EBU5305

Interactive Media Design and Production

Digital Media Compression

Media compression : main points

- Compression can be lossless or lossy
- Lossless techniques include RLE and Huffman
- Lossy techniques include JPEG, MPEG (for video), Psychoacoustics (for audio)

Reading

- BurgChapter1.pdf p. 34-49
- BurgChapter2.pdf p. 7-19
- BurgChapter3.pdf p. 52-64

Media Compression Techniques

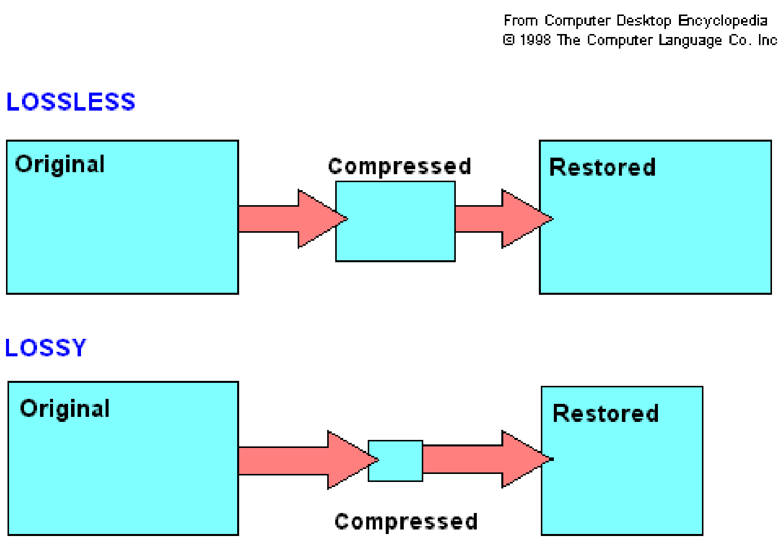
- For storage
  - Typical uncompressed colour image requires about 1 MB storage

- For transmission over Internet, local networks and other networks with restricted bandwidth

Basics of Media Compression

- **Lossless encoding**
  - redundant data is not encoded
  - no information is lost in compression and decompression cycle
  - possible because some data appear more often that other and some data normally appear together
- **Lossy encoding**
  - details not perceived by users are discarded; e.g. small change in lightness (luminance) noticed more than change of colour details
  - approximation of the real data

Lossless versus Lossy

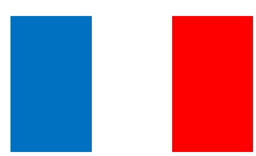


Media compression : main points

- Compression can be lossless or lossy
- Lossless techniques include RLE and Huffman
- Lossy techniques include JPEG, MPEG (for video), Psychoacoustics (for audio)

Run-length encoding (RLE)

- Physically reduce any type of repeating byte sequence (**run**), once the sequence of bytes reach a predefined number of occurrence (**length**)
  - E.g. area of solid colour



- Lossless

- Use **flag** that does not appear in the data stream (e.g. **!**)

ABCAAAABBBBCCCCCD **17 bytes**
ABC!4A!4B!5CD **13 bytes**

Question

- How would you encode the following sequence of symbols using RLE ?

ABCAAAAAABBBBBCCCCCCCCCCCD

Question

- How about this sequence ?

ABBCCCAADDDBBBAABCDAABB

RLE suitable image examples

Huffman encoding

- Encode frequent bit patterns with short code, infrequent bit patterns with longer codes
- Example of statistical coding
- Effective when probabilities vary widely
- Lossless
- Fixed-length inputs become variable-length outputs

symbol	probability	code
A	0.40	00
B	0.20	01
C	0.20	10
D	0.10	110
E	0.10	111

Huffman encoding : example

symbol	probability	code
A	0.40	00
B	0.20	01
C	0.20	10
D	0.10	110
E	0.10	111

ABBAAADCBAA

↳ **00 | 01 | 01 | 00 | 00 | 00 | 110 | 10 | 01 | 00 | 00**

Huffman encoding : example

symbol	probability	code
A	0.40	00
B	0.20	01
C	0.20	10
D	0.10	110
E	0.10	111

5 symbols → at least 3 bits / symbol

ABBAAADCBAA 3 bits x 11 = 33 bits

↳ **00010100000011010010000** 23 bits

Huffman encoding : example

symbol	probability	code
A	0.40	00
B	0.20	01
C	0.20	10
D	0.10	110
E	0.10	111

VARIABLE LENGTH ENCODING

ABBAAADCBAA 3 bits x 11 = 33 bits

→ **00010100000011010010000** 23 bits

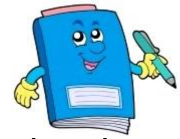
Question

Decode the following portion of message:

