Multimedia Systems

Dr Laura Toni
l.toni@ucl.ac.uk

University College London

Today Lecture

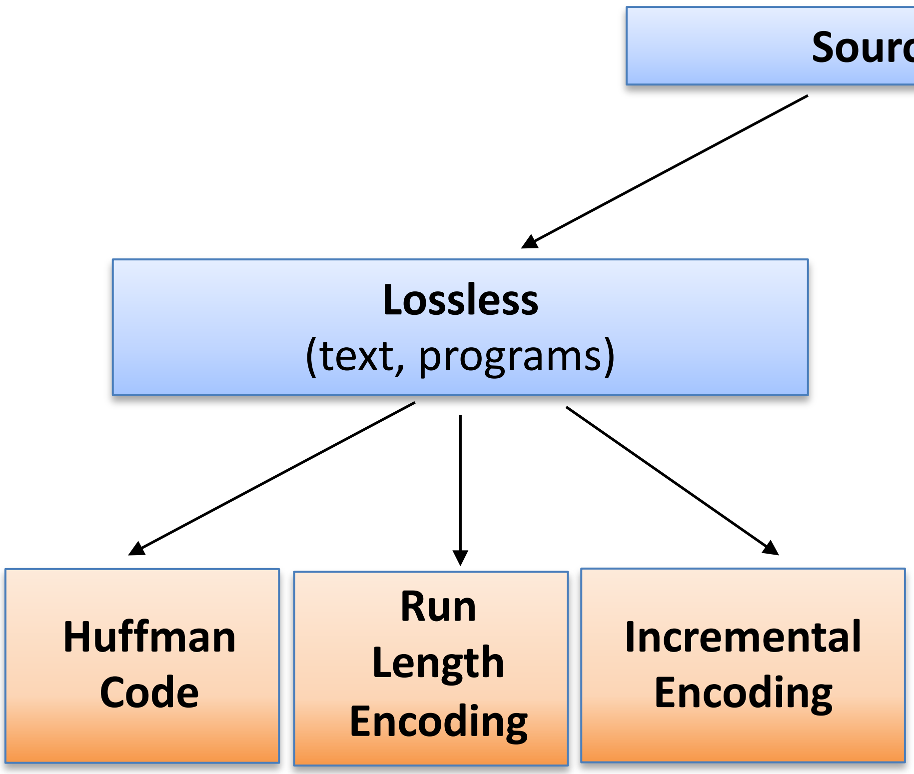
Introduction to Information compression

- Source Coding
- Information and Entropy
- Variable length coding
- Quantization

Multimedia Systems

- Image and Lossy Compression
  - Transforms
  - JPEG Quantization
  - JPEG Lossless Compression
- Video Compression
  - Motion Compensation

Source Coding



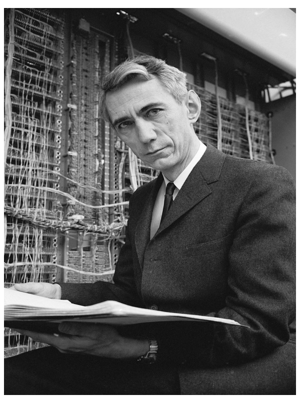
Source Coding

Source Encoding

Lossless

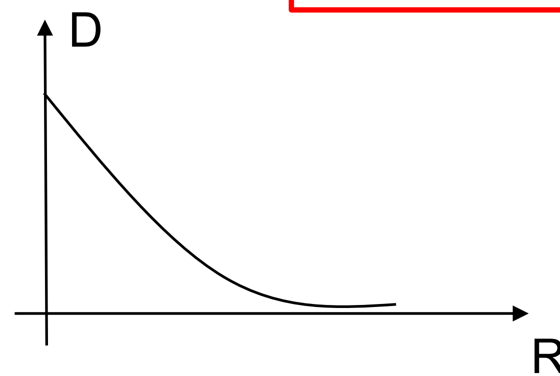
Lossy

Speech and Video coding



Shannon’s first theorem (1948)

Expected code length relates to the source entropy



Source recovered with a distortion D (function of the coding rate)

Why do we compress?

These are related to the 3 types of redundancy in images/videos:

- Interpixel redundancy
- Psychovisual redundancy
- Coding redundancy

Coding Redundancy

- Some colors are more common than others
- For example, black, brown, and red hardly appear in this picture
- This is sometimes called *coding redundancy*

Interpixel Redundancy

- Blue pixels tend to occur next to other blue pixels; yellow pixels are near other yellow pixels
- This spatial correlation is sometimes called *interpixel redundancy*
- There are also interspectral and interframe redundancy

Psychovisual Redundancy

- Some parts of the scene are very homogeneous (sky)
- Other parts are very busy (flowers) and could hide noise
- This is sometimes called *psychovisual redundancy*

Why Lossy?

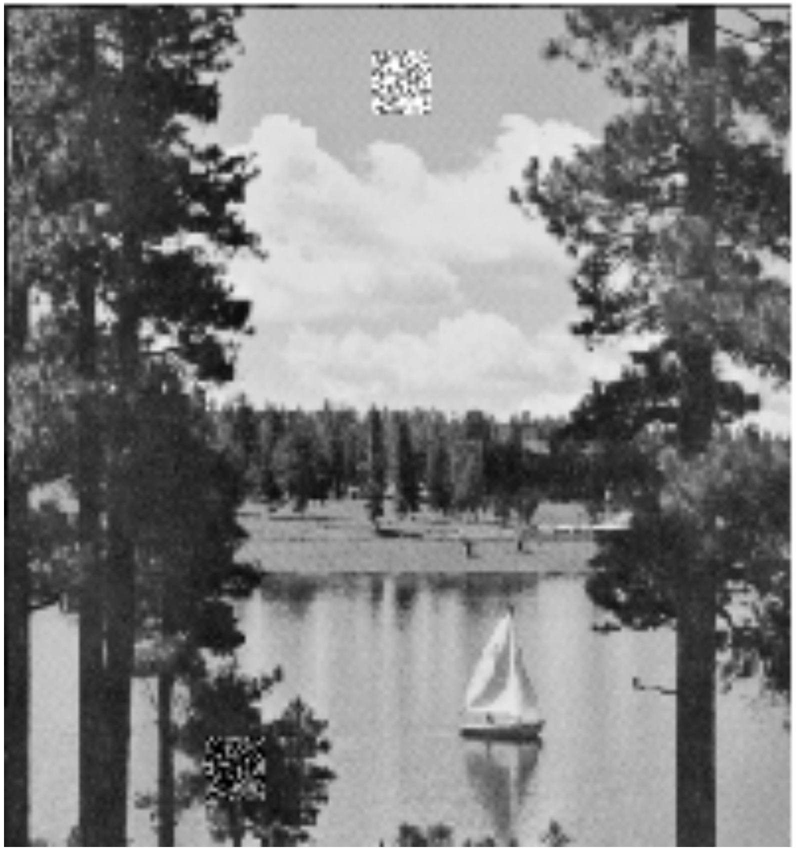
Human Visual System Issues

- We can get away with lossy compression because *your eye doesn't see everything anyway*
  - Contrast sensitivity function
  - Mach bands
  - Spatial masking
  - Oblique effect

Spatial Masking

- A stimulus is harder to see in the presence of large visible spatial and temporal changes in luminance
- Line presented near a luminance edge is harder to see as it gets closer

Spatial Masking



Implication: Can allow more error in busy parts of the picture

How well is the compression doing?

- We would like to have a low bit rate and yet a high image/video quality

  (there are also other factors – not covered today- such as complexity, error resilience, delay etc.)

