Multimedia Compression

- Image Compression
  - JPEG
- Video Compression
  - Spatial Compression
  - Temporal compression
  - MPEG
- Entropy Coding

JPEG

- Image standard developed by the Joint Photographic Experts Group.
- It is a lossy compression - some information is permanently lost.
- JPEG compresses color images as three separate greyscale images.
- The core of JPEG is to work out what data to throw away.

JPEG

Original image
24-bit RGB bitmap image
with 73,242 pixels
 219kB in size

Image Compression: JPEG

JPEG quality level

Q = 50 15kB

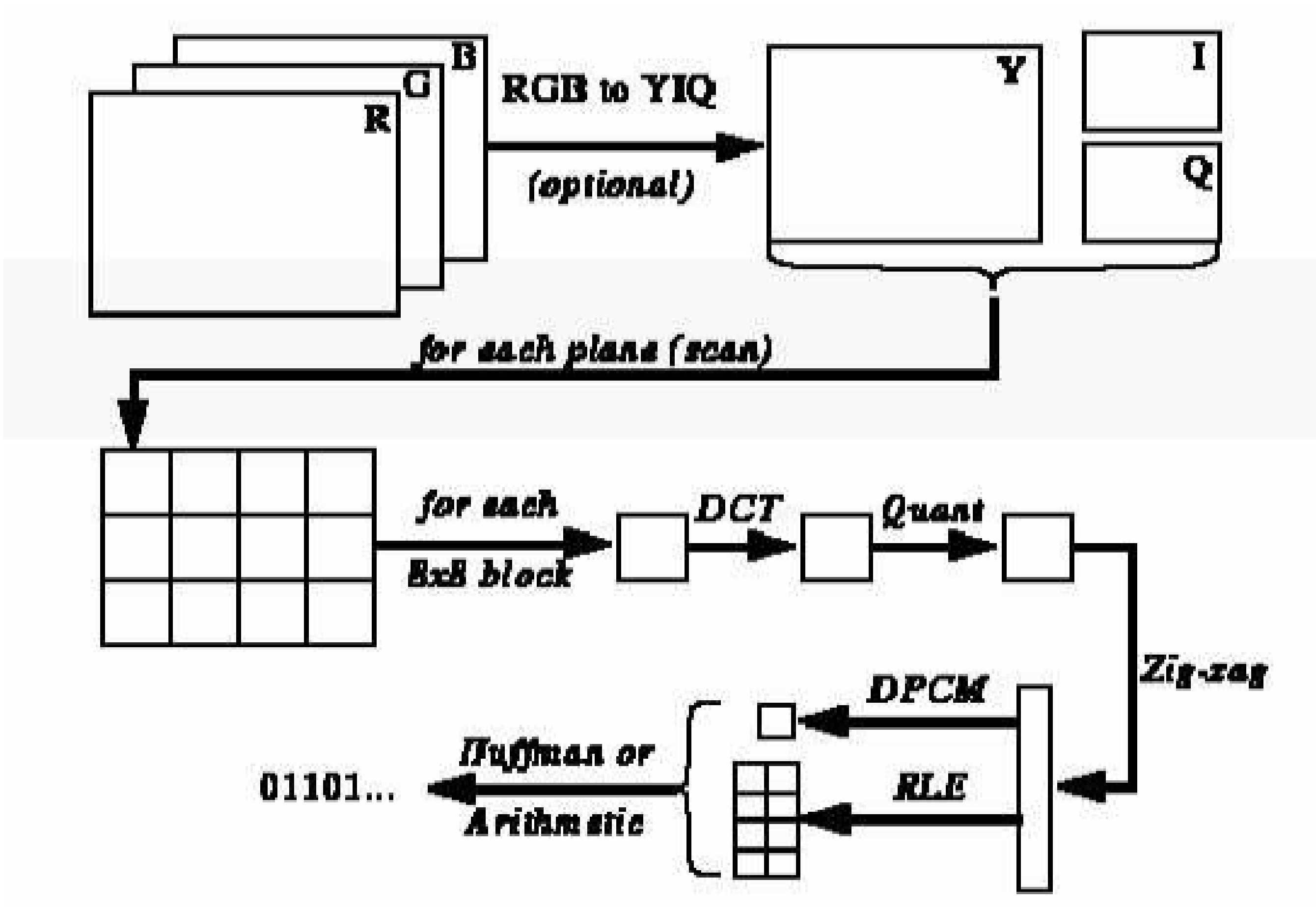
Q = 25 9kB

Q = 10 5kB

Q = 1 2kB

File size in bytes

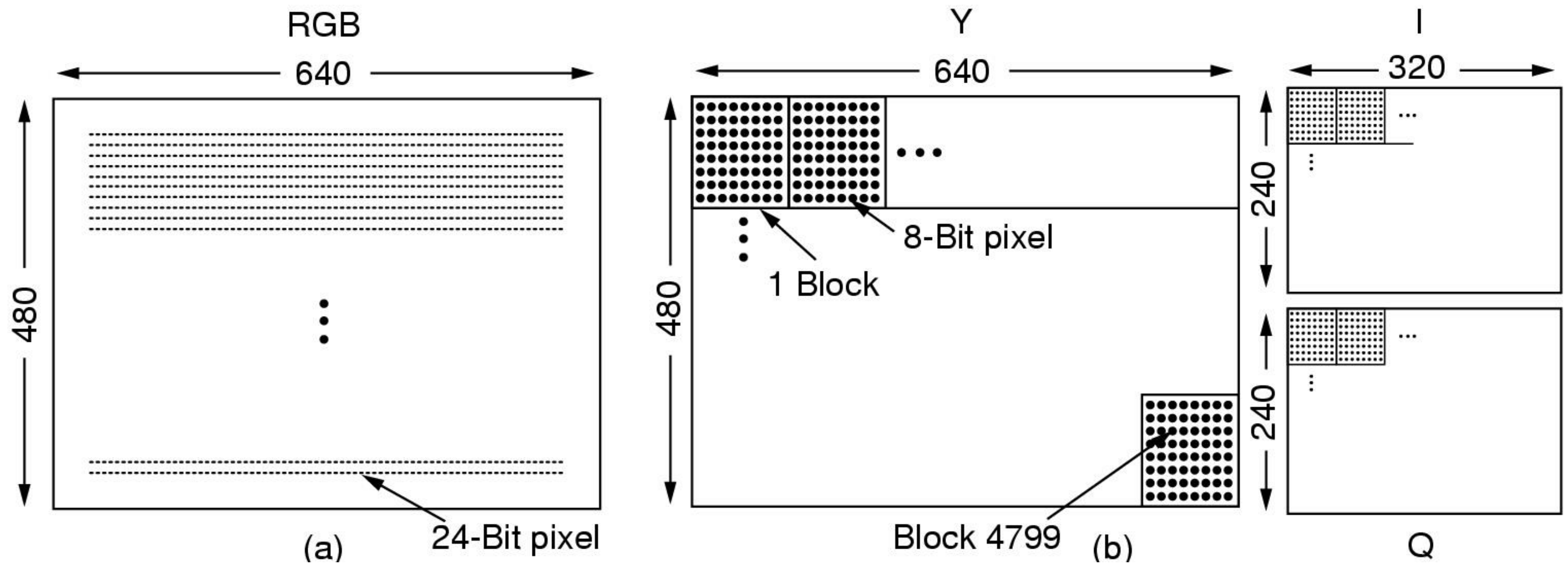
JPEG - Compression Steps



JPEG - Compression Steps

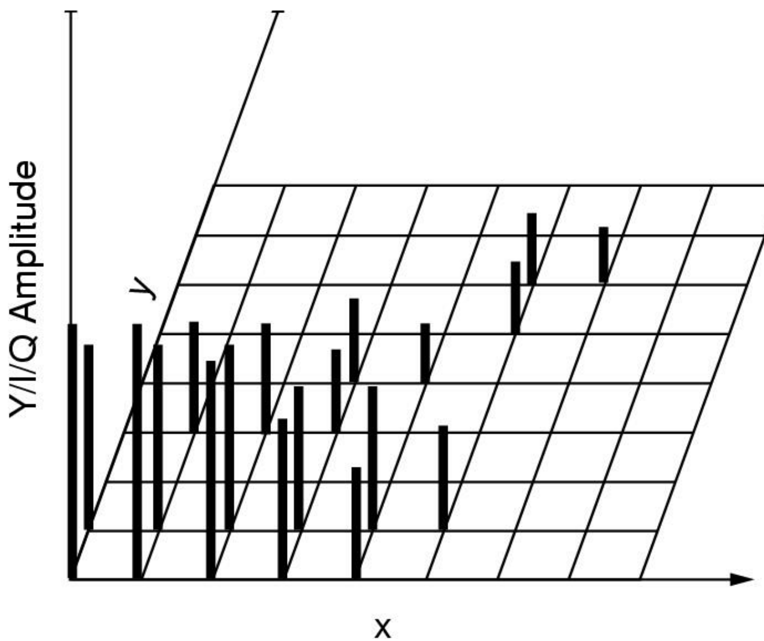
- Step 1: Block preparation
- Step 2: DCT (Discrete Cosine Transform) transformation
- Step 3: Quantization
- Step 4: Zig-zag scanning of coefficients
- Step 5: RLE compression and Huffman coding

