

Video Source Coding

Focus of the lecture

- Still image transform & compression (JPEG)
- Moving picture compression (MPEG)
 - temporal redundancy
 - group of pictures
 - motion prediction and compensation

Image Compression

- Redundancy
 - within a frame
 - between frames
 - in the bit stream
- Irrelevance
 - relevance of image and motion information to human vision sense
 - Eye is more sensitive to brightness than to colour.
- Aiming for compression ratios 1:100 or higher
 - How can this be achieved? Stay tuned...

DTV Compression (MPEG)

- Redundancy
 - Differential Pulse Code Modulation (DPCM)
 - Huffman coding
- Irrelevance
 - reducing colour (chrominance) resolution
 - image source coding: Discrete Cosine Transform (DCT), Hadamard transformation, image DSP
 - quantization

Image Coding: Luminance & Chrominance

- Human visual sensory system
 - is more sensitive to brightness (luminance) than to colour (chrominance)
 - led to black-and-white (B/W) TV: 5 MHz bandwidth
- Colour TV
 - adds only about 1.3 MHz extra to luminance band
 - Colour signals are made up of Red, Green and Blue.
 - must be backwards compatible with B/W TV sets
 - Rather than sending RGB and brightness, the property of luminance was determined as a weighted mix of RGB (determined by experimentation on humans)

$$Y = 0.3 R + 0.59 G + 0.11 B$$

$$C_b = 0.49 (B - Y)$$

$$C_r = 0.88 (R - Y)$$

Image Coding: Luminance & Chrominance

- Luminance (perceived brightness) and chrominance (perceived colour tone) signals are encoded as Y (luminance) and as C_b & C_r (chrominance w.r.t. Y)
 - If one would also transmit green, this would lead to redundancy with Y
- Chrominance is only sampled every second pixel along a line and only every second line
 - is known as “4:2:0”, i.e., for every 4 Y pixels there are 2 C_b and 2 C_r pixels in one line and 0 of each C in the next line
- A frame of the video therefore becomes two arrays, one for luminance and one for chrominance

Chrominance

- **MPEG-2 sampling structures for progressive sampling**

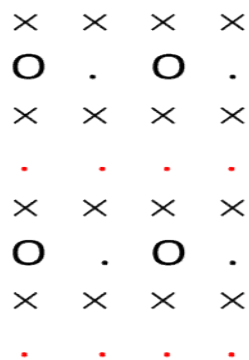
- **4:4:4 (progressive sampling);**



- **4:2:2 (progressive sampling)**



- **4:2:0 (progressive sampling)**



Note: X = luminance, O = chrominance, . = no chrominance

More complicated schemes for interlaced sampling.

There are more variants, e.g., interchange format in JPEG, SIF in MPEG-1, etc. See Reimers p. 80

Bit Resolution

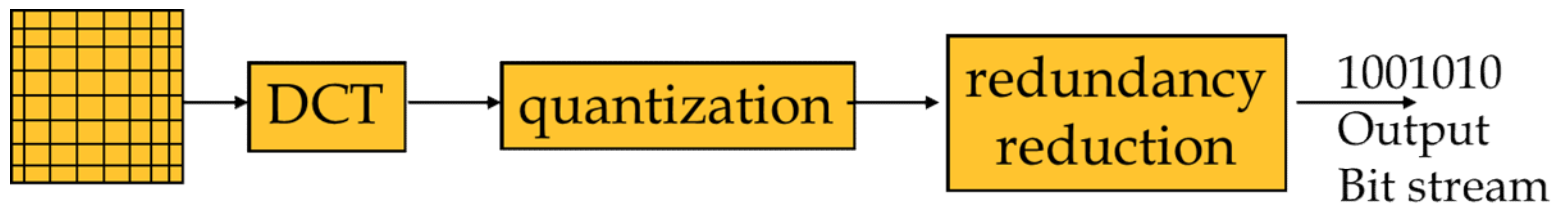
- Original studio recordings (Master copy) are in 10-bit resolution
 - Want these to be as high quality as possible – stored on disk so space not a problem.
- Experiments show that human perception using just 8 bits is satisfactory
 - Yields 20% data reduction
 - This data is lost, it can never be recovered at the receiver
- Further reduction by irrelevance

Omitting VBI/HBI

- HBI/VBI carry no useful information in digital TV (matrix LCD/LED technology)
 - was needed in analogue CRT sets to allow for sweeping electron beam back to beginning of next line (HBI) or back to top left of new image field (VBI)
 - In analogue signals the VBI contained teletext.
 - In digital signals this data is sent separately in the extra bandwidth available so the VBI carries nothing at all.
- HBI savings: $12 \mu\text{s} / 64 \mu\text{s} = 19\%$
- VBI savings: $50 \text{ lines} / 625 \text{ lines} = 8\%$
- Total: $\sim 25\%$ [due to some overlap]

Frame Encoding

- Split image into 8×8 pixel blocks (*macroblocks*)
- Use 2-dimensional Discrete Cosine Transform (DCT) to convert representation of each block from spatial domain to (spatial) frequency domain
- Quantize highest transformed coefficients only (lossy encoding)
- Reduces redundancy in bit stream
 - natural images have mostly low spatial frequencies
 - high spatial frequencies are rare (e.g., pin striped suit)