1111101111010011011110101101100110111110101011001100100111111

symbol	probability	code
A	0.40	00
B	0.20	01
C	0.20	10
D	0.10	110
E	0.10	111

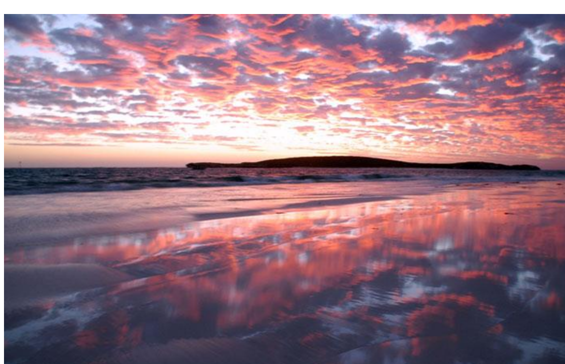
Question



Consider the following sequence of letters (each letter represents one byte): AABBBBCDDDAAAAA

- Encode the sequence using RLE (Run Length Encoding) and calculate the compression rate.
- Considering the following Huffman codes: 00, 01, 10, 110; how would you encode the sequence of bytes using Huffman encoding? What is the compression rate you achieve?

Huffman suitable image examples



Media compression : main points

- Compression can be lossless or lossy
- Lossless techniques include RLE and Huffman
- Lossy techniques include JPEG, MPEG (for video), Psychoacoustics (for audio)

JPEG: Joint Photographic Expert Group

- Standard for compressing and decompressing still images
- Based on DCT and applies to colour and grey-scaled still images
  - Independent of image sizes
  - Applicable to any aspect ratio (W/H)
  - Colour space and number of colours independent
  - Image content of any complexity, with any statistical properties
  - Run on many standard processors
  - Compression ratio dictated by user or application

JPEG is lossy!

Compression
1:100

Compression
1:10

Steps of JPEG compression

algorithm jpeg
/*Input: A bitmap image in RGB mode.
Output: The same image, compressed.*/

{
Divide image into 8 x 8 pixel blocks

(Convert image to a luminance/chrominance model such as YUV)

Use Discrete Cosine Transform (DCT) to transform the pixel data from the spatial domain to the frequency domain

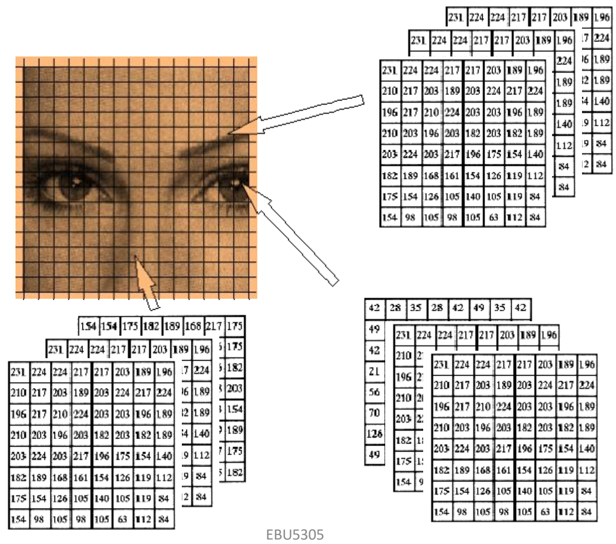
Quantize frequency values

Arrange the block in a zigzag order

Do run-length encoding

Do entropy encoding (e.g., Huffman)
}

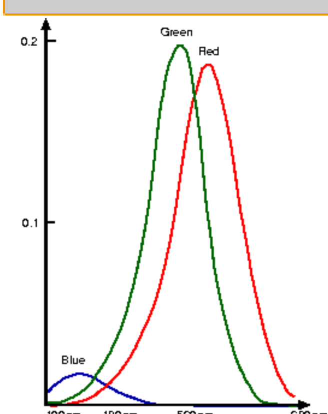
8x8 pixel blocks (colour image)

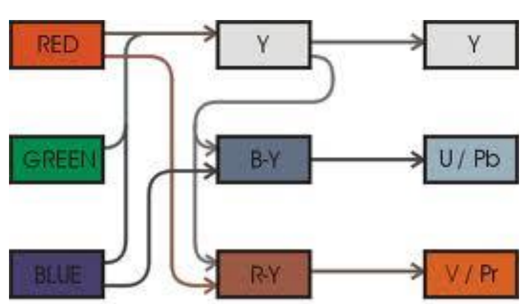


EBU5305

RGB to YUV (or YCrBr)