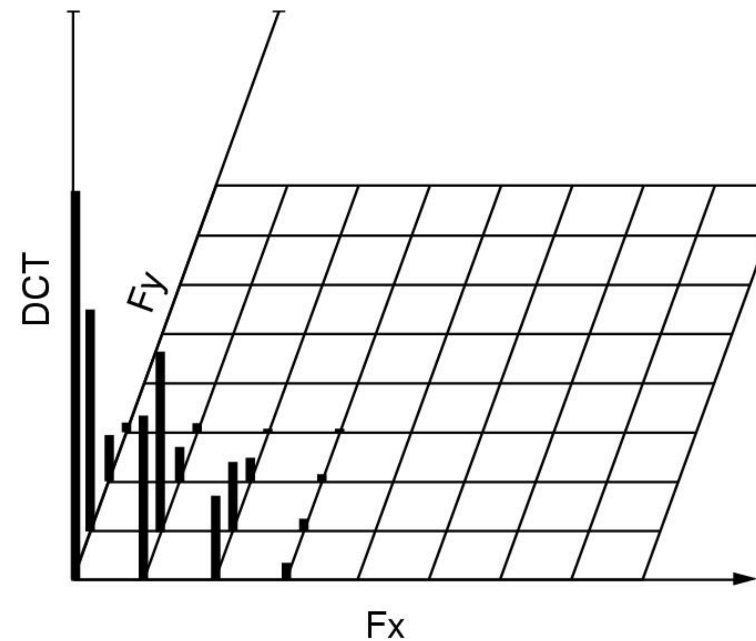
Step 1: Block Preparation



Step 2: DCT Transformation

For each block, do DCT transformation





$$F(u,v) = \frac{1}{4} C(u) C(v) \left[\sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y) * \cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16} \right]$$

Step 3: Quantization

DCT Coefficients

150	80	40	14	4	2	1	0
92	75	36	10	6	1	0	0
52	38	26	8	7	4	0	0
12	8	6	4	2	1	0	0
4	3	2	0	0	0	0	0
2	2	1	1	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Quantized coefficients

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

Quantization table

1	1	2	4	8	16	32	64
1	1	2	4	8	16	32	64
2	2	2	4	8	16	32	64
4	4	4	4	8	16	32	64
8	8	8	8	8	16	32	64
16	16	16	16	16	16	32	64
32	32	32	32	32	32	32	64
64	64	64	64	64	64	64	64

- Computation of the quantized DCT coefficients
- LOSSY COMPRESSION occurs in the step!
- (0,0) Coefficient is DC Coefficient
- Other (i,j) coordinates are AC Coefficients

Step 4: Zig-Zag Organization (from matrix to vector)

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

Step 5: RLE and Huffman Coding

- Take the Zig-zag vector, apply RLE (run length encoding) and then apply Huffman coding on the values in the vector

Video Compression

- Input to any video compression algorithm is a sequence of bitmapped images (The digitalized video)
- *Spatial (intra-frame) compression and temporal (inter-frame) compression are used together in most contemporary video codecs.*
  - ***Spatial compression***: each individual image can be compressed in isolation
  - ***Temporal compression***: sub-sequences of frames can be compressed by only storing the differences between them.

Compression and Decompression

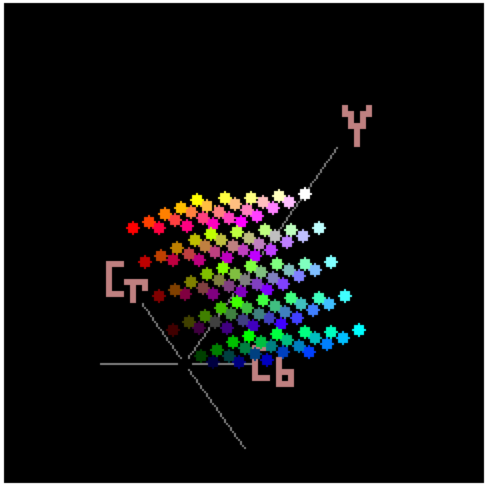
- Compression and decompression need not take the same time
- If they do, the codec is said to be *symmetrical*, otherwise it is *asymmetrical*.
- Codecs which take much longer time to decompress video than to compress it are essentially useless.
  - since playback must take place at a reasonably fast frame rate.

Spatial Compression

- Spatial compression is image compression applied to a sequence of bitmapped images.
- It could be lossy or lossless.
- Lossy compression
  - Can deteriorate the image quality.
  - Can provide high compression ratios to reduce video data to manageable proportions.

Spatial Compression

- Spatial compression of individual video frames is usually based on a Discrete Cosine Transformation, like JPEG.
- JPEG compression is applied to the three components of a colour image separately.
- Video data is usually stored using $Y'C_BC_R$ colour, with chrominance sub-sampling.



Applications

- **Motion JPEG or MJPEG**
  - The technique of compressing video sequences by applying JPEG compression to each frame
  - MJPEG was formerly the most common way of compressing video.
- **DV compression**
  - It is purely spatial.
  - It extends the JPEG technique by using a choice of sizes for transform blocks.