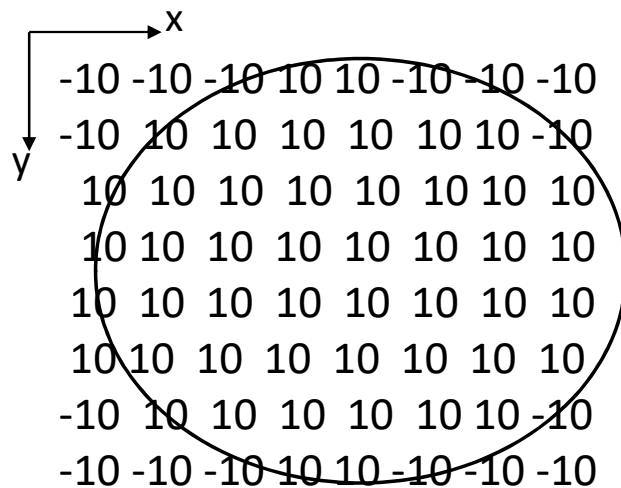


Frame Encoding

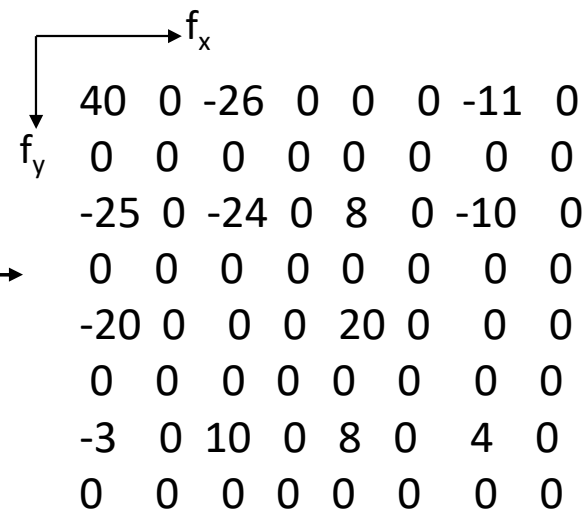
- Quantisation levels can change depending on how much we want to compress the image.
- More compression = coarser quantisation – worse quality
- Less compression = fine quantisation – better quality.

Discrete Cosine Transform

- DCT has same *number* of coefficients (lossless transformation), but the *values* will be distributed differently



Pixel values – spatial domain



DCT transform to frequency domain

Frequency and Image processing

- High frequency = fine detail (abrupt change);
low frequency = large area (slow change)
- Top left coefficient = DC component (constant for whole macroblock:
carries the most information for a single block.

Redundancy Reduction

- First step: convert 2-D array to 1-D vector
- Done by scanning in zig-zag manner
 - this orders coefficients from lowest to highest frequency
 - this ensures that any patterns occur in the bit-stream and are not 'broken up'

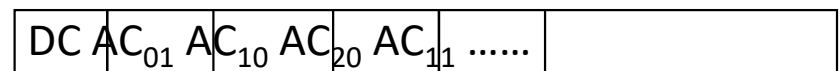
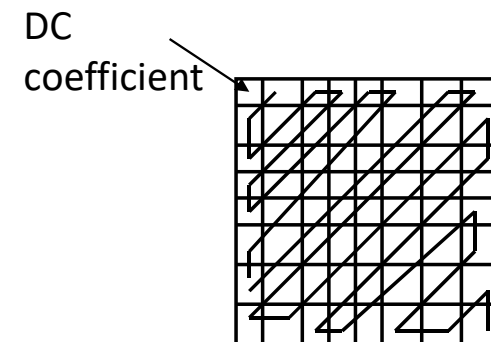


Image Quantisation: Quantisation Table

- Coarse quantization is used at high frequencies
- Quantization depends on image
- Quantization tables are constructed and then transmitted
 - one table can apply to sequence of several images
 - quantization info is contained in JPEG file
- After quantization, most coefficients will be zero except those towards the top left corner of the $n \times n$ square of DCT coefficients
 - use run-length coding for greater efficiency

Reconstruction

- Reconstruction of image: by
 - Inverse quantization, followed by
 - Inverse DCT
- Reconstructed image will not be identical to original, but differences should be almost imperceptible, provided the quantization step is not too large (detail)

DCT & Quantization Computation

- 2-D DCT for $N_x \times N_y$ image blocks:

$$G(f_x, f_y) = \frac{C_x C_y}{4} \sum_{x=0}^{N_x-1} \sum_{y=0}^{N_y-1} g(x, y) \cos \left[(2x+1) \frac{\pi f_x}{2N_x} \right] \cos \left[(2y+1) \frac{\pi f_y}{2N_y} \right]$$

where $C_x = \frac{1}{\sqrt{2}}$ if $f_x = 0$ and $C_x = 1$ if $f_x > 0$

– For $N_x = N_y = 8$:

$$G(f_x, f_y) = \frac{C_x C_y}{4} \sum_{x=0}^7 \sum_{y=0}^7 g(x, y) \cos \left[(2x+1) \frac{\pi f_x}{16} \right] \cos \left[(2y+1) \frac{\pi f_y}{16} \right]$$

- Quantized DCT cff.:

$$G_Q(f_x, f_y) = \text{round} [G(f_x, f_y) / Q(f_x, f_y)]$$

where $Q(f_x, f_y)$ = quantization step for $G(f_x, f_y)$

Example of DCT

- DCT quantization for luminance signal:
 - Pixels values of black and white ranges from 0 to 255
 - Pure black is represented by 0 and pure white by 255
 - $g = Y$; range = 0...255 \Leftrightarrow 8 bits for each $g(x,y)$
 - $G(0,0)$ needs up to $8+6+0-3 = 11$ bits:
 - sum of $8 \times 8 = 2^6$ coefficients, $\cos(0)=2^0$, $C_0 \cdot C_0/4 = 1/8 = 2^{-3}$
 - Quantized DCT cff.:

$$G_Q(f_x, f_y) = \text{round}[2^{11} / Q(f_x, f_y)]$$

- E.g. $G_Q(0,0) = \text{round}[2^{11} / 16] = 2^7 = 128$ amplitude levels,
 $G_Q(7,7) = \text{round}[2^{11} / 99] = 21$ amplitude levels