- If you use lossy compression, need to be able to measure the quality

MSE Metric

- The most common computable measures are the MSE

$$MSE = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} (F(i,j) - G(i,j))^2$$

where F is the input image, G is the output image, and the images are of size N by M

PSNR Metric

- The MSE is often reported in logarithmic form as a signal-to-noise ratio:

$$SNR = 10\log_{10}\frac{D_0}{MSE}\,dB$$

where D_0 is a normalization factor

  - D_o often chosen to be the square of the maximum possible input value (e.g., 255^2)
  - Then it's called a "peak SNR" or PSNR

JPEG Example

Original

1 bpp

0.5 bpp

0.25 bpp

Thank You

Multimedia Systems

Dr Laura Toni
l.toni@ucl.ac.uk

University College London

Today Lecture

Introduction to Information compression

- Source Coding
- Information and Entropy
- Variable length coding
- Quantization

Multimedia Systems

- Image and Lossy Compression
    - Transforms
    - JPEG Quantization
    - JPEG Lossless Compression

- Video Compression
    - Motion Compensation

Block Diagram for Image Compression

Signal Decomposition:

– Compact energy into a small number of coefficients
– Decorrelate the components of the signal

Quantization:

– Make approximations to the transform coefficients
– Selectively throw away information

Lossless coding

– Variable length coding (e.g., Huffman Coding)
– Choose short/long codewords to minimize number of bits

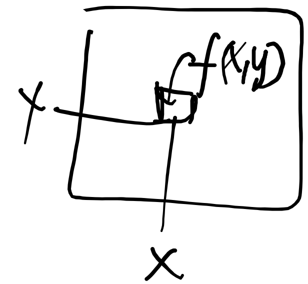
Transform Coding

$$x \rightarrow \boxed{A} \rightarrow u \qquad\qquad u \rightarrow \boxed{A^{-1}} \rightarrow x$$

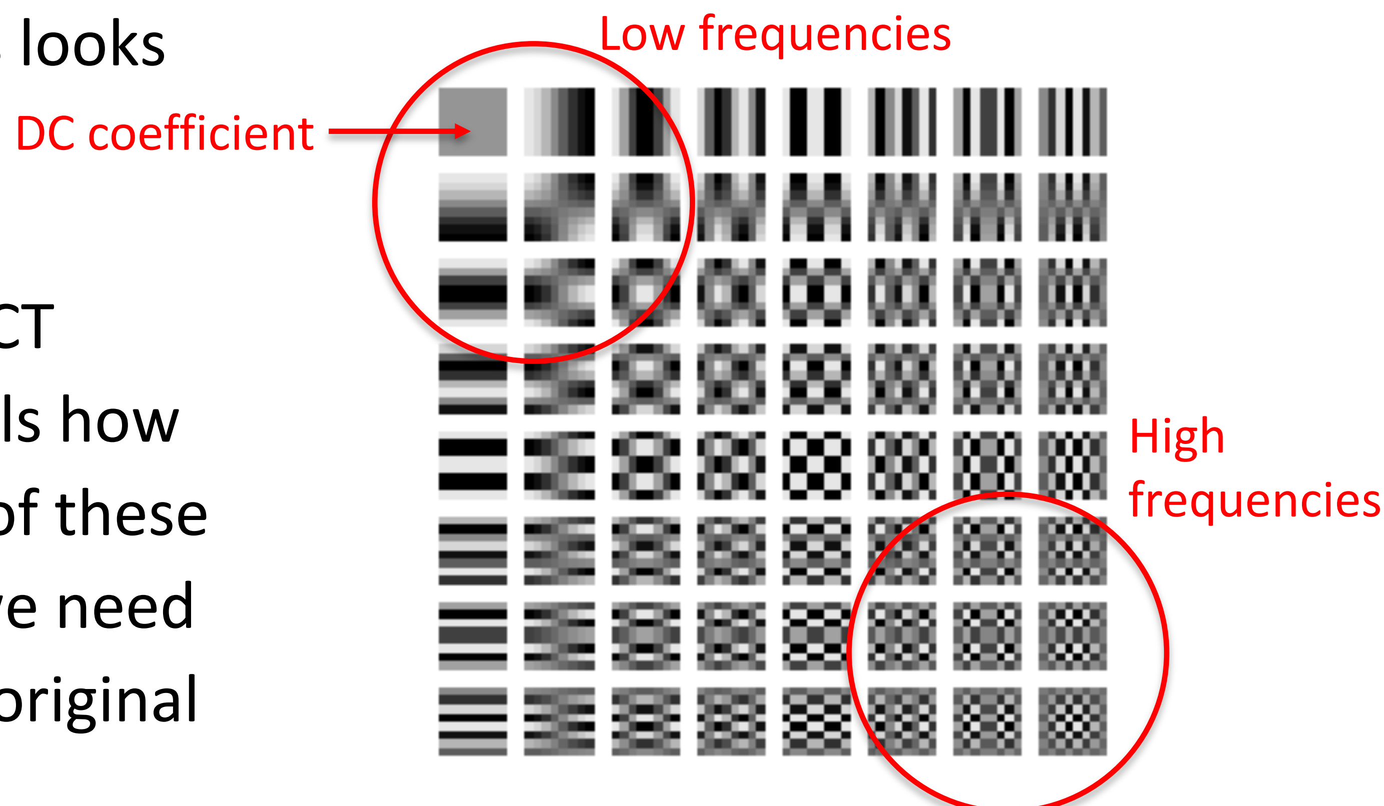
- We are interested in transforming a signal or image from the **spatial domain** to the **frequency domain**
- by itself, the transform – does not save any bits
  – does not introduce any distortion
- both of these happen when we throw away information - "lossy compression" implemented by the quantizer

2-d DCT Basis Functions

$$C(u,v)=\alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right]$$



- The whole array of basis functions looks like this:



The array of DCT coefficients tells how much of each of these basis images we need to regenerate original image

$$x_t \rightarrow \boxed{\mathcal{F}} \rightarrow \underline{X(f)} \rightarrow \boxed{\mathcal{F}^{-1}} \rightarrow x_t$$

Why is the DCT a good idea?

smooth edge

- Smooth areas in the images have pixels highly correlated
- Pixels around edges are less correlated

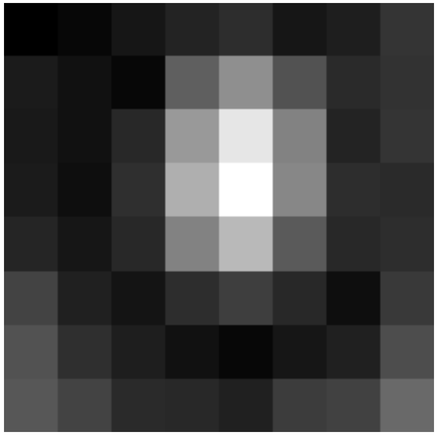
- Our visual system is less sensitive to distortion around edges
- We want to protect smooth areas the most

- DCT has excellent energy compaction for highly correlated data (smooth areas).
- Low frequencies in the DCT correspond to smooth areas
- We can quantize less heavily low frequencies

low fr. ← smooth → more sensitive

hig fr. ← edge → less sensitive

Quantization after Transforming



N

N

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

DCT

N

N

$$\begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix}$$

- After the DCT, we still have N^2 numbers to represent a block:

  N^2 pixels $\leftrightarrow$ N^2 coefficients

- However, the N^2 coefficients will have
  - Lot of values near zero
  - Lot of values which represent high frequency info

Block Diagram for Image Compression

Signal Decomposition:

- Compact energy into a small number of coefficients
- Decorrelate the components of the signal

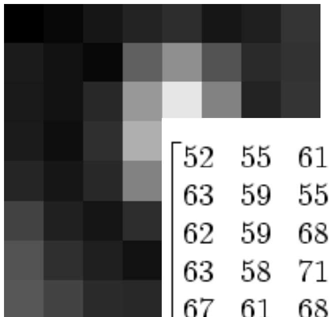
Quantization:

- Make approximations to the transform coefficients
- Selectively throw away information

Lossless coding

- Variable length coding (e.g., Huffman Coding)
- Choose short/long codewords to minimize number of bits

Let’s continue the example



$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

source

Pixel intensity (N^2 values)

DCT

$$\begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix}$$

DCT

Transformed coordinates (N^2 values)

How can we apply the quantizer?