Temporal Compression

- Certain frames in a sequence are designated as key frames.
- Key frame occurs at regular intervals.
  - These key frames are either left uncompressed or only spatially compressed.
- Each frames between the key frames is replaced by a difference frame.
  - which records only the differences between the frame which was originally in that position and either the most recent key frame or the preceding frame.
  - The differences will only affect a small part of the frame.

Temporal Compression

Subtract the corresponding pixel values in each frame.

First Frame

Second Frame

Frame difference

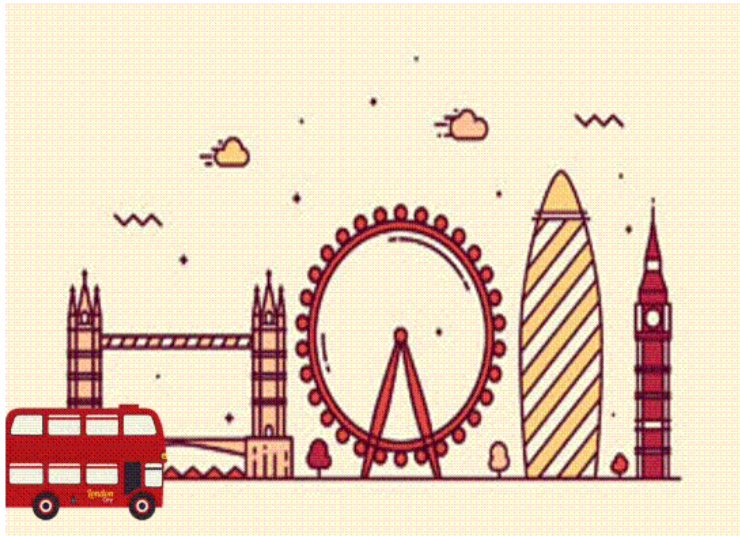
Temporal Compression

- MPEG
  - first expressed systematically broad principles to achieve either much higher compression ratios, or better quality at the same ratio, relative to DV or MJPEG.
  - It combines temporal compression based on *motion compensation* with spatial compression.
  - Temporal compression works by computing the difference between frames instead of storing every one in full.
  - In MPEG terminology, *I-pictures (Intra)* are only spatially compressed. *P-pictures (predictive)* are computed from a preceding I- or P-picture.

Motion Compensation

- **Motion compensation** is the technique of incorporating a record of the relative displacement of objects in the difference frames, as a motion vector.
- In existing codecs, motion compensation is applied to macroblocks (16*16 pixels), since coherent objects cannot usually be identified.
- Although basing difference frames on preceding frames probably seems the obvious thing to do, it can be more effective to base them on following frames.

Motion Compensation



Motion Compensation

- Black areas: the difference of two consecutive frames

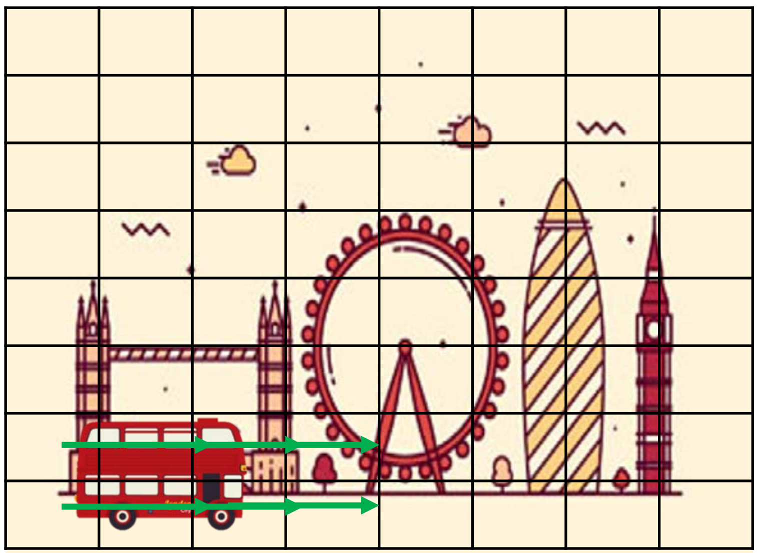
Motion Compensation

- The pixels values for the bus in the second frame are all there in the first frame.
- Record displacement of the bus by motion vector together with the changed pixels in the smaller area.

