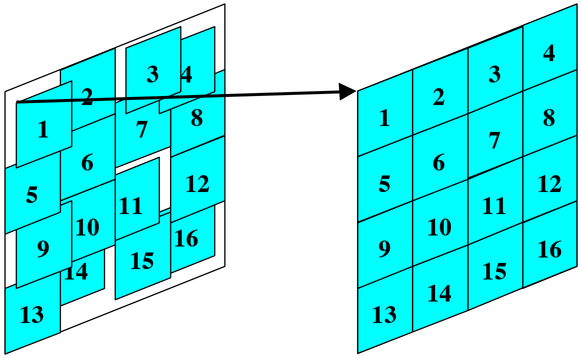
- Mpeg1/2, H.261, H.263, H.264, HEVC all based on the same principles



Motion Prediction

Previously Coded Frame
(Reference Frame)

Current Frame
(To be Predicted)



Reference Frame ***Predicted Frame***

Note: Images courtesy of Dr J. Apostolopoulos (HP labs)

Motion Prediction

Prediction of
Current Frame

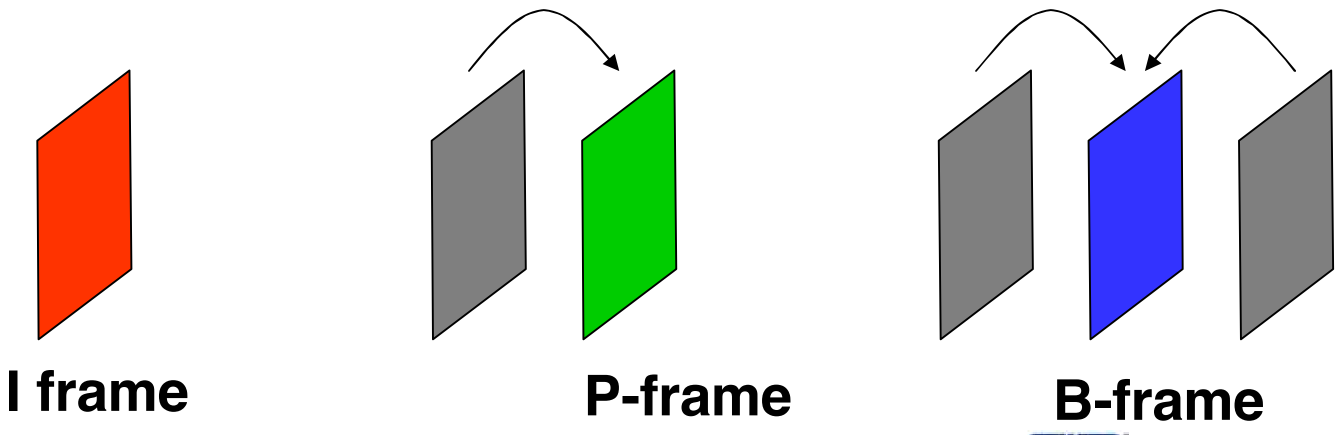
Prediction Error
(Residual)

Note: Images courtesy of Dr J. Apostolopoulos (HP labs)

Video Compression Terminology

I-frame: Intra coded frame. Frame compressed independently of all the others
P frame: Predictively coded frame, coded based on previously coded frame
B frame: Bi-directionally predicted frame, coded based on both previous and future coded frames

Group of pictures (**GOP**): number of frames between two **I**-frames



I-frame Coding

- Similar in spirit to Jpeg (not Jpeg 2000)
- 8x8 DCT
- Arbitrary weighting matrix for coefficients
- Differential coding of DC-coefficients
- Uniform quantization
- Zig-zag-scan, run-length coding
- Entropy coding
- Please note that H.264 use also intra-prediction.

Discrete Cosine Transform (DCT)

Definition of the 1-D DCT:

- Let A be the DCT matrix of size NxN then

$$A_{n,m} = b(n)\sqrt{\frac{2}{N}}\cos\frac{\pi n(m+0.5)}{N} \qquad 0 \le n,m < N$$