Let's continue the example

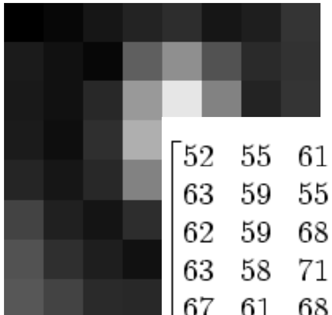
We consider a quantizer matrix, in which higher frequency are quantized more heavily

Low frequencies

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

High frequencies

Let's continue the example



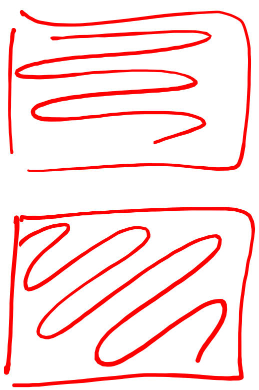
$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

source

DCT

$$\begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix}$$

DCT



Q matrix

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

Quantizer

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

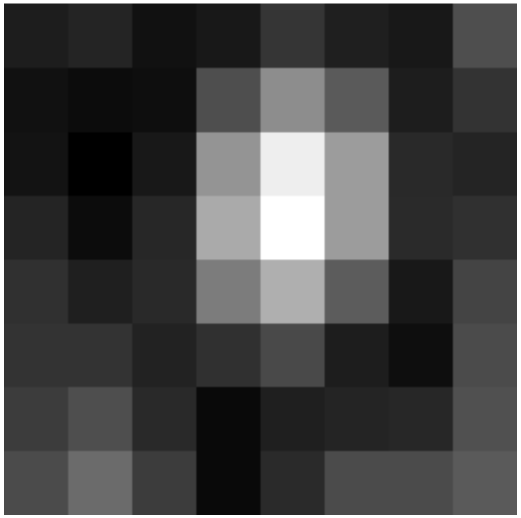
Quantized DCT

−26, −3, 0, −3, −2, −6,

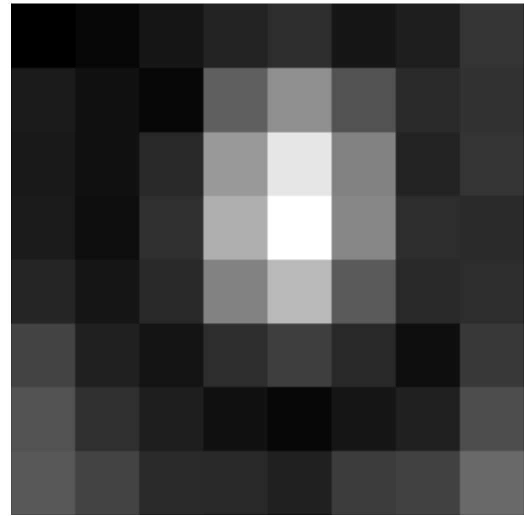
→ (0, 30)

Can we reconstruct the image?

original



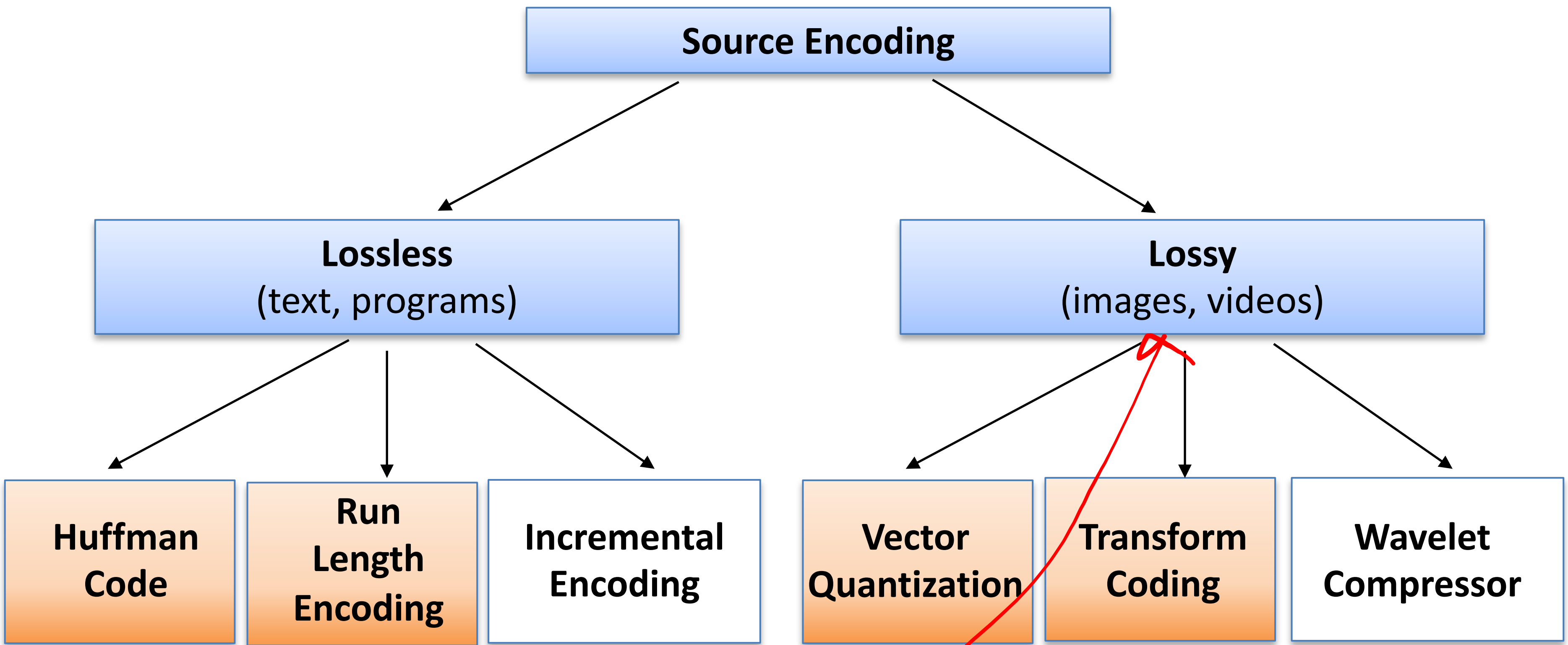
decompressed

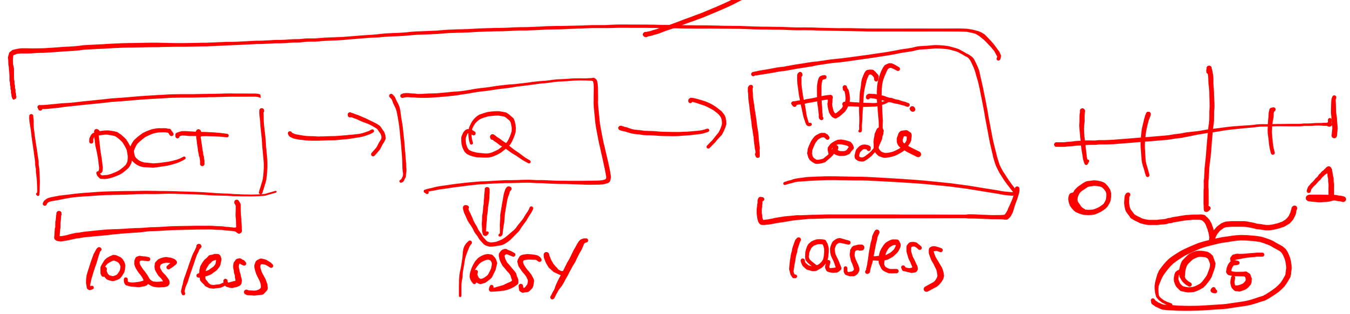


however visually the blocks are not very different

Hence, we highly compressed the image without having “perceptual” losses!