Motion Vector

Motion Compensation

- Motion compensation based on Macroblocks



Motion vectors for Macroblocks

After applying motion compensation on Frame 1

MPEG: Motion Picture Experts Group

- MPEG-1 was designed for video recorder-quality output (320x240 for NTSC) using the bit rate of 1.2 Mbps.
- MPEG-2 is for broadcast quality video into 4-6Mbps
- MPEG takes advantage of *temporal and spatial redundancy*. Temporal redundancy means that two neighboring frames are similar, almost identical.
- MPEG-2 output consists of three different kinds of frames that have to be processed:
  - I (Intracoded) frames - self-contained JPEG-encoded still pictures
  - P (Predictive) frames - Block-by-block difference with the last frame
  - B (Bidirectional) frames - Differences with the last and next frames

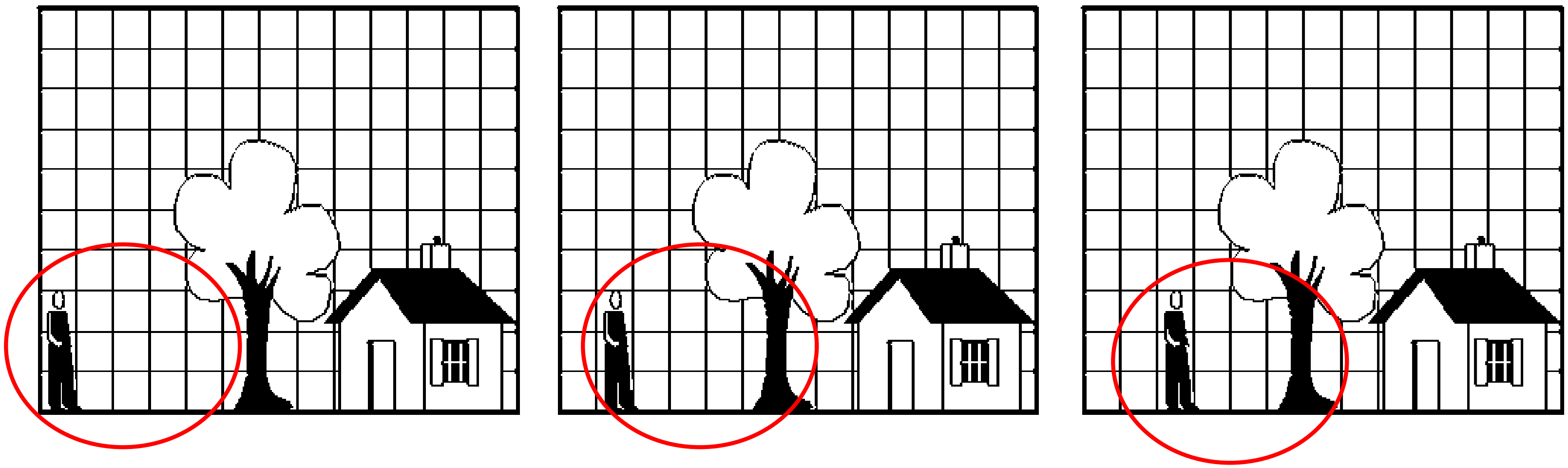
The MPEG Standard

- Suitable for stored video because it is an ***asymmetric lossy*** compression.
- Encoding takes long time, but decoding is very fast.
- MPEG display sequence: IPPBBBPPIPPBBBPPI
- MPEG transmitted dependency sequence: IPPPBBBPIPPPBBBPI
- Frames are delivered at the receiver in the dependency order rather than display order, hence we need buffering to reorder the frames.

MPEG/Video I-Frames

- I frames (intra-coded images)
- MPEG uses JPEG compression algorithm for I-frame encoding
- I-frames
  - use 8x8 blocks defined within a macro-block.
  - use DCT (Discrete Cosine Transform) on blocks.
  - do Quantization by a constant value for all DCT coefficients
    - no quantization tables exist