with

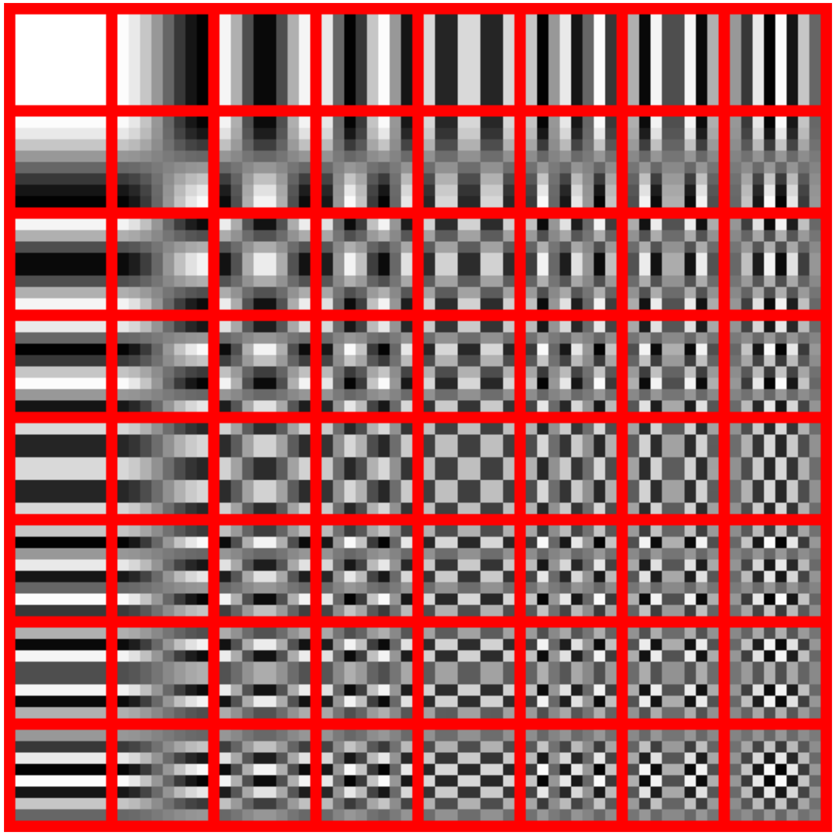
$$b(n) = \begin{cases} 1/\sqrt{2} & n = 0, \\ 1 & 1 \le n < N. \end{cases}$$

2-D Discrete Cosine Transform

- Assume X is a matrix whose entries correspond to the pixel values of an NxN block of an image
- The 2-D transform is first performed on rows and then on columns of the image.
- In matrix form $A (A X^T)^T = A X A^T$
- Therefore each basis element of the 2-D transform is given by:
- $A_i A_j^T$ where A_i is the i-th column of A.
- Note that video compression standards use an integer version of the DCT transform.

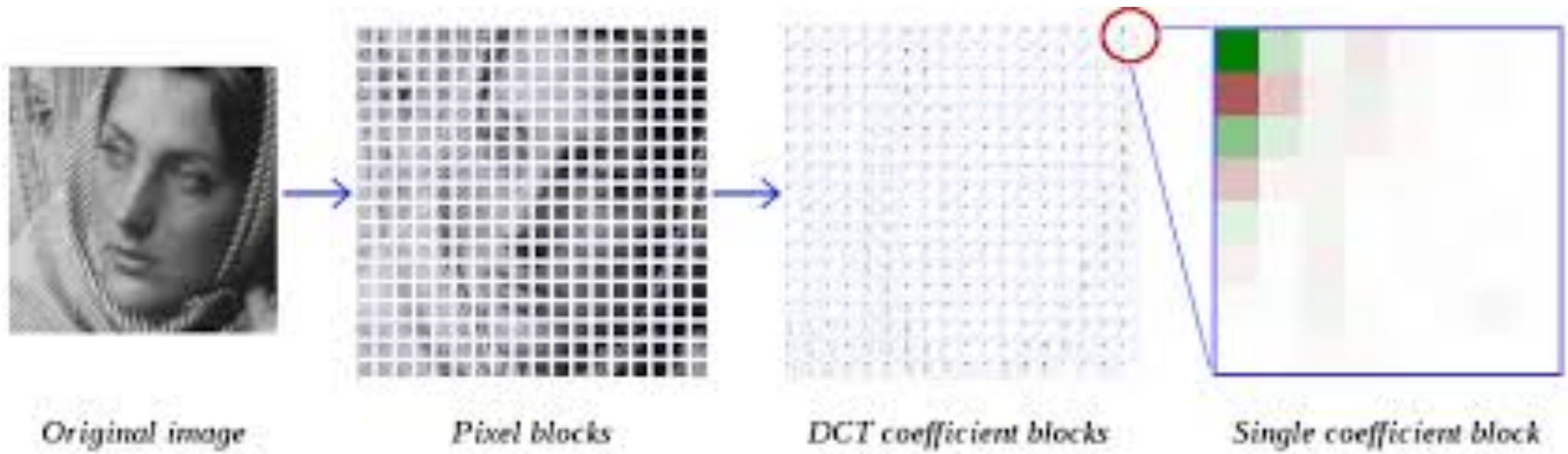
2-D Discrete Cosine Transform (cont'd)