Coordinate Domain of Image (g)

- Example: given 8×8 luminance block $g(x,y)=Y(x,y)$:

139	144	149	153	155	255	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

Transform (Spectral) Domain: DCT

- 8×8 block of computed DCT coefficients $G(f_x, f_y)$:

235.6	-1.0	-12.1	-5.2	2.1	-1.7	-2.7	-1.3
-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
-1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4

Quantisation Table

- 8×8 block of given quantization coefficients $Q(f_x, f_y)$:

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	65
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	1001	103	99

- Note: coarser quantization at higher frequencies, e.g. $Q(7,7) \gg Q(0,0)$

Quantized DCT

- 8×8 block of quantized coefficients $G_Q(f_x, f_y)$:

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

- e.g. $G_Q(1,0) = \text{round}(-22.6/12) = \text{round}(-1.88) = -2$

Reconstructed DCT

- 8×8 inverse quantized coefficients $G^{-1}(f_x, f_y)$:

240	0	-10	0	0	0	0	0
-24	-12	0	0	0	0	0	0
-14	-13	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

- $-2 \times 12 = -24$

Reconstructed Image

- 8×8 estimated (reconstructed) luminance $\underline{g}(x,y)$:

144	146	149	152	154	156	156	156
148	150	152	154	156	156	156	156
155	156	157	158	158	157	156	155
160	161	161	162	161	168	157	155
163	163	164	163	162	150	158	156
163	164	164	164	162	160	158	157
160	161	162	162	162	161	159	158
158	159	161	161	162	161	159	158

- compare 148 vs. original 144 (nonquantized)

Reconstruction Error

- Absolute error: $\underline{g}(x,y) - g'(x,y)$:

-5	-2	0	1	1	-1	-1	-1
-4	1	1	2	3	0	0	0
-5	-1	3	5	0	-1	0	1
-1	0	1	-2	-1	-1	2	4
-4	-3	-3	-1	0	-5	-3	-1
-2	-3	-3	-3	-2	-3	-1	0
2	1	-1	1	0	-4	-2	-1
4	3	0	0	1	-3	-1	0

Temporal Redundancy

- Adjacent frames in a natural sequence differ only slightly from each other: *temporal redundancy*
- Principle: first send one full frame, then send only the difference between this frame and the next one (difference)
 - duration of one image defines time constant τ (MPEG)
 - can also choose duration of single pixel or line
 - risk of losing full frame due to corruption of image (noise)
- Problem: what if signal reception starts mid-way?
- Solution: send full frames periodically (e.g., every 10th frame)
 - avoids accumulation of reconstruction errors

