

Introduction to Image and Video Coding

© Iain Richardson / Vcodex Ltd 2001-2002



Outline

- The need for compression
- The coding model
- Image coding
- Video coding
- Standards

The need for compression

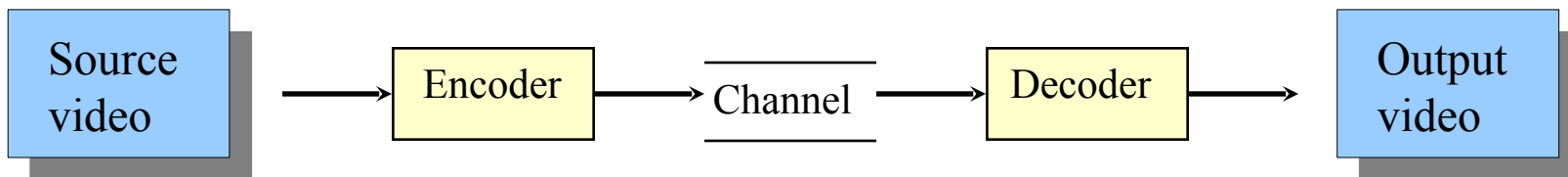
- Colour image, 352x288 pixels
- “Full” colour depth: 24 bits per pixel (8 bits red, green, blue)
 - 304128 bytes
- Reduced colour depth: 12 bits per pixel
 - 152064 bytes



The need for compression

- Video signal: 25 frames per second
- “VHS video” quality: 352x288 pixels per frame, 12 bits per pixel
 - 30.4 Mbits per second
- “Television” quality: 704x576 pixels per frame, 12 bits per pixel
 - 121.7 Mbits per second
- too much data for cost-effective transmission or storage
- need compression

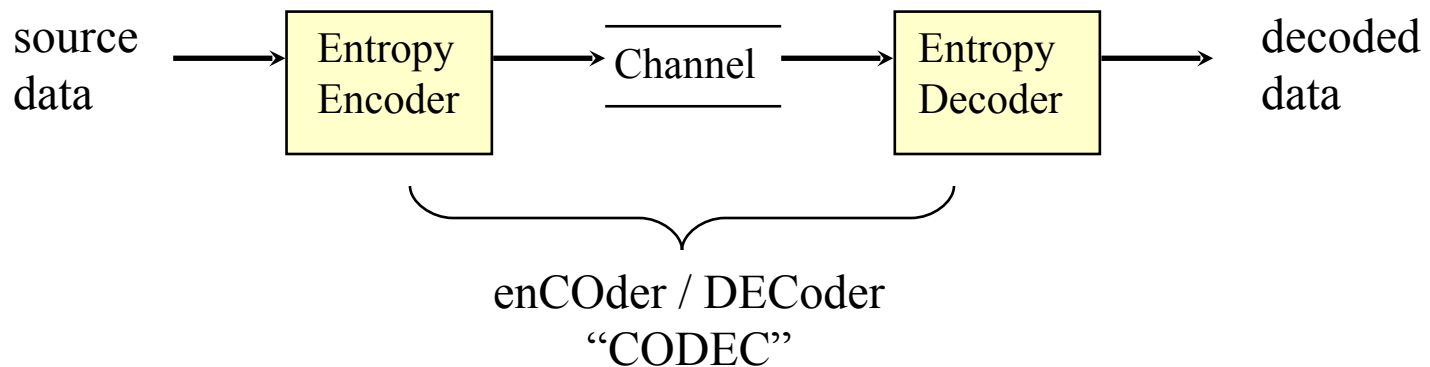
Image or Video CODEC



- Encode (compress) and decode (decompress) still images or moving video
- Key issues:
 - compression efficiency and image quality
 - computational complexity
 - frame rate

The coding model

- General-purpose compression: entropy encoding
 - remove **statistical redundancy** from data
 - e.g. encode common values with short codes, uncommon values with longer codes
- Good for text files, poor for images / video