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

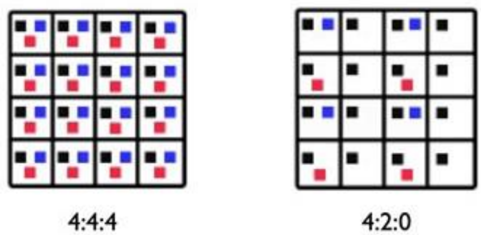




EBU5305

Chroma Sub-Sampling

E.g. In 4:2:0 chroma sub-sampling, the **Cb (U)** and **Cr (V)** channels are only sampled on each alternate line.



Steps of JPEG compression

algorithm jpeg
/*Input: A bitmap image in RGB mode.
Output: The same image, compressed.*/

{
Divide image into 8 x 8 pixel blocks

Convert image to a luminance/chrominance model such as YUV

Use Discrete Cosine Transform (DCT) to transform the pixel data from the spatial domain to the frequency domain

Quantize frequency values

Arrange the block in a zigzag order

Do run-length encoding

Do entropy encoding (e.g., Huffman)
}

Discrete Cosine Transform (DCT)

- Separate image into parts (spectral sub-bands) of differing importance (with respect to the image visual quality)
- Transforms signal or image from the spatial domain to the frequency domain

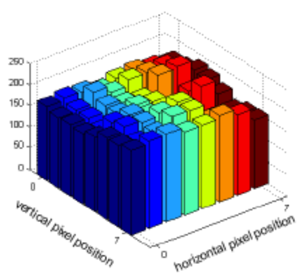
$$B(k_1,k_2) = \sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1} 4\cdot A(i,j)\cdot\cos\left[\frac{\pi\cdot k_1}{2\cdot N_1}\cdot(2\cdot i+1)\right]\cdot\cos\left[\frac{\pi\cdot k_2}{2\cdot N_2}\cdot(2\cdot j+1)\right]$$

Input image = $N_1 \times N_2$ pixels
A(i,j): intensity of pixel

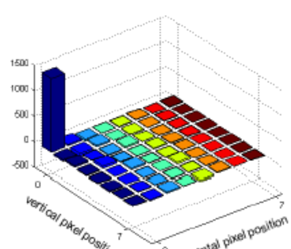
$B(k_1,k_2)$ =DCT coefficient in row k_1 and column k_2 of DCT matrix

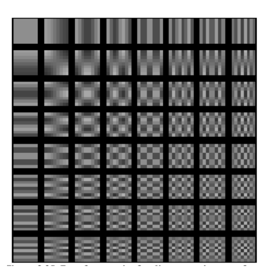
Discrete Cosine Transform (DCT)

Pixel matrix



DCT coefficient matrix





Steps of JPEG compression

algorithm jpeg
/*Input: A bitmap image in RGB mode.
Output: The same image, compressed.*/

{
Divide image into 8 x 8 pixel blocks

Convert image to a luminance/chrominance model such as YUV

Use Discrete Cosine Transform (DCT) to transform the pixel data from the spatial domain to the frequency domain

Quantize frequency values

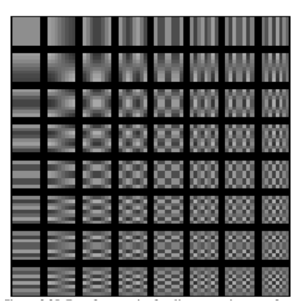
Arrange the block in a zigzag order

Do run-length encoding

Do entropy encoding (e.g., Huffman)
}

DCT coefficients matrix

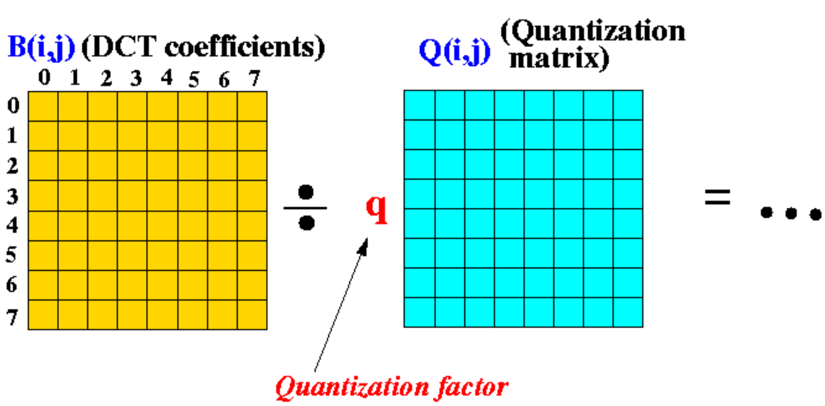
- Describes each 8×8 block in terms of how much the detail changes (colour/luminance information per pixel)
- Values arranged from lowest frequencies to highest frequencies
  - Lowest frequencies represent average value for the block
  - Highest frequency represent fine detail (⇒ can be dropped)
    - Eye unable to perceive brightness levels above or below thresholds
    - Gentle gradation of brightness of colour are more important to the eye than abrupt changes