The MPEG Standard



Consecutive Video Frames

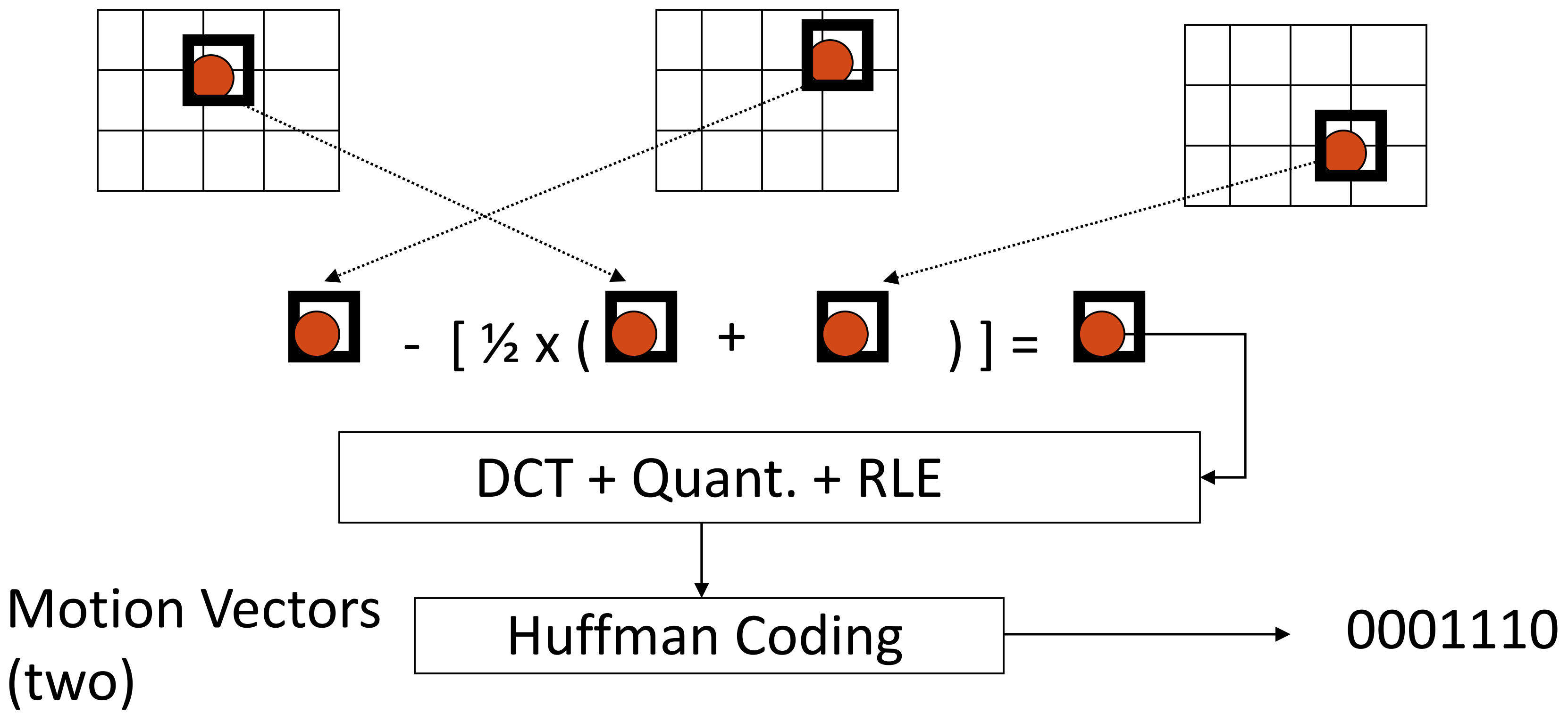
Usefulness of P frames (motion-based compression)

MPEG/Video P-Frames

- P-frames (predictive coded frames) requires previous I-frame and/or previous P-frame for encoding and decoding
- Use motion estimation method at the encoder
  - Define match window within a given search window.
  - Matching methods:
    - SSD correlation uses SSD = $\Sigma_i(x_i\text{-}y_i)^2$
    - SDA correlation uses SAD = $\Sigma_i|x_i\text{-}y_i|$

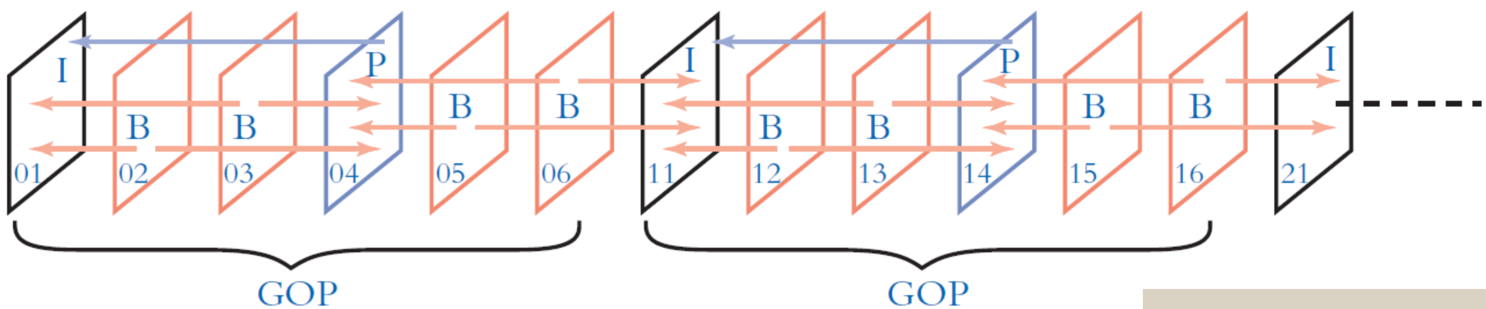
MPEG/Video B-Frame

- B-frames (bi-directionally predictive-coded frames) require information of the previous and following I and/or P-frame



MPEG Video Sequence

- A video sequence is encoded as a Group of Pictures (GOP). If B-pictures are used, a GOP may have to be reordered into display order for decoding.
- B-pictures use following pictures as well as preceding ones as the basis of frame differences and motion compensation.