Temporal Redundancy



Frame 1

-



Frame 2



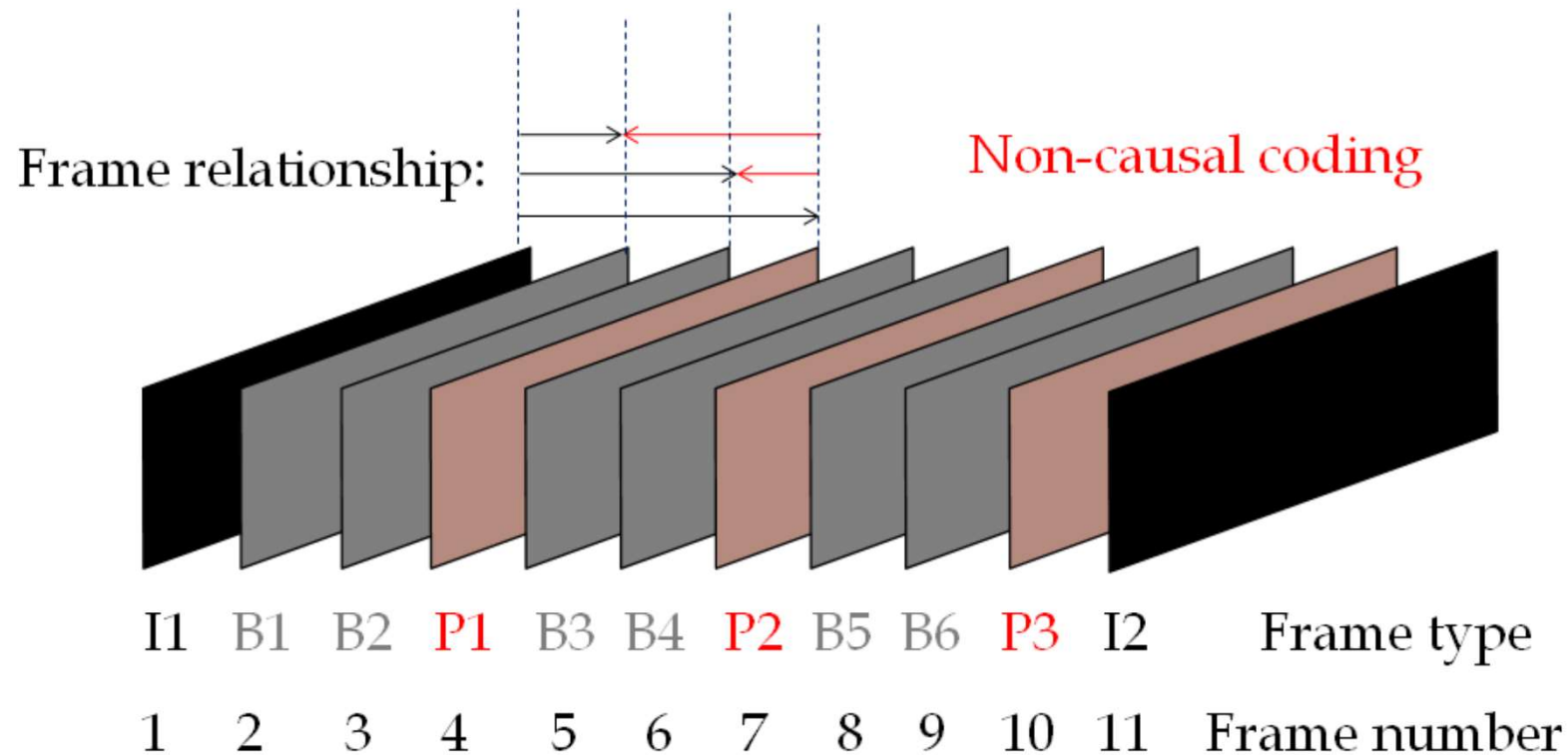
Frame 1 - Frame 2

Video Image coding

- Three types of frames:
 - Intra-coded (*I-frames*): frames (images) are coded independently of any other frame
 - Progressive (*P-frames*): predicted from previous frame
 - Bi-directional (*B-frames*): prediction from both the previous (backward) and next (forward) frames
 - extends forward prediction to backward prediction also, increasing predictive power and image quality
- Sequence of different types of frames forms a *group of pictures*
 - *I*-frames: least compression (because independent of *P* and *B*)
 - *B*-frames: highest compression (because reliant on *I* and *P*)

Group of Pictures: Viewing

Viewing (or generation) & coding sequence:

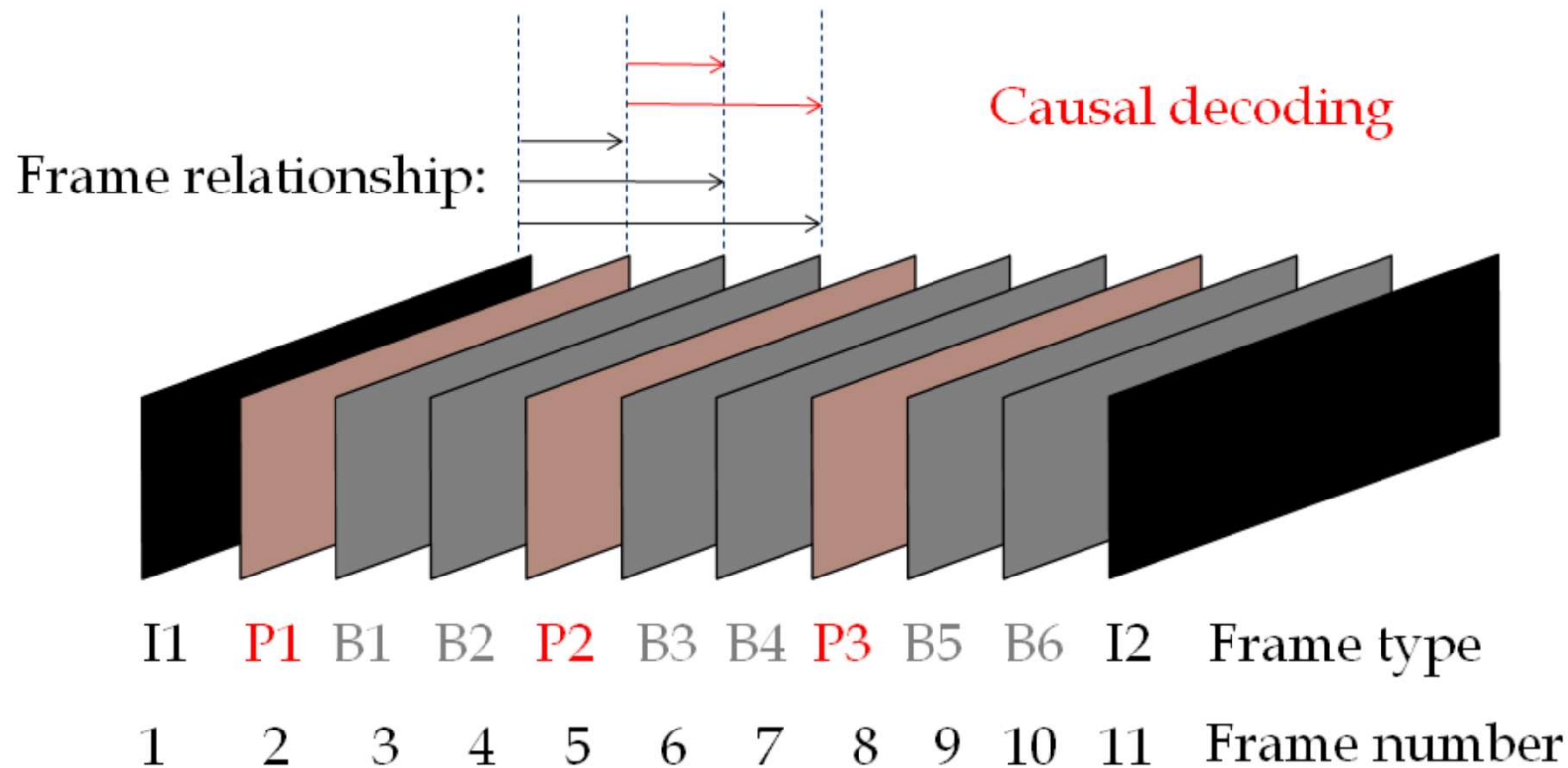


I, P, B Frame Re-Ordering

- *I*-frames do not exploit temporal redundancy.
- If there is an *I*-frame every 13 frames, then after a channel change the next *I*-frame is available after maximum 0.5 s (frame rate 25 Hz)
- *P*-frames are predicted from the previous *I*-frame
- *B*-frames are predicted from *I*- and/or *P*-frames immediately before and after the *B*-frame
 - Necessary for decoder to have the subsequent *P*- or *I*-frame before a *B*-frame can be decoded
 - Hence the sequence of the frames must be changed before transmission: the *I*- and *P*-frames that are used for a *B*-frame prediction are always transmitted before this *B*- frame: $I_1, P_1, B_1, P_2, B_2, \dots$

Group of Pictures: Transmission

Transmission & decoding sequence:



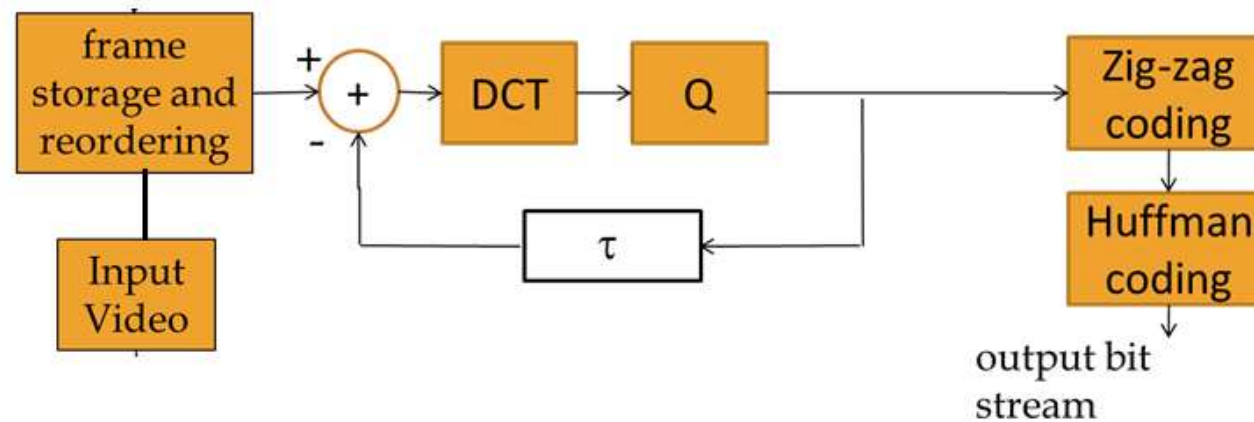
Storage Requirements

- Decoder needs subsequent *P*- or *I*- frame before a dependent *B*-frame can be decoded
- Sequence of frames is therefore re-arranged before transmission of the sequence
 - Viewing sequence \neq transmission sequence
- Encoder needs to store four frames for coding two consecutive *B*-frames (why?)
- Decoder needs to store only two frames: only the relevant *I*- and *P*- (or *P*- and *B*-) frames are needed
 - Simple and cost-effective real-time decoder for consumer