Source Coding





Take Home Message

- Information theory tells us to look at the probability of the source when coding (most probable words, shortest bits)
- Quantizer follows the same concept: quantize heavily the least probable ranges

- Image compression is composed of
  - There is redundancy in the images.
  - Artefacts in high-frequencies (edges) are less noticeable from the human eye
  - Source transformation (to compact the energy – most high frequency values will be low)
  - Quantizer (2D quantizer that will cut off high-frequencies)
  - Zig-zag rastering will maximize the probability of having 0s
  - Huffman lossless code as last step

Thank You

Multimedia Systems

Dr Laura Toni
l.toni@ucl.ac.uk

University College London

Today Lecture

Introduction to Information compression

- Source Coding
- Information and Entropy
- Variable length coding
- Quantization

Multimedia Systems

- Image and Lossy Compression
  - Transforms
  - JPEG Quantization
  - JPEG Lossless Compression

- Video Compression
  - Motion Compensation

Compressing Digital Video

- Exploit *temporal redundancy* between frames
  - Only the sun has changed position between these 2 frames

Previous Frame

Current Frame

- Exploit *spatial redundancy* within frames (like JPEG: transform, quantize, Variable Length Coding)

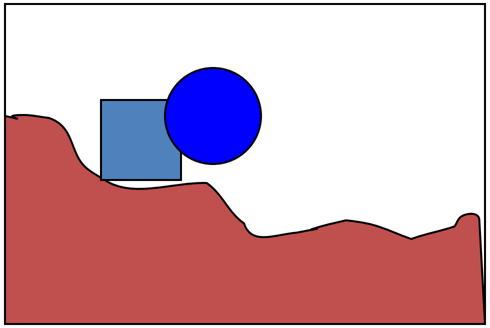
Difference Frames

- Differences between two frames can be caused by
  - Camera motion: the outlines of background or stationary objects can be seen in the Diff Image
  - Object motion: the outlines of moving objects can be seen in the Diff Image
  - Illumination changes (sun rising, headlights, etc.)
  - Noise
  - Scene cuts

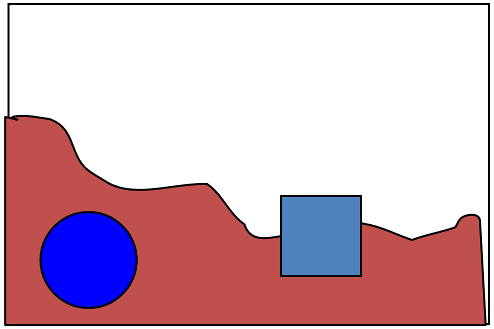
- This difference might still have some correlation – we want to remove it with compression

Types of Motion: Translation

- Translational motion: simple movement of typically rigid objects
- Camera pans vs. movement of objects



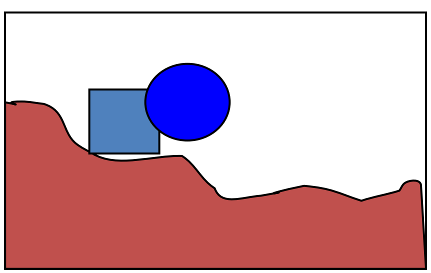
Frame n



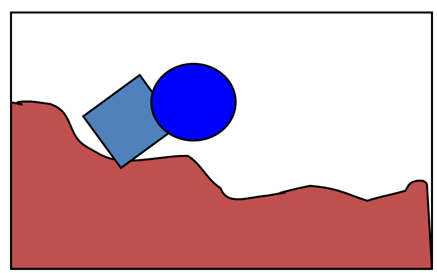
Frame n+1

Types of Motion: Rotation & Zoom

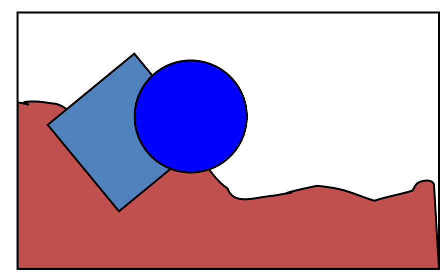
- Rotation: spinning about an axis
  - Camera versus object rotation
- Zooms – zoom in, zoom out
  - Camera zoom vs. object zoom (movement in and out)



Frame n



Frame n+1 (Rotation)



Frame n+2 (Zoom)

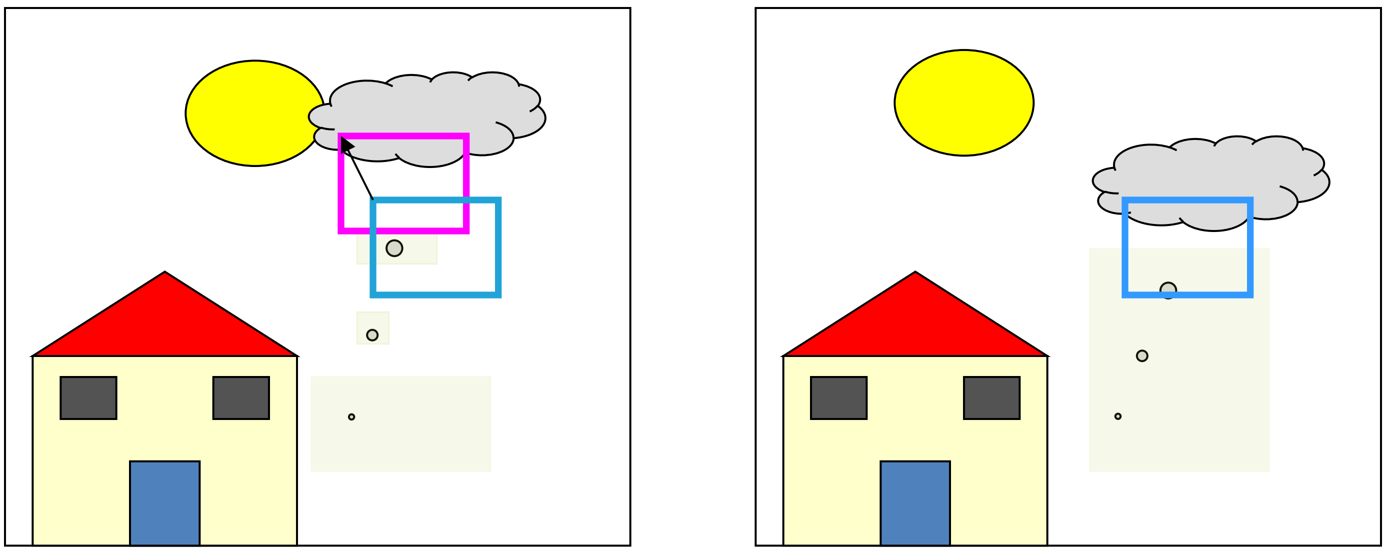
Motion Estimation

- For some portion of the frame, estimate its movement between 2 frames- the *current frame* and the *reference frame*