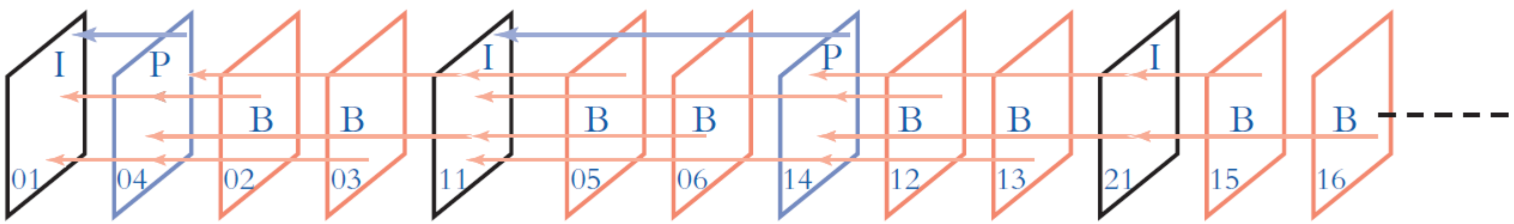


An MPEG sequence in display order

The arrows indicate the forward and bi-directional prediction.

MPEG Video Sequence

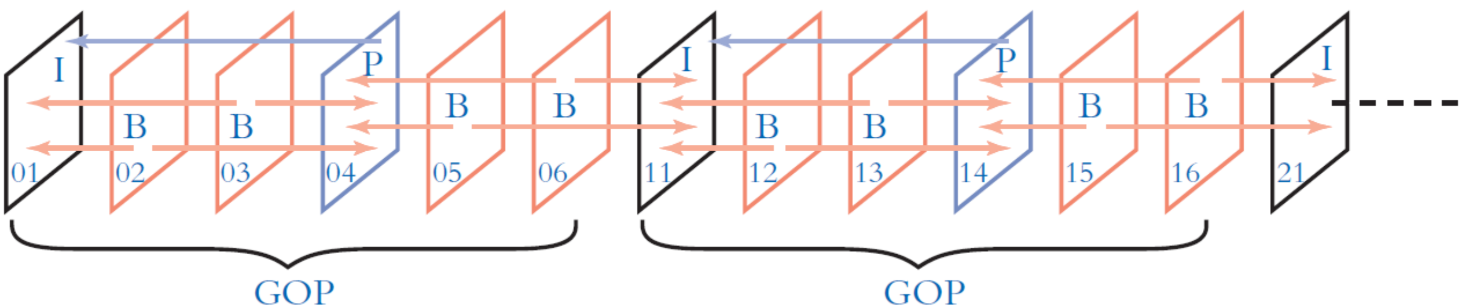
- All three types of picture are compressed using the MPEG-1 DCT-based compression method.
- P-pictures compress three times as much as I-pictures, and B-pictures one and a half times as much as P-pictures.
- Only I-picture allow random access.
- There is a trade-off to be made between compression and computational complexity when choosing the pattern of a GOP.



An MPEG sequence in bit stream order (for transmission)

Exercises

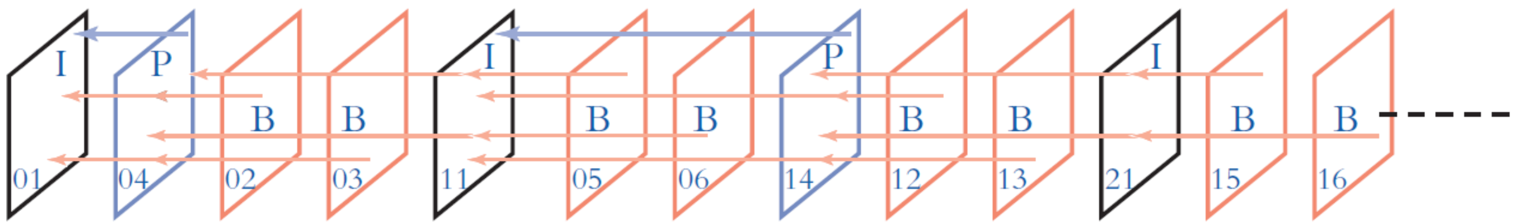
1. For a MPEG video sequence has a display order as IBBPBBIBBPBBI as below, work out the sequence in bit-stream order (for transmission) and explain why their orders are different.



2. Explain the differences of I, P and B-pictures.

Solution to 1

The key to this question is the B pictures has bi-directional referencing for motion compensation. So the following reference picture of B picture should to bring to the front.



Solution to 2

Explain the differences of I, P and B-pictures.

I-picture

- is spatial compressed.
- Allows random access.
- Has lowest compression ratio.