Quantisation = Division

$$F^{Q}(u,v) = Integer\ Round\left(\frac{F(u,v)}{Q(u,v)}\right)$$

Quantisation = Division



Quantisation tables

a. Low compression

1	1	1	1	1	2	2	4
1	1	1	1	1	2	2	4
1	1	1	1	2	2	2	4
1	1	1	1	2	2	4	8
1	1	2	2	2	2	4	8
2	2	2	2	2	4	8	8
2	2	2	4	4	8	8	16
4	4	4	4	8	8	16	16

b. High compression

1	2	4	8	16	32	64	128
2	4	4	8	16	32	64	128
4	4	8	16	32	64	128	128
8	8	16	32	64	128	128	256
16	16	32	64	128	128	256	256
32	32	64	128	128	256	256	256
64	64	128	128	256	256	256	256
128	128	128	256	256	256	256	256

Steps of JPEG compression

algorithm jpeg
/*Input: A bitmap image in RGB mode.
Output: The same image, compressed.*/

{
Divide image into 8 x 8 pixel blocks

Convert image to a luminance/chrominance model such as YUV

Use Discrete Cosine Transform (DCT) to transform the pixel data from the spatial domain to the frequency domain

Quantize frequency values

Arrange the block in a zigzag order

Do run-length encoding

Do entropy encoding (e.g., Huffman)
}

Zigzag

150	80	20	4	1	0	0	0
92	75	18	3	1	0	0	0
26	19	13	2	1	0	0	0
3	2	2	1	0	0	0	0
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

150	80	92	26	75	20	4	18	19	3	1	2	13	3	1	0	1	2
2	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								

Steps of JPEG compression

algorithm jpeg
/*Input: A bitmap image in RGB mode.
Output: The same image, compressed.*/

{
Divide image into 8 x 8 pixel blocks

Convert image to a luminance/chrominance model such as YUV

Use Discrete Cosine Transform (DCT) to transform the pixel data from the spatial domain to the frequency domain