- What is some portion?
  - Individual pixels (all of them)?
  - Lines/edges (have to find them first)
  - Objects (must define them)
  - Uniform regions (just chop up the frame)

Motion Vectors

A *motion vector* (MV) describes the offset between the location of the block being coded (in the current frame) and the location of the best-match block in the reference frame

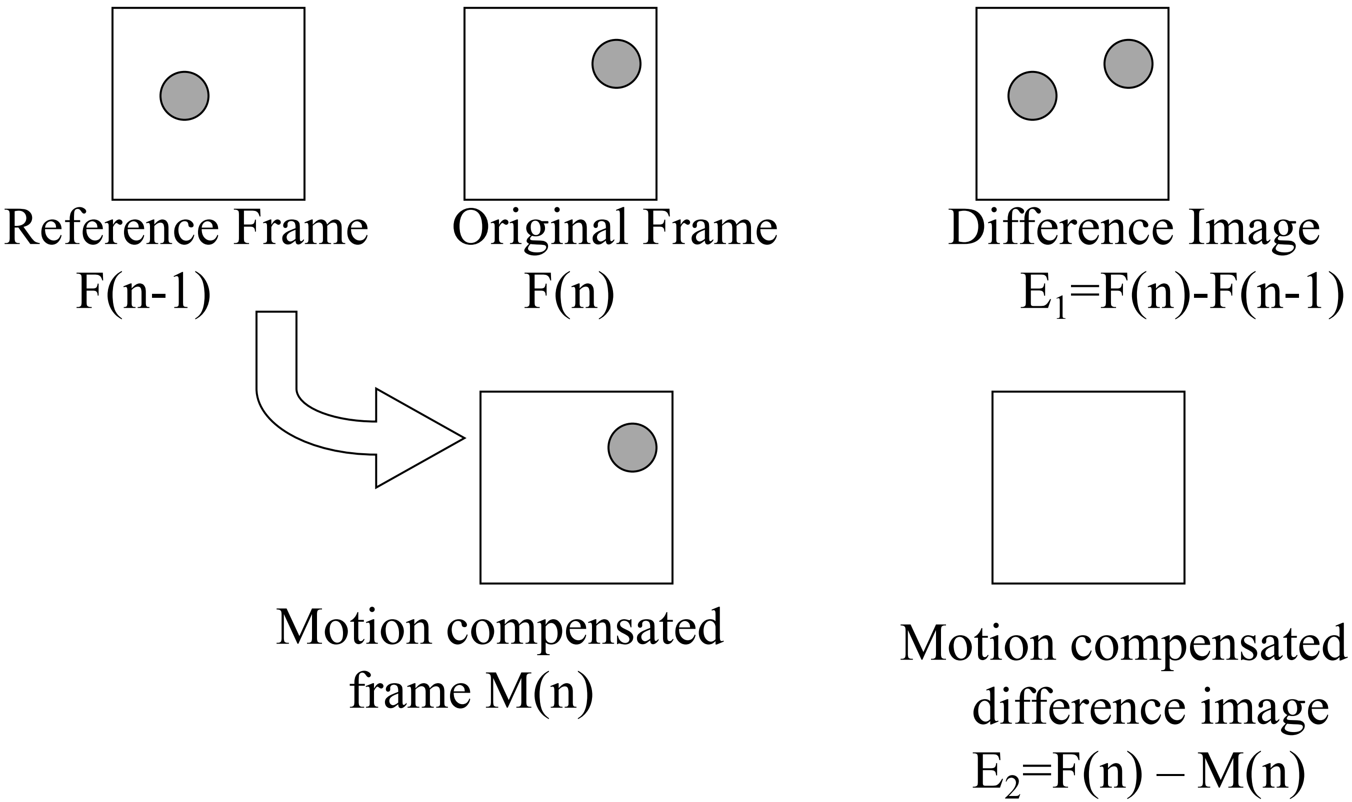


T=1 (reference) T=2 (current)

Motion Compensation

- This glued together frame is called the *motion compensated frame*
- The encoder can also form the difference between the motion compensated frame and the actual frame.
- This is called the *motion compensated difference frame*
- This difference frame formed using MC should have less correlation between pixels than the difference frame formed without using MC

Motion compensated difference frames

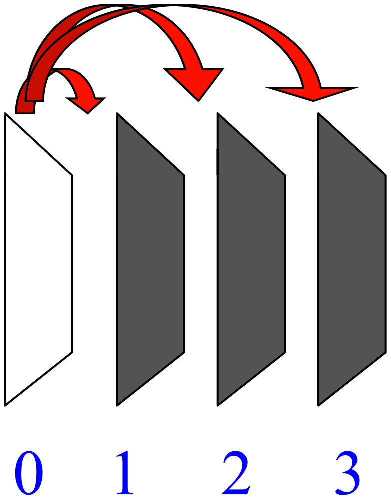


- E_1 & E_2 called prediction error or difference frames
- Do we prefer to compress and send E_1 or MV+E_2 ?

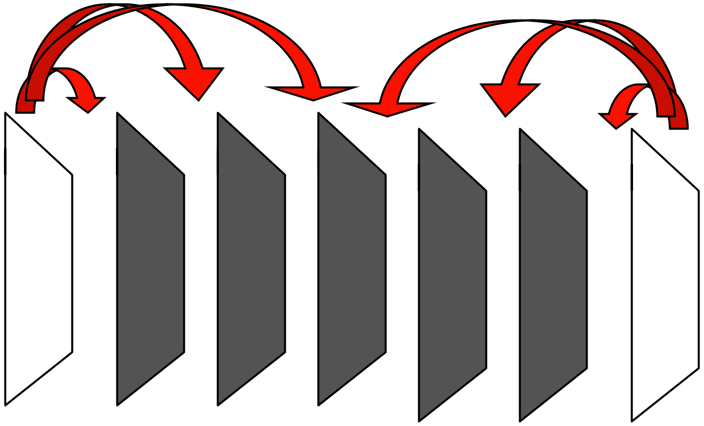
Motion estimation philosophy

- *Most of the time,* for a given representation quality
  - fewer bits to send MV+E_2 instead of sending E_1
  - fewer bits to send E_1 instead of sending F itself.

Reference Frame



Temporal Location of Reference



Thank You

Today Lecture

Introduction to Information compression

- Source Coding
- Information and Entropy
- Variable length coding
- Quantization

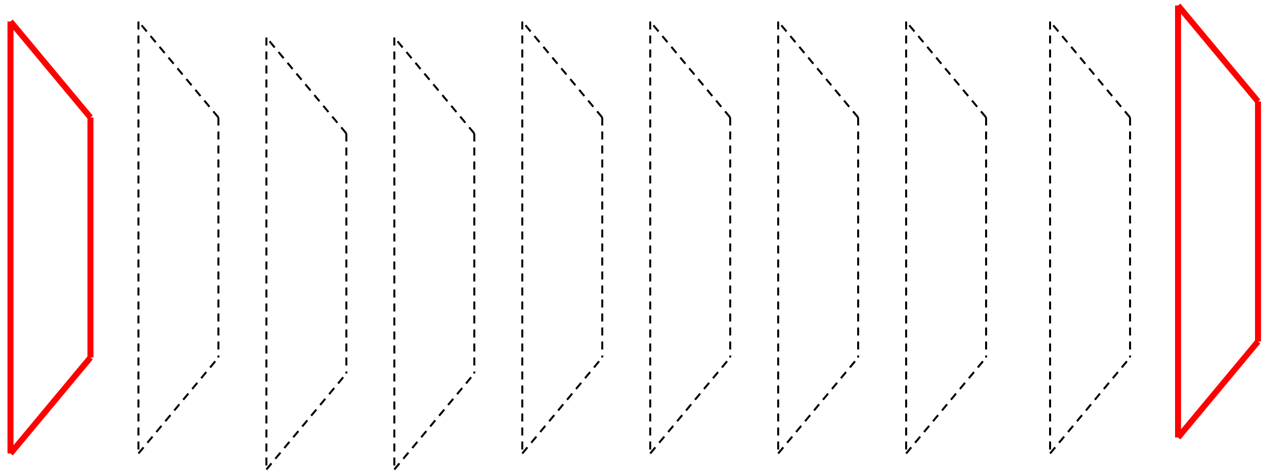
Multimedia Systems

- Image and Lossy Compression
  - Transforms
  - JPEG Quantization
  - JPEG Lossless Compression