P-picture

- Uses motion compensation based on temporal compression
- Uses preceding I or P pictures as reference.
- Has higher compression ratio than I frame.

B-picture

- Uses bi-directional motion compensation.
- Takes following and preceding I and P-pictures as references.
- Achieves highest compression ratio and highest computational complexity.

H.264/MPEG-4 AVC

- it is jointly developed by ITU-T and ISO/IEC.
- It address the full range of video applications.
  - including standard-definition and high-definition broadcast television, video streaming over the Internet, delivery of high-definition DVD content, and the highest quality video for digital cinema applications.
- Before it, MPEG-2 has gained mass-market acceptance
  - in areas such as DVD and digital television broadcast (over cable and satellite).
- The new H.264/MPEG-4 AVC standard represents the single largest improvement in coding efficiency and quality since the introduction of MPEG-2.

Steps for H.264/MPEG4 AVC

- Motion estimation and intra estimation
- Transform (inverse transform)
- Quantization (and inverse quantization)
- Loop filter
- Entropy coding

Motion Estimation

- Motion estimation is used to identify and eliminate the temporal redundancies that exist between individual pictures.
- H.264 : *smaller macroblocks* to contain and isolate the motion.
- H.264 : *multiple reference frames.*

Intra Estimation

- Intra estimation is used to eliminate spatial redundancies.
- Intra estimation attempts to predict the current block by extrapolating the neighboring pixels from adjacent blocks in a defined set of different directions.
- The difference between the predicted block and the actual block is then coded.

Transform

- H.264/MPEG-4 AVC uses a DCT-like 4x4 integer transform.
- In contrast, MPEG-2 and MPEG-4 ASP employ a true DCT 8x8 transform that operates on floating-point coefficients.
- Smaller block size reduces blocking and ringing artefacts.

Quantization

- The coefficients from the transform stage are quantized, which reduces the overall precision of the integer coefficients and tends to eliminate high frequency coefficients, while maintaining perceptual quality.

Loop Filter

- H.264 has a de-blocking filter for Macroblocks to remove artifacts

Entropy Coding

- Lossless encoding
- Examples: run-length coding, Huffman coding, arithmetic coding

Entropy Encoding

- All compression systems require two algorithms:
  - Encoding at the source
  - Decoding at the destination
- Simple lossless compression algorithm is the **Run-length Encoding (RLE)**, where multiple occurring bytes are grouped together.
- What is the compression ratio of AAAAAAAABBBCCCCCCCDD?

Entropy Encoding

- We wish to construct and transmit a message using N symbols.
- Simplest method: use binary numbers of equal length L bits to represent each symbol ($L \geq log_2(N)$).
  - *Example: consider message of 5 symbols, then L will be approximately 3 bits ($L \geq log_2(5)$)*
  - This type of coding is called **fixed-length coding**
- **Fixed-length coding is not efficient – need statistical coding**

Entropy Encoding

- Statistical encoding:
  - Given sequence of symbols: s_1, s_2, …and probability of occurrence of each symbol $P(s_i)=P_i$
  - Example: P(A) = 0.16, P(B)=0.51, P(C)=0.09, P(D)=0.13, P(E)=0.11
    - Fixed-length coding encode A,B,C,D, E with 3 bits as A=000, B=001, C=010, D=011, E=100
  - Question: What is the minimum average number of bits per symbol in statistical encoding?
- The theoretical minimum average number of bits per codeword is known as **ENTROPY(H)**
- According to Shannon:- $\sum_i P_i log_2 P_i$ bits per codeword
  - Example above: - (0.16 x $\log_2$0.16 + 0.51 x $\log_2$0.51 + 0.09 x $\log_2$0.09 + 0.13 x $\log_2$0.13 + 0.11 x $\log_2$0.11) = 1.3 bits per codeword (approx)
  - This type of coding represents the **variable-length coding**

Huffman Encoding

- Statistical encoding
- To determine Huffman code, it is useful to construct a binary tree
- Leaves are characters to be encoded
- Nodes carry occurrences of the characters belonging to the subtree

- How does a Huffman code look like for symbols with statistical symbol occurrence probabilities:

$P(A) = 8/20, P(B) = 3/20, P(C) = 7/20, P(D) = 2/20$?

Summary

- Compression is very important due to the large amount of multimedia data, especially in video (e.g., HDTV)
- Consider lossless entropy coding schemes if you do not want to lose any data
- Keep in mind symmetric vs asymmetric compression schemes

Further Exploration

- JPEG Compression
- JPEG tutorial: http://johnloomis.org/ece563/notes/compression/jpeg/tutorial/jpegtut1.html
- JPEG online converter: http://www.jpeg-optimizer.com/

- MPEG-4 and H.264/AVC
- http://web.cs.ucla.edu/classes/fall03/cs218/paper/H.264_MPEG4_Tutorial.pdf