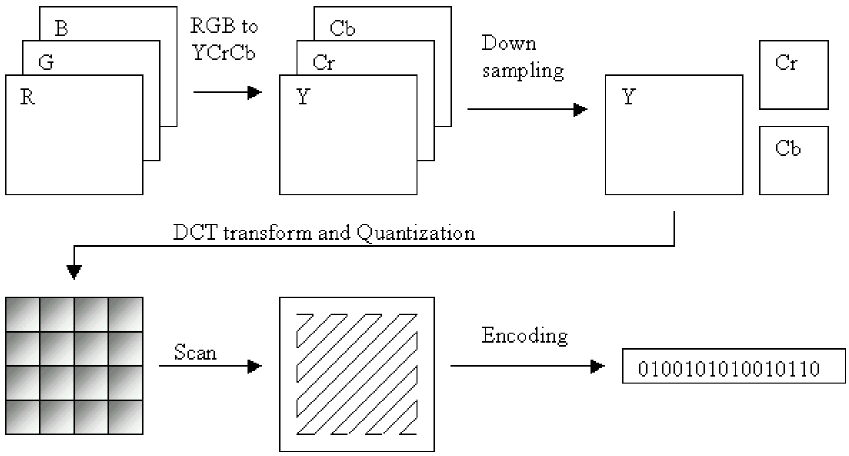
Quantize frequency values

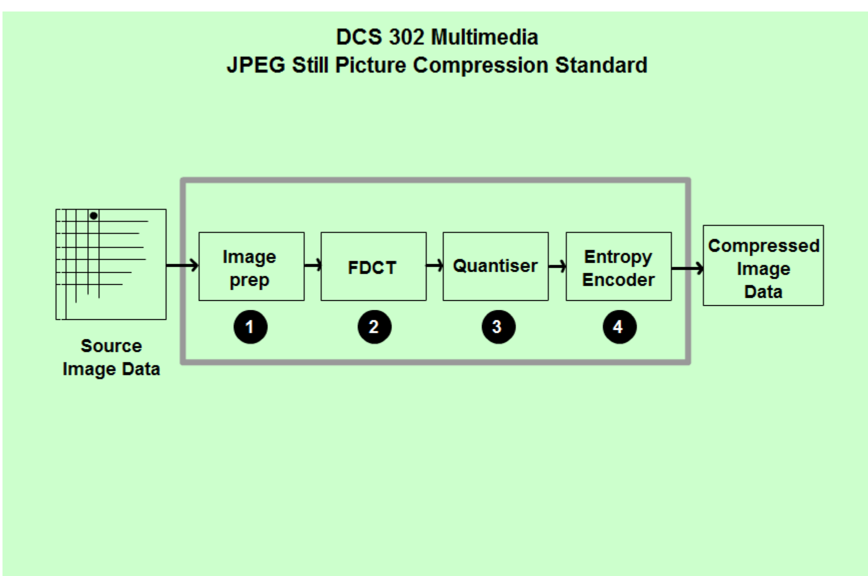
Arrange the block in a zigzag order

Do run-length encoding

Do entropy encoding (e.g., Huffman)
}

JPEG overview





Some comparisons

Original size of color image (Kb)				313.076
Original size of B/W image (Kb)				104.437
Quality Factor	Color JPEG		B/W JPEG	
	File Size (Kb)	Comp. Ratio	File Size (Kb)	Comp. Ratio
75	23.039	13.59	21.02	4.97
20	8.457	37.02	7.599	13.74
5	4.009	78.09	3.257	32.07
3	3.268	95.80	2.522	41.41

Question

Consider the 4x4 block of pixels in Table 1, where: r corresponds to the fully saturated bright red colour; g to the fully saturated bright green colour; b to the fully saturated bright blue colour; w to white; bk to black; and gy to mid-grey.

r	b	g	b
gy	r	gy	gy
gy	w	r	w
gy	bk	bk	r

Table 1

i) Extract from the block of pixels shown in Table 1, the 4x4 matrix of values, comprised between 0 and 255, that correspond to the R channel.

Question

ii) Consider that the matrix shown in Table 2 is the matrix of DCT coefficients obtained by transforming the matrix of values of question i). Apply quantisation to this matrix, using the quantisation matrix shown in Table 3, and calculate a new matrix of quantised values.

1000	800	100	40
600	400	30	10
150	80	10	3
20	10	5	2

Table 2

10	20	50	99
20	50	99	99
50	80	99	99
99	99	99	99

Table 3

Question

iii) Arrange the quantised values in zig zag order and do Run Length encoding.

iv) To obtain a better quality image after compression, what would you change to the quantisation matrix of Table 3?

Media compression : main points

- Compression can be lossless or lossy
- Lossless techniques include RLE and Huffman
- Lossy techniques include JPEG, MPEG (for video), Psychoacoustics (for audio)

Video compression

- Video as a sequence of pictures (or frames)
- JPEG algorithm used for intra-frame coding
  - e.g. moving JPEG (MJPEG) exploits only intra-frame coding (spatial redundancy)
- High correlation between successive frames (temporal redundancy)
  - motion estimation and motion compensation: inter-frame coding
  - use a combination of actual frame contents and predicted frame contents

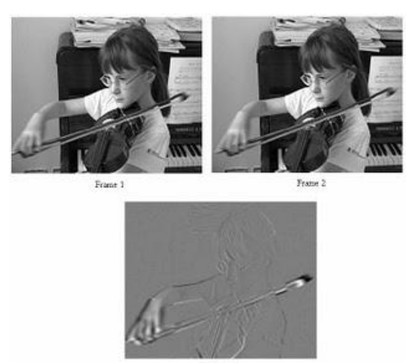
Inter-frame and intra-frame coding

- **Inter-frame** compression uses one or more earlier or later frames in a sequence to compress the current frame.
- **Intra-frame** compression uses only the current frame, which is effectively image compression.
- MPEG 1 & 2 use both inter-frame and intra-frame coding

45

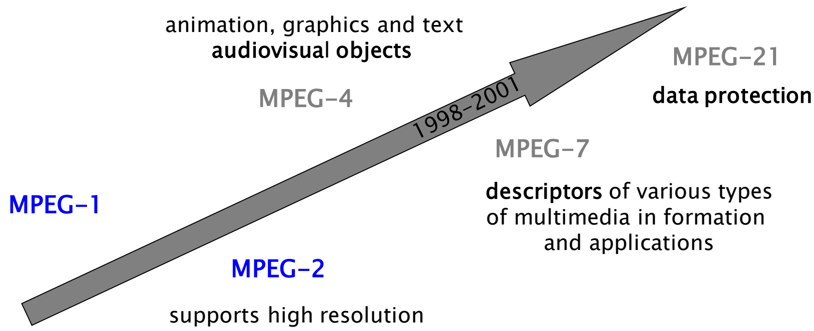
Temporal redundancy

- Temporal redundancy occurs when, in a sequence of images, consecutive images contain similarities.
  - For example, in a sequence of a tennis match, the court does not change; only the players and the ball make significant movements.
- With temporal compression, only the changes from one frame to the next are encoded.