- Video Compression
  - Motion Compensation

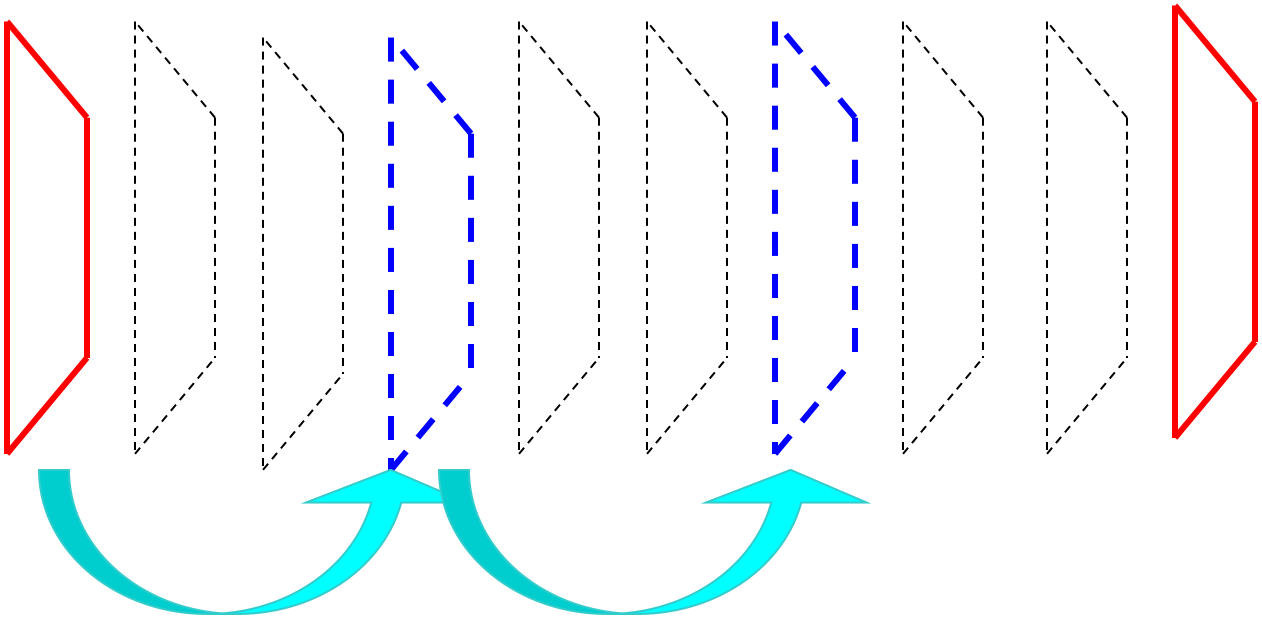
MPEG Frame Types

- Intra (I) pictures: coded by themselves, as still images. No temporal coding. No motion vectors.



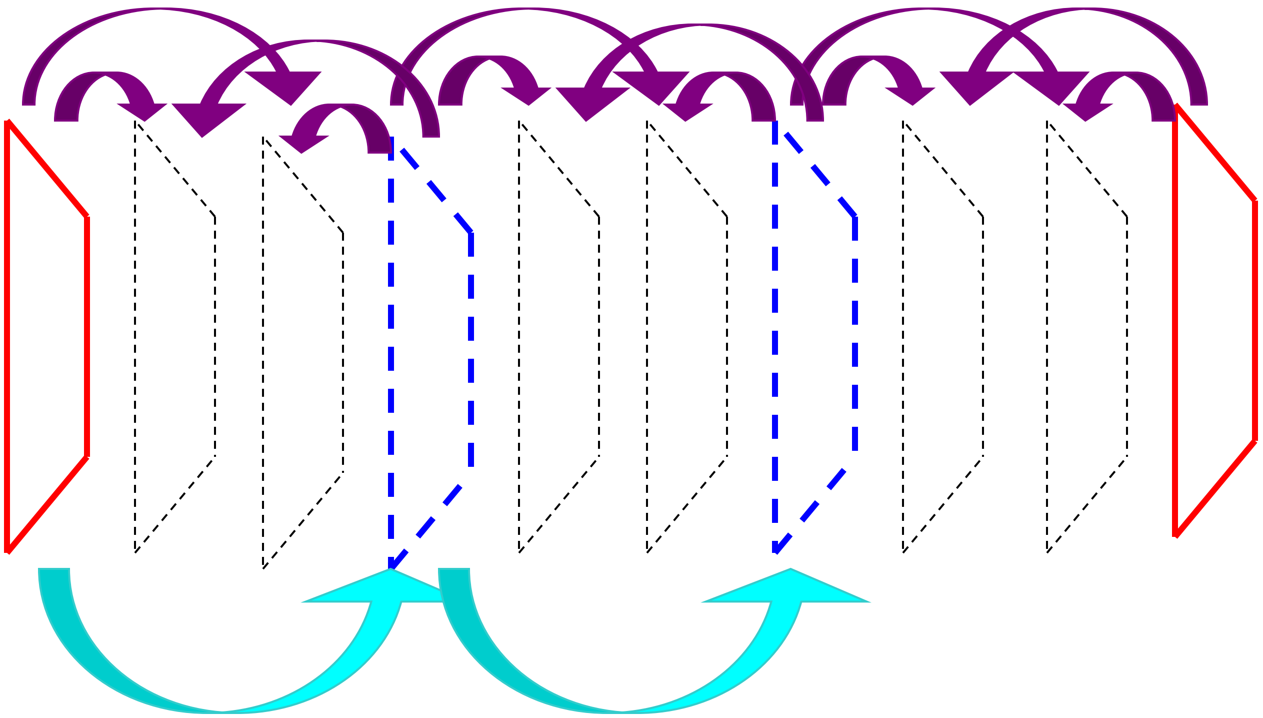
MPEG Frame Types

- Forward Motion Compensated predicted (P) pictures – forward motion compensated from the previous I or P frame