MPEG-2 Video Encoder

- Intra-frame coding operations

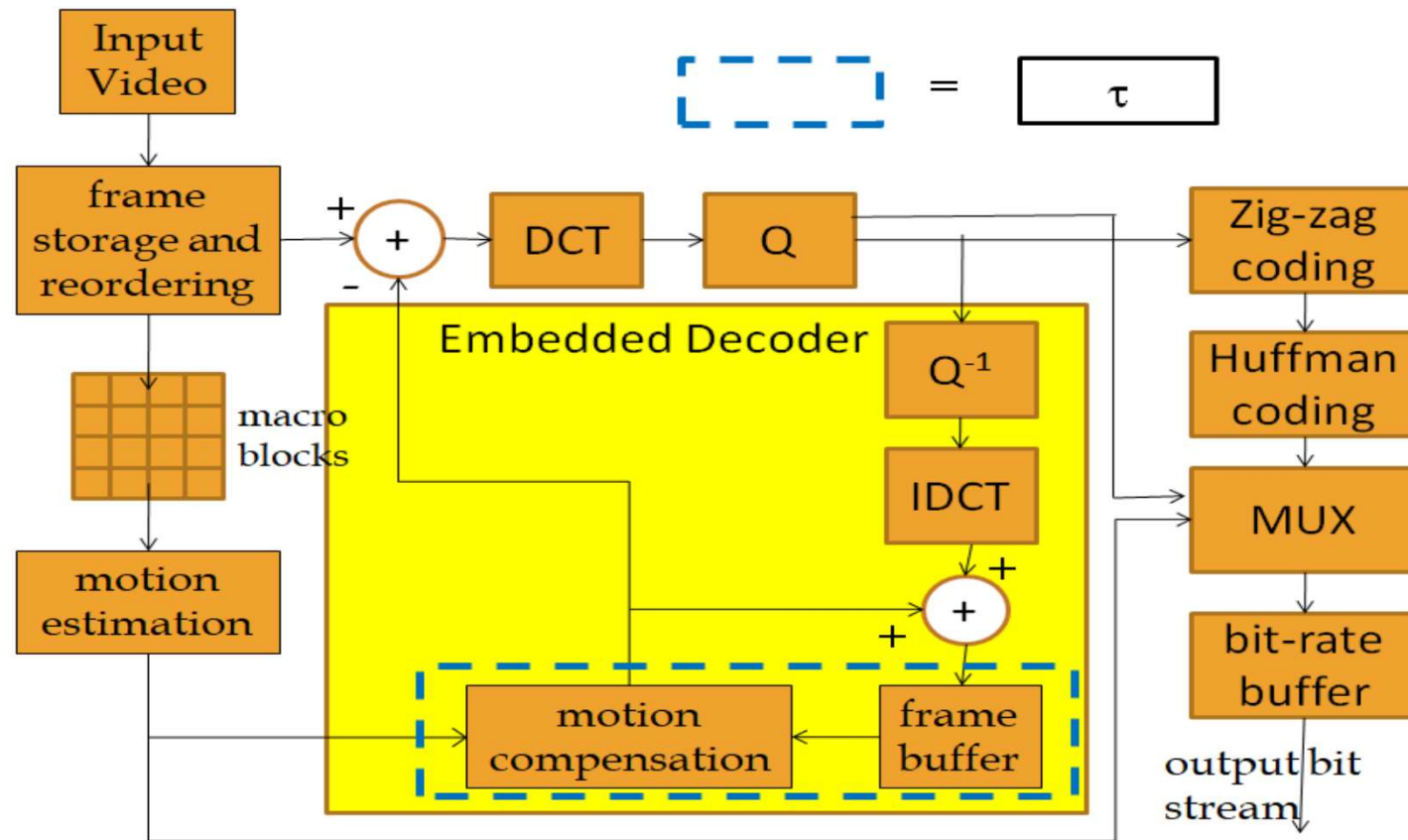
1. Find the difference picture (P- or B-frames)
2. DCT converts blocks to spatial frequency domain
3. Quantization (quite complex to determine correct tables)
4. Zig-zag coding (2D \rightarrow 1D)
5. Huffman (+ run-length) coding



MPEG-2 Video Encoder

- Inter-frame coding operations
 1. Frame re-ordering and storage (decoder needs subsequent *P*- or *I*-frames before decoding a *B*-frame)
 2. Current frame is split into macro-blocks.
 3. Motion estimation describes movement of macro-blocks
- Finally, the inter-frame and intra-frame information is combined for transmission

MPEG-2 Video Encoder



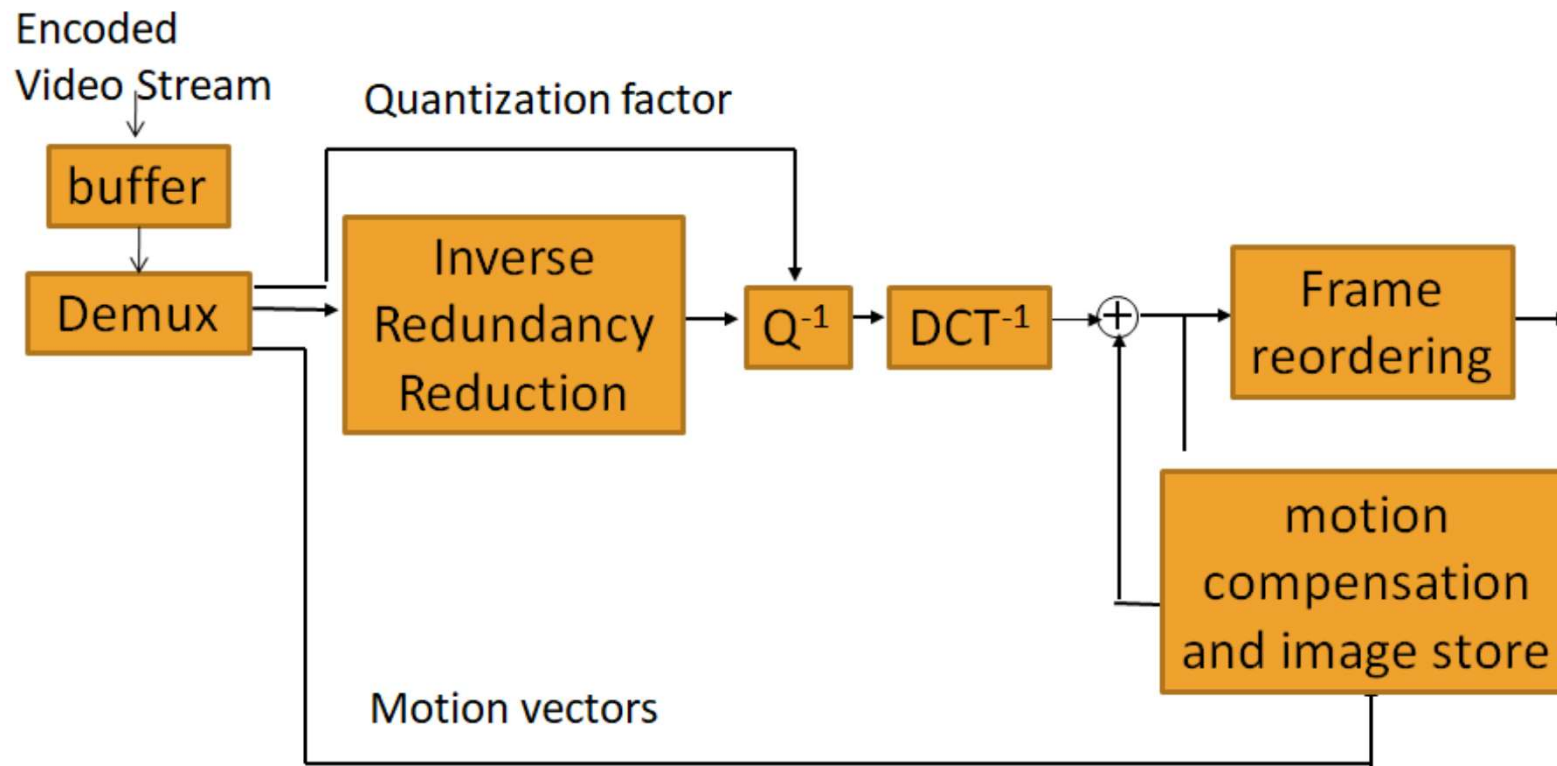
Embedded Decoder & Bit rate buffer

- Needs difference between current picture and previous (or next) picture.
- Must calculate difference between current picture and *quantized* previous picture, in order to keep encoder and decoder synchronized.
- Each image is different in its level of information content
 - Complex images need more bits to be encoded than simple images: *adaptive coding*
- Overall bit-rate in transmission should remain constant.
- If buffer nears overflow then the quantization is made coarser, so less data is sent to the buffer.

Summary: Principles of Encoder

- Video: same as for compression of still images
- **Image store** allows for comparing an image with a previous image (**differential coding**)
- Temporal redundancy reduction is achieved by identifying motion in the sequence and sending **motion vectors** in the bit stream to the decoder
- **Quantization steps** are **adapted** to obtain **constant bit-rate** for data at output of encoder (real-time!)
 - Output buffer absorbs differing data rates produced by the encoder
 - If buffer is close to overflow, quantization is made coarser so that less data is sent to the buffer (lower data rate)

MPEG Video Decoder



Input buffer has constant rate input but transfers data to demux at variable rate

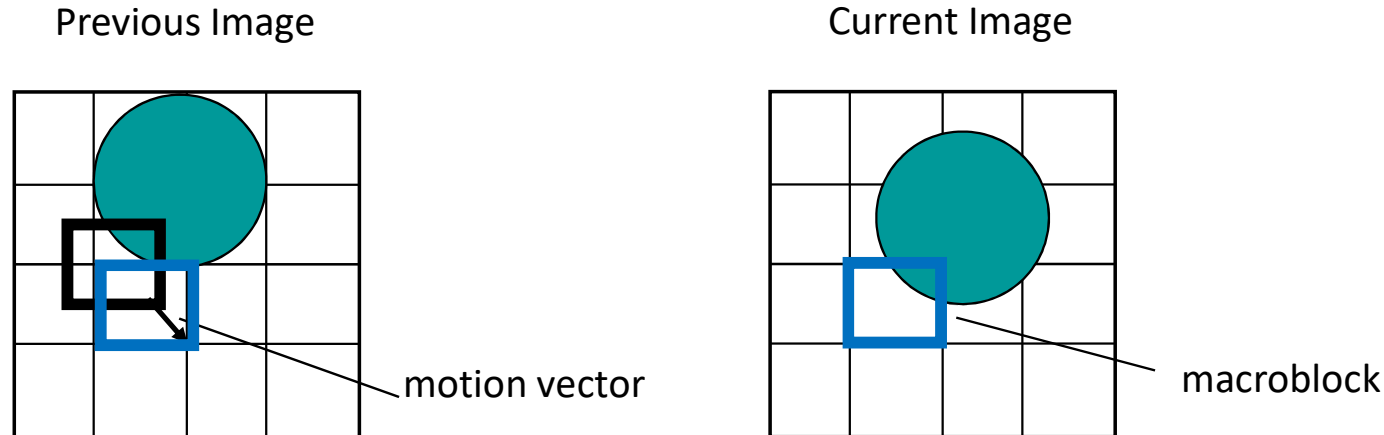
Decoder Components



- **Demux** separates coded image data from quantization factor and motion vectors
- **Inverse redundancy**: to original redundant format
- **Inverse quantization**
 - uses quantization factor for portion of data in data stream
- **Inverse DCT**: reverse transform of coefficients, back to spatial domain (Y , C_R , C_B)
- Predicted values added, making use of the **motion vectors**
- Frames are put back in **correct order**

NB: decoder is simpler than encoder.

- This is desirable: every receiver must have a low-cost decoder; only broadcaster can afford high-cost encoder

P-frame Motion Estimation



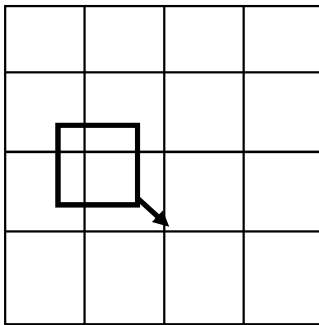
- Image is divided into 16×16 -pixel macroblocks ($= 2 \times 2$ DCT blocks)
- If we simply code the pixel-by-pixel difference between corresponding macroblocks , then a large number of difference pixels need to be encoded for moving artefacts.
- If the macroblock can be compared with a shifted macroblock  then the differences are small, but need extra encoding and transmission of motion vector
- For *P*-frames, prediction is unidirectional (forward)

Motion Estimation

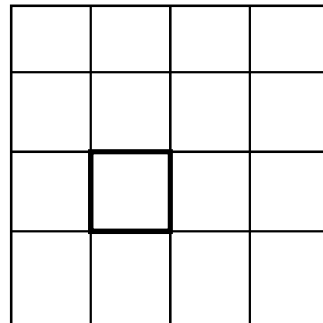
- Requires sending information about how the macroblock has moved. This is a motion vector.
- Calculation of motion vectors is computationally intensive
 - Most common method: shift the macroblock in all possible directions (within a defined search area); then choose the 'best' i.e., the vector for which pixel difference is smallest
 - Computational effort is proportional to size of search area, but larger search area yields better image quality for chosen bit rate

B-frame Motion Estimation

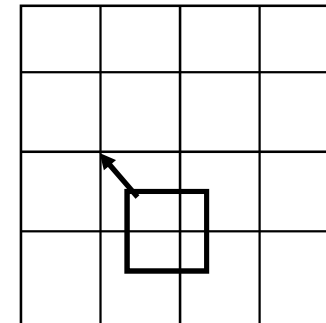
previous image A



current image $(A+C)/2$

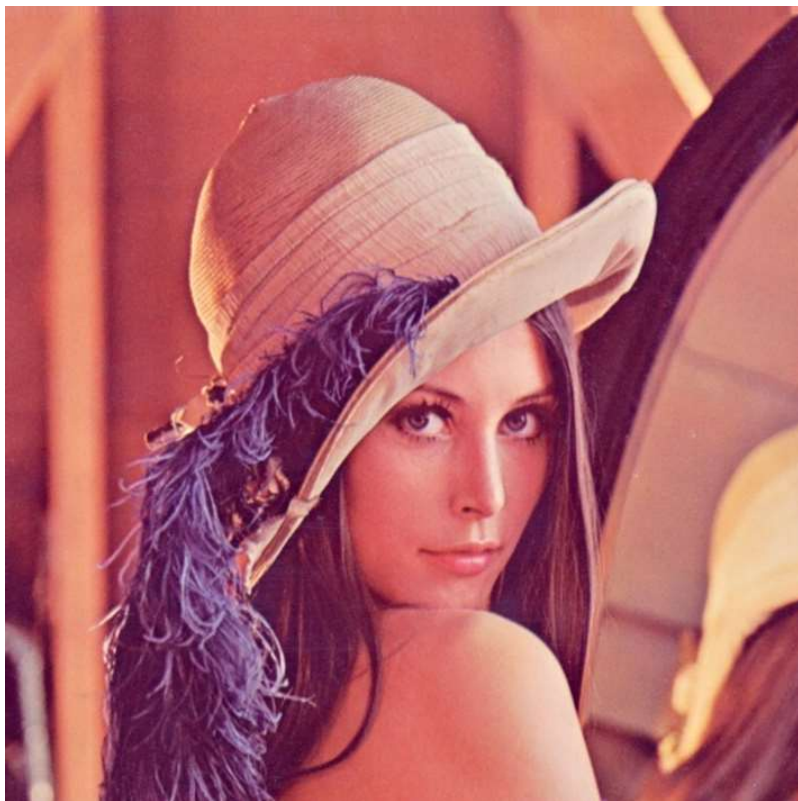


next image C



- Bidirectional prediction: also next (future) image is analyzed
- This enables motion vector to go backward in time
- Prediction of macroblock *can* be e.g. mean of the values between previous and next macroblocks
- Increases compression and computational cost

Quantization Effect



768k



67k (= 11.5 x)

Quantization Effect II

- JPEG compression technique is very good at concealing quantization effects up to very large compression ratios
 - Here, compression ratio 30 is still acceptable
 - At compression ratio 120, quantization effects become very visible



24k (= 30 x)

Quantization Effect III



768k



6k (= 120 x)

MPEG-2 Profile for DVB

	<i>Simple Profile (no B-frames)</i>	<i>Main Profile (4:2:0)</i>	<i>SNR scalable Profile</i>	<i>SNR+spatial scalable Profile</i>	<i>High Profile (4:2:2)</i>
Low Level		352x288, 4 Mbps	352x288, 4 (3) Mbps		
Main Level	72x576, 15 Mbps	720x576, 15 Mbps	72x576, 15 (10) Mbps		720x576 (352x288), 20 (15,4) Mbps
High-1440 Level		1440x1152, 60 Mbps		1440x1152 (720x576), 60 (40,15) Mbps	1440x1552 (720x576), 80 (60,20) Mbps
High Level		1920x1152, 80 Mbps			1920x1152 (960x576), 100 (80,25) Mbps

Conclusion

- Encoding a single frame removes redundancy due to different parts of the image being similar and frequency characteristics of the human eye ([intra-frame coding](#))
- Encoding a sequence of frames removes redundancy associated with similarity of consecutive images ([inter-frame coding](#))
- Combined inter- and intra-frame coding allow for compression 800 Mbps $\rightarrow \leq 8$ Mbps

DCT Matrix generation

- DCT equation is given by:

$$D(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right] \quad 1$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{if } u > 0 \end{cases} \quad 2$$

$$T_{ij} = \begin{cases} \frac{1}{\sqrt{N}} & \text{if } i = 0 \\ \sqrt{\frac{2}{N}} \cos\left[\frac{(2j+1)i\pi}{2N}\right] & \text{if } i > 0 \end{cases}$$