46

MPEG family of standards: MPEG-1, MPEG-2, MPEG-4, MPEG-7 and MPEG-21

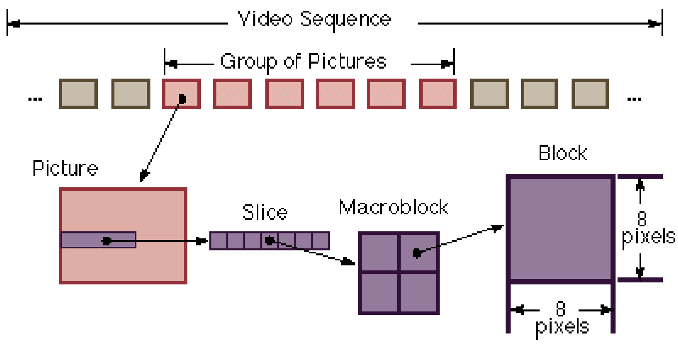


MPEG-1 and MPEG-2

- Applications
  - Digital storage media (CD-ROM = 1.5 MB/s, DVD = 1.1MB/s and hard disc ≥ 3 MB/s)
  - Mostly for asymmetric applications (frequent decompression, compression only once; e.g. electronic publishing, game, entertainment)
  - MPEG-1 & 2 are lossy compression schemes.

Symmetric applications (equal use of compression and decompression; e.g. electronic publishing production, video mail, video conference)

Structure of Digital Videos



Two main types of compression

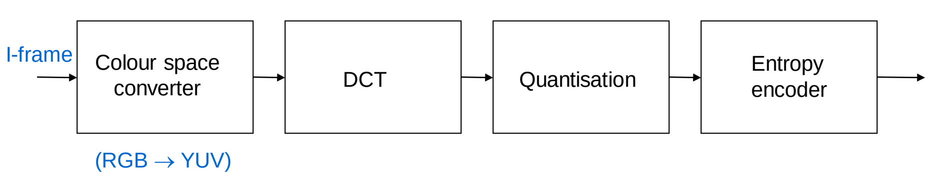
- Spatial compression (intra-frame coding)
  - Allow fast random access
  - Applied to key frames (e.g. every 12^{th} frame, start of new scene, etc)
    - **I-frames**
- Temporal compression (inter-frame coding)
  - Allow high compression
  - Exploit temporal redundancy between frames
  - Applied to frames occurring between key frames
  - Based on motion detection
    - **P- and B- frames**

MPEG video encoding

- input frames are preprocessed
  - color space conversion
  - spatial resolution adjustment
- frame types (Intra or Inter) are decided for each frame/picture
- each picture is divided into macroblocks of 16 X 16 pixels
- macroblocks
  - are intracoded for I frames
  - are predictive coded or intracoded for P and B frames
  - are divided into six blocks of 8 X 8 pixels
    - 4 luminance and 2 chrominance
    - DCT is applied to each block → transform coefficients
      - » quantized
      - » zig-zag scanned
      - » variable-length coded

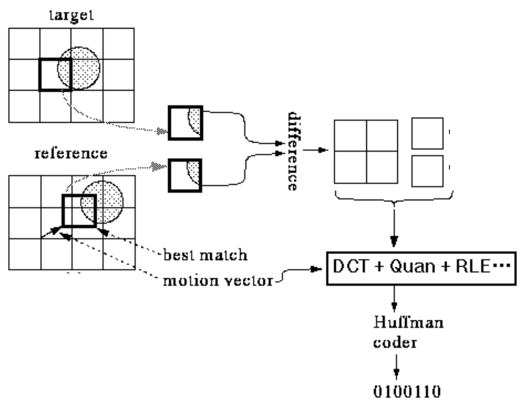
I-frames (Intra-coded frames)

- Self-contained (without reference to other frame)
- Treated as a still image and make use of JPEG
- Compression rate is the lowest
- Points for random access in video stream