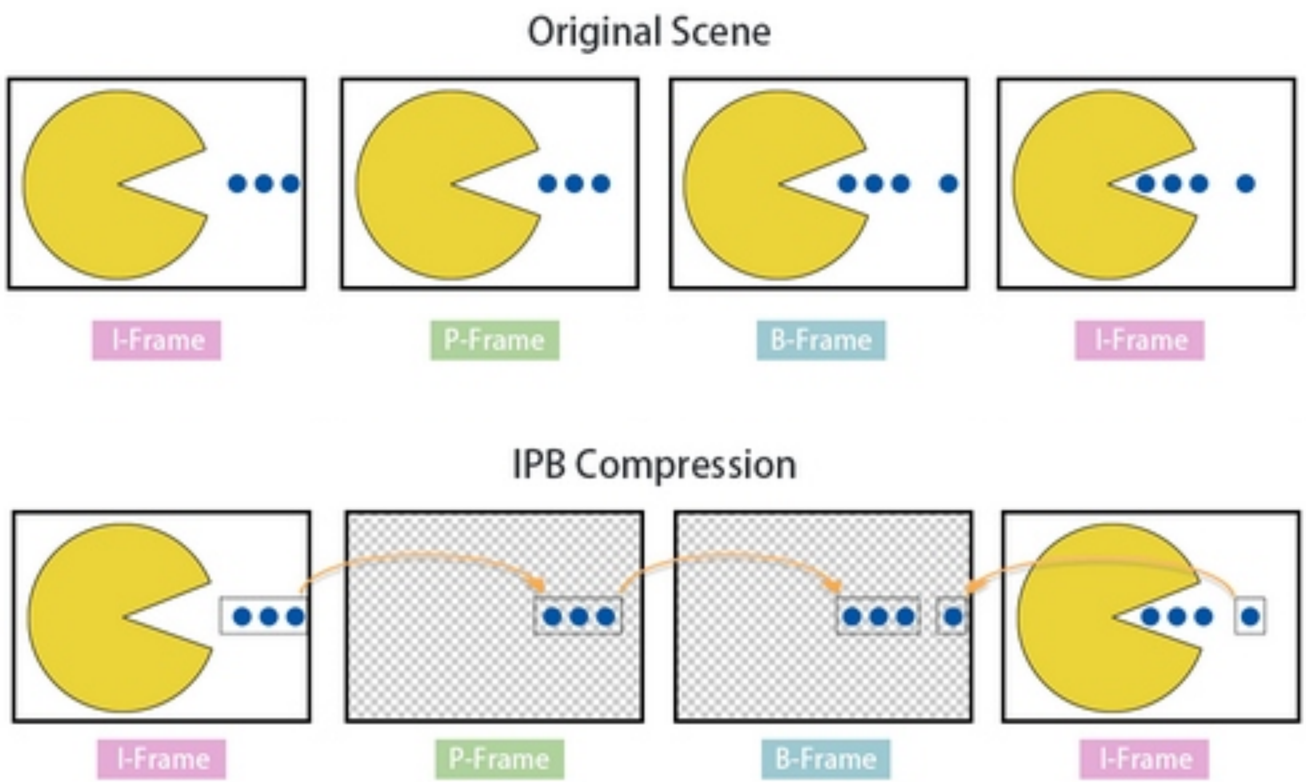


MPEG Frame Types

- Motion Compensated interpolated (B) pictures – forward, backward, and interpolatively motion compensated from previous/next I/P frames