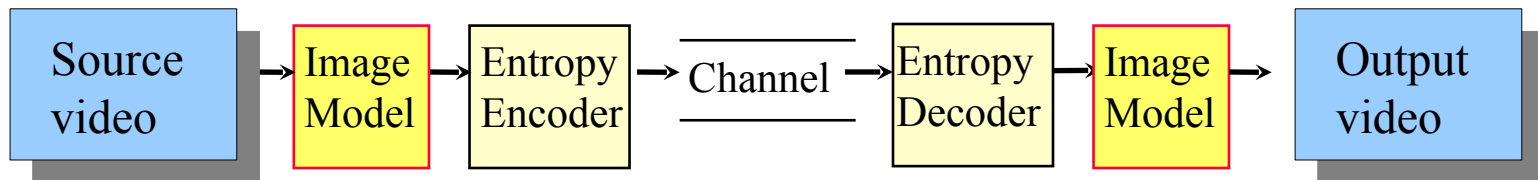


Outline

- The need for compression
- **The coding model**
- Image coding
- Video coding
- Standards

The coding model

- Solution: add a **model** that attempts to represent the image/video signal in a form that can be easily compressed by the entropy encoder
- model exploits the **subjective redundancy** of images and video
- decoded image may not be identical to original image



The coding model

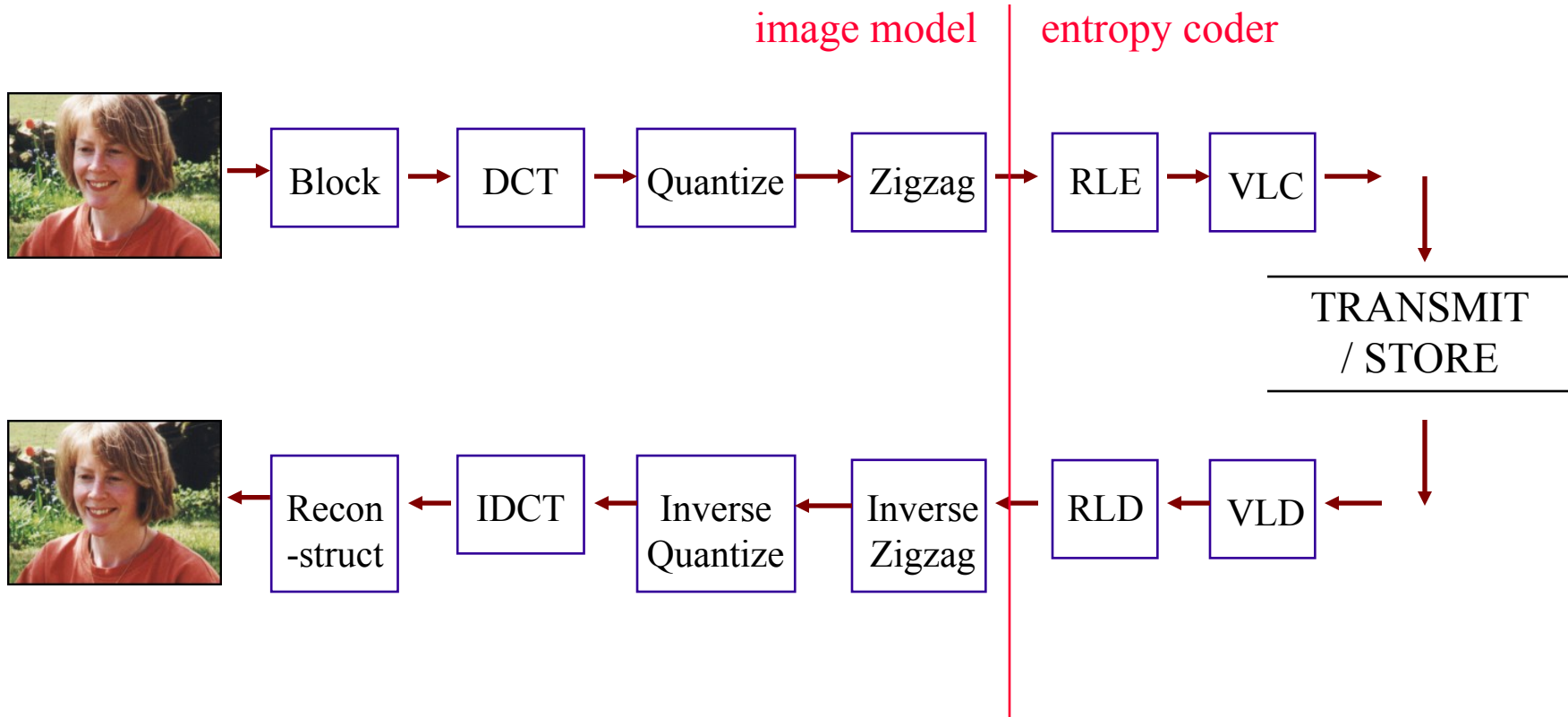
- Image properties that are useful for compression
 - many of the pixels of a typical photographic image contain little or no “useful” detail (e.g. “flat” areas)
 - the eye is insensitive to “high frequency” image information