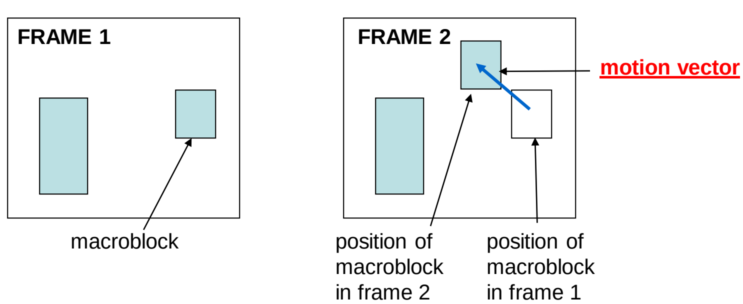
P-frames (Predictive-coded frames)

- Use a reference frame for motion estimation
- Hold only the changes in the image from a previous frame
- Compression rate is higher

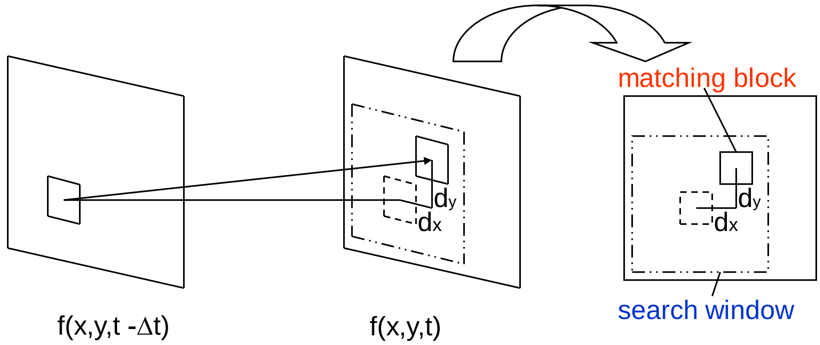


Motion estimation

- Based on looking for matching macroblocks of data in successive frames, and measuring the difference between matching macroblocks in terms of:
  - Position within the frame (expressed as motion vectors)
  - Luminance and colour difference (error term)

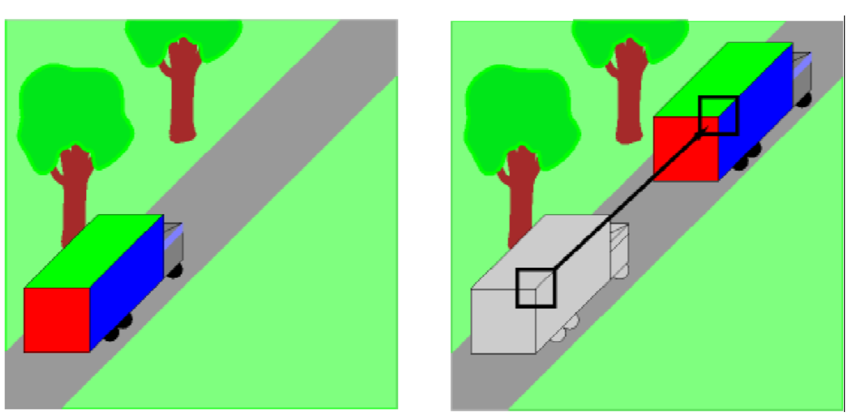


Motion estimation



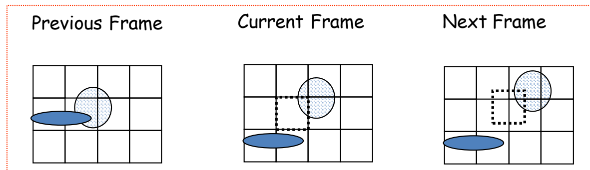
$$\min \sum_{B} \left\| f(x,y,t) - f(x-d_x, y-d_y, t-\Delta t) \right\|$$

Motion estimation of macroblocks



B-frames (Bi-directionally predictive-encoded frames)

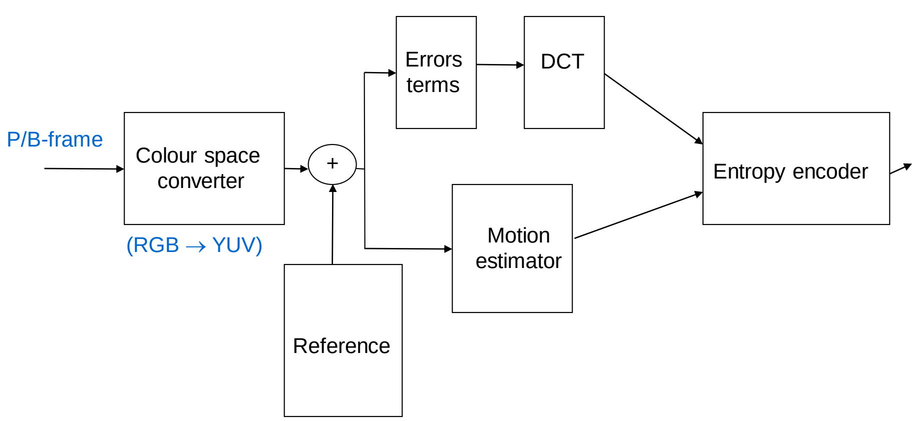
- Prediction limit
  - Some macroblocks may need information that is not present in the *previous reference frame*