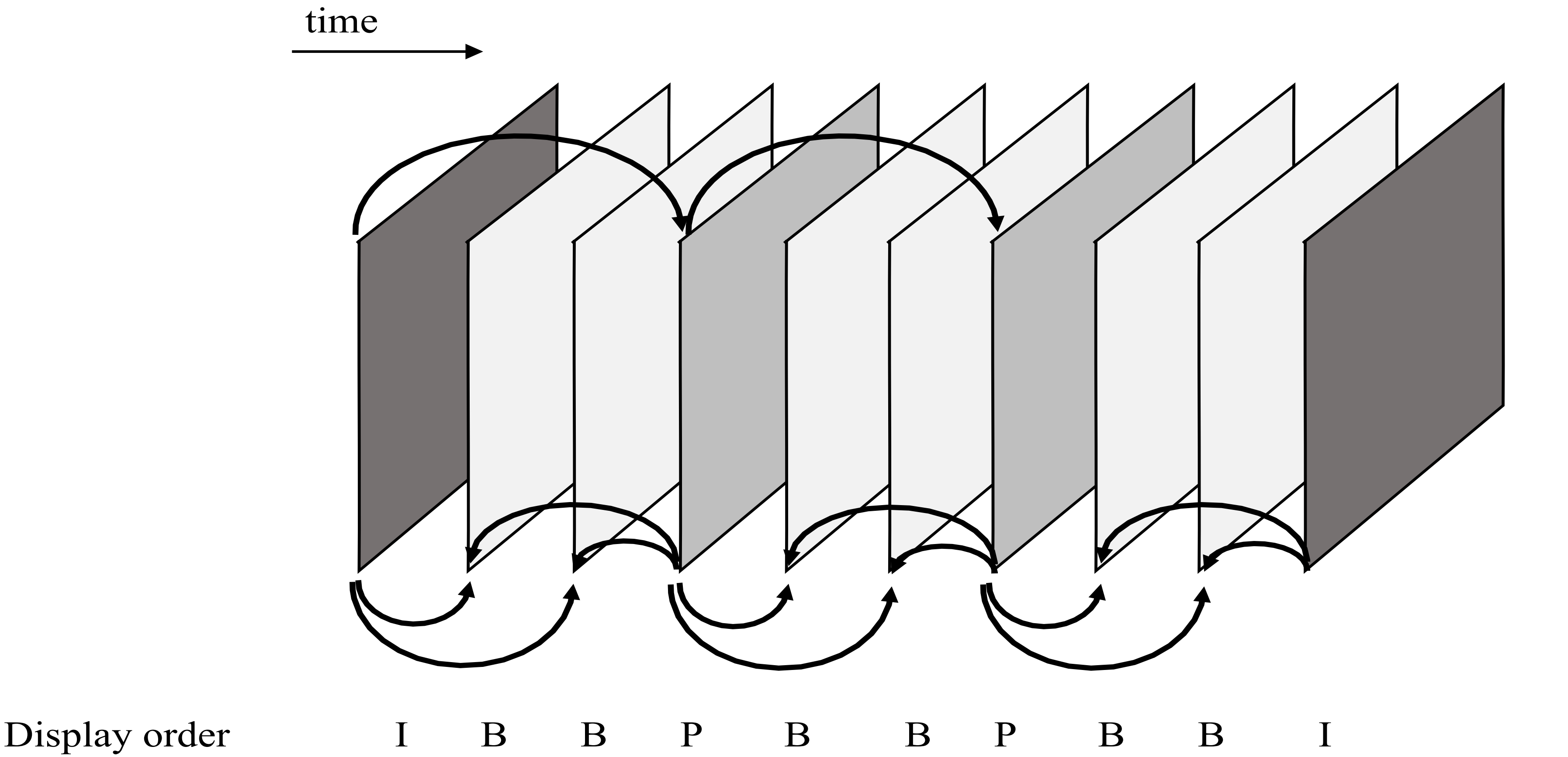


I-P-B Structure - example

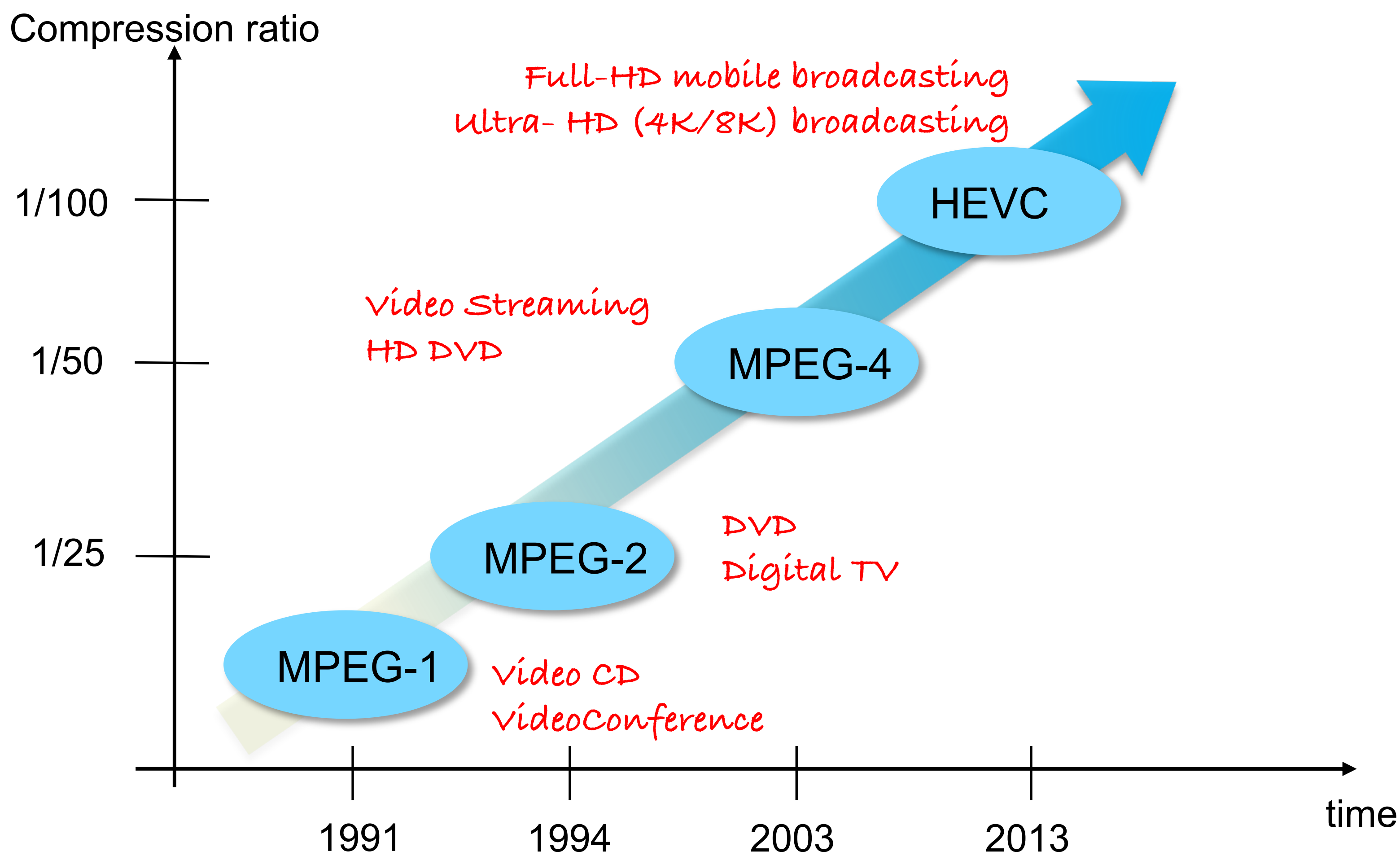


Graphical example that shows how in P and B frames we actually only compress the "difference" with respect to one (or more) reference frame

I-P-B Structure - example



History of Video Compression Standards



Competing Codecs

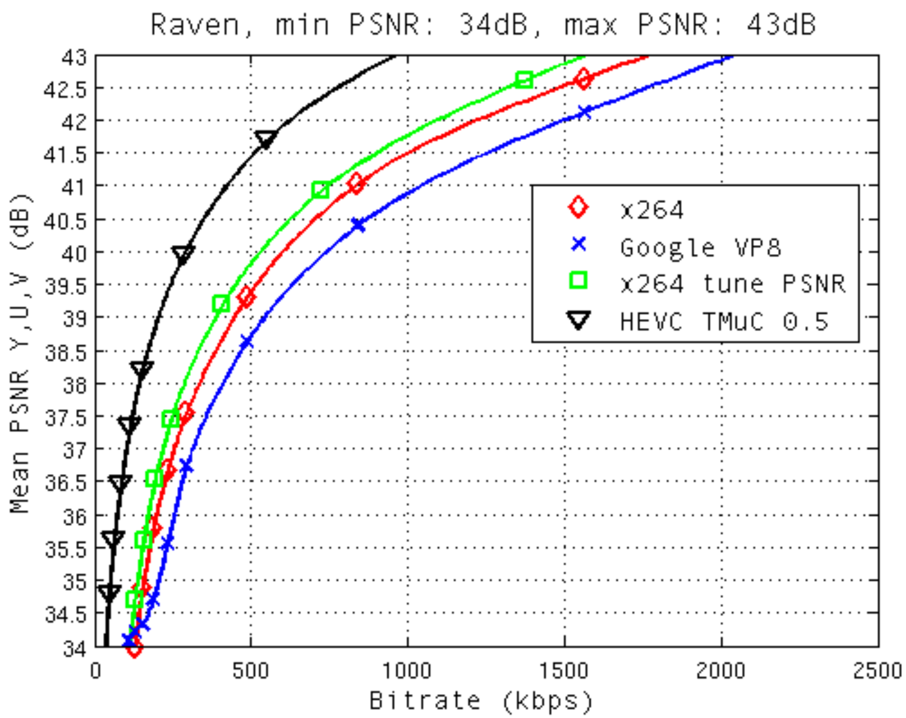
- VP8/VP9,
- AV1
- MPEG/ITU HEVC/H.265

- use scalar or vector quantization
- use temporal compression
- highly asymmetric
- 4K full-HD videos
- Compression up to 50% compared to previous codecs

Competing Codecs

- VP8/VP9,
- AV1
- MPEG/ITU HEVC/H.265

Performance comparison



"Rate-distortion performance of contemporary video codecs: Comparison of Google/WebM VP8, AVC/H. 264, and HEVC TMuC", E Ohwovoriole and Y Andreopoulos - LENS Symp., London, 2010. http://www.ee.ucl.ac.uk/~iandreop/OHWOVORIOLE_LCS_2010_H264_VP8_HEVC_comparison.pdf

Raw Bandwidth

Raw Bandwidth = color depths * W * L * refresh frequency

Resolution (WxL)	24bit@25fps	8bit@25fps	8bit@15fps	8bit@5fps
4K (3840x2160)	5 Gbit/s	1.6 Gbit/s	995 Mbit/s	331 Mbit/s
HDTV (1920x1080)	1.3 Gbit/s	414 Mbit/s	248 Mbit/s	83 Mbit/s
VGA (640x480)	184 Mbit/s	61 Mbit/s	36 Mbit/s	12 Mbit/s
SCIF (704x576)	240 Mbit/s	80 Mbit/s	48 Mbit/s	16 Mbit/s
CIF (352x288)	60 Mbit/s	60 Mbit/s	12 Mbit/s	4 Mbit/s
QCIF (176x144)	12 Mbit/s	12 Mbit/s	3 Mbit/s	1 Mbit/s

- Uncompressed video is **BIG**
- A DVD would hold max 5 secs of uncompressed video at 1920x1080 resolution
- 6 MHz channel for transmission therefore maximum possible bitrate is 18 Mb/sec
- Requires compression of 83:1

Future Video Applications

Immersive Communications
(holoportation, from Microsoft)

Virtual Reality,
360 Videos

Telepresence
(from Cisco)

Speech/Audio/Video Coding Summary

- Requirements for Coding of each Source
- Speech Coding Principle
- Discontinuous Transmission
- GSM Speech Coding
  - RPE-LPC
  - Enhanced Full Rate
- Different Video Coding Formats, motion compensation

Web References

- 3G Latest

  http://www.umts-forum.org/
- GSM Information

  http://www.gsmworld.com/
- Fraunhofer Full-HD Voice

  http://www.full-hd-voice.com/
- MPEG Information

  http://www.chiariglione.org/mpeg/
- Speech Coding

  http://www-mobile.ecs.soton.ac.uk/speech_codecs/
- Netflix TechBlog

  http://techblog.netflix.com/search/label/encoding