Outline

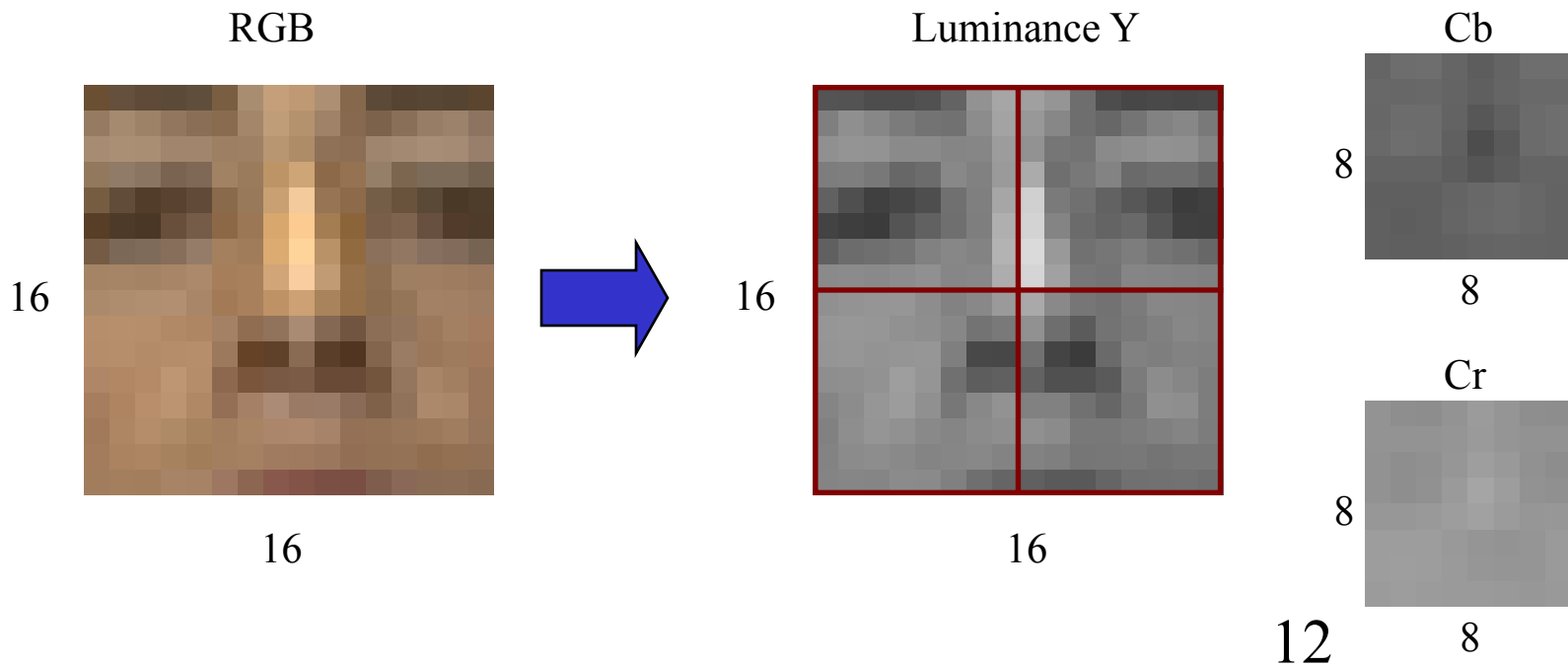
- The need for compression
- The coding model
- **Image coding**
- Video coding
- Standards

Image CODEC (e.g. JPEG)