64 bases of the 8x8 DCT



Note: Figure taken from *http://en.wikipedia.org/wiki/JPEG*

DCT of an Image



Note: Figure taken from *xiph.org*

Quantisation

- In Jpeg and in the video compression standards, a weighting matrix is employed so that more bits are used to quantise low-frequency coefficients.
- This is equivalent to the bit allocation we discussed before.
- The quantized DCT coefficients are computed as follows:

$$b_{j,k} = round\left(\frac{g_{j,k}}{q_{j,k}}\right)$$

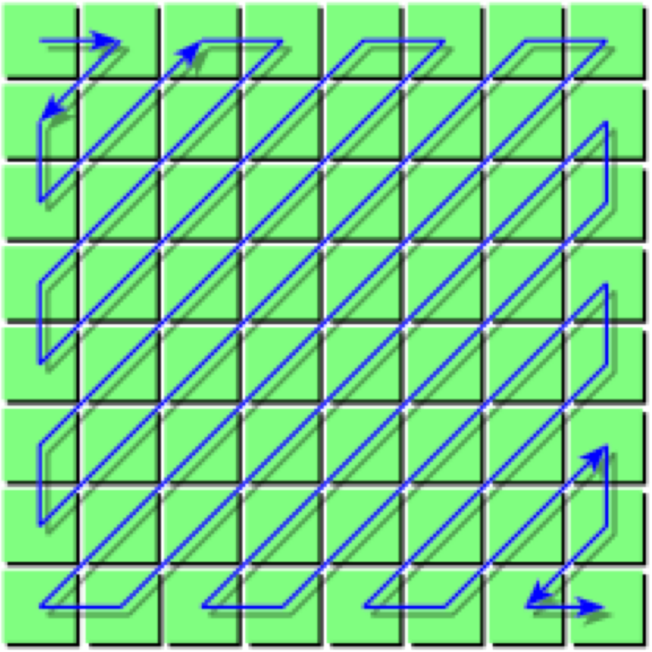
where $g_{j,k}$ is the un-quantised DCT coefficient and $q_{j,k}$ is one entry of the quantisation matrix

Quantisation (cont'd)

- An example of the quantisation matrix

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}.$$

I-frame Coding



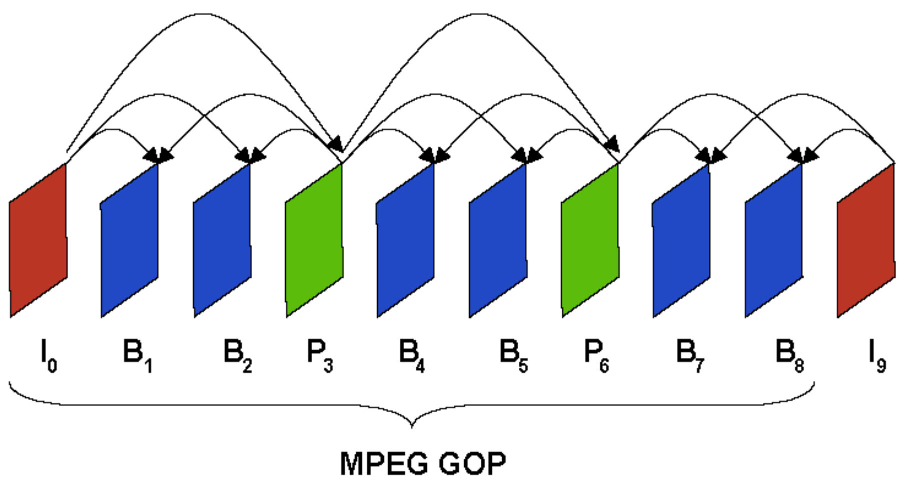
Zig-zag encoding

Run length coding
– Many coefficients are zero after quantization
– Run length Representation:
  - Ordering coefficients in the zig-zag order
  - Specify how many zeros before a non-zero value

Note: Figure taken from *http://en.wikipedia.org/wiki/JPEG*

P-frame and B-frame Coding

- Prediction:
  - Search for a block in a reference frame that has the lowest matching error (lowest sum of absolute errors between pixels)
  - Motion estimation at sub-pixel resolution
  - Hierarchy between I,P and B frames shown below



Video Compression Standard

- Mpeg1/2, H.261, H.263, H.264 all based on the same principles described before.
- Most of the coding-efficiency improvement happened at the motion estimation stage
- H.264/AVC (current standard) uses:
  - YCbCr colour space and 4:2:0 sampling (i.e., half vertical and half horizonal resolution for Cb and Cr).
  - variable-size blocks for motion estimation (e.g., 16x16, 8x8, 4x4 and rectangular blocks),
  - sophisticated motion prediction with sub-pixel precision (1/4)
  - Context Adaptive Binary Arithmetic Encoder (CABAC)
- The new standard under development will use more sizes for blocks, DCT or DST, further improved motion estimation and entropy encoders.