  - Such information might be available in a *subsequent frame* …

- MPEG uses a third frame type (B-frame)
  - to form a B-frame: search for matching MBs in both past and future frames
  - typical pattern is IBBPBBPBB IBBPBBPBB IBBPBBPBB
  - actual pattern is up to encoder, and need not be regular

P- and B- frames



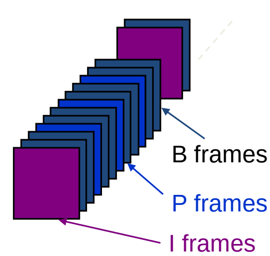
58

P- and B- frames

- P-frame contains 1/3 data of I-frame
  - motion vectors
  - error macroblock
- B-frame contains 1/2 to 1/5 data of P-frame
  - least data but most computational
  - delay issue:
    - need to transfer future I or P frames before any dependent B-frames can be processed

MPEG 1&2 summary

- Images coded INTRA (I):
  - Random access
  - Error resiliance
- Images coded INTER (P):
  - Prediction from previous decoded image (I, P)
- Images coded BI-INTER (B):
  - Prediction from previous and / or future decoded image (I, P)
  - allow effective prediction of uncovered background (areas of the current picture that were not visible in the past and visible in the future)



Question

Suppose an MPEG encoder uses the nine-frame sequence IBBPBBPBB.

- Draw a diagram showing the dependencies between the first 10 frames of a compressed clip produced by this encoder.

Media compression : main points

- Compression can be lossless or lossy
- Lossless techniques include RLE and Huffman
- Lossy techniques include JPEG, MPEG (for video), Psychoacoustics (for audio)

Sound Compression

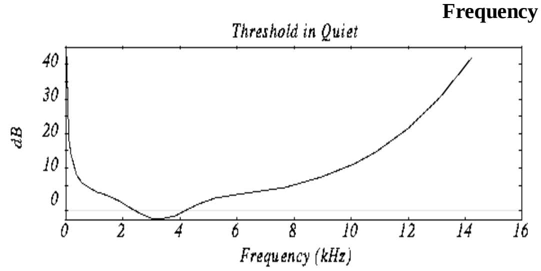
- Uncompressed 3 minutes song in stereo = 25 MB
- Lossless compression does not work well (complex and unpredictable nature of sound waveforms)
- Obvious compression technique: **silence compression**
  - Detect silence = samples falling below a threshold
  - Treat them as zero and compress using RLE (Run Length Encoding)
  - Silence is rarely absolute ⇒ not strictly lossless

63

Psychoacoustics

- Lossy compression that uses a mathematical description of the ear and brain perceived sounds
- Threshold of hearing = minimal level at which sound can be heard

Loudness = sound pressure level decibels (db)
Frequency = pitch (Hz)

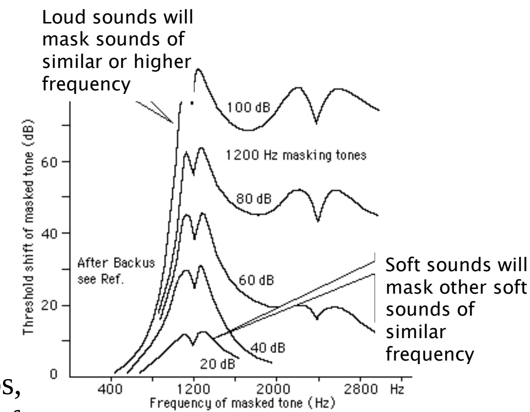


64

Psychoacoustics

- **Frequency masking**:
  - A loud tone may mask a softer tone of similar or higher frequency

- **Temporal masking**:
  - After a loud sound stops, there is a small delay before we can hear a softer tone



Media compression : main points

- Compression can be lossless or lossy
- Lossless techniques include RLE and Huffman
- Lossy techniques include JPEG, MPEG (for video), Psychoacoustics (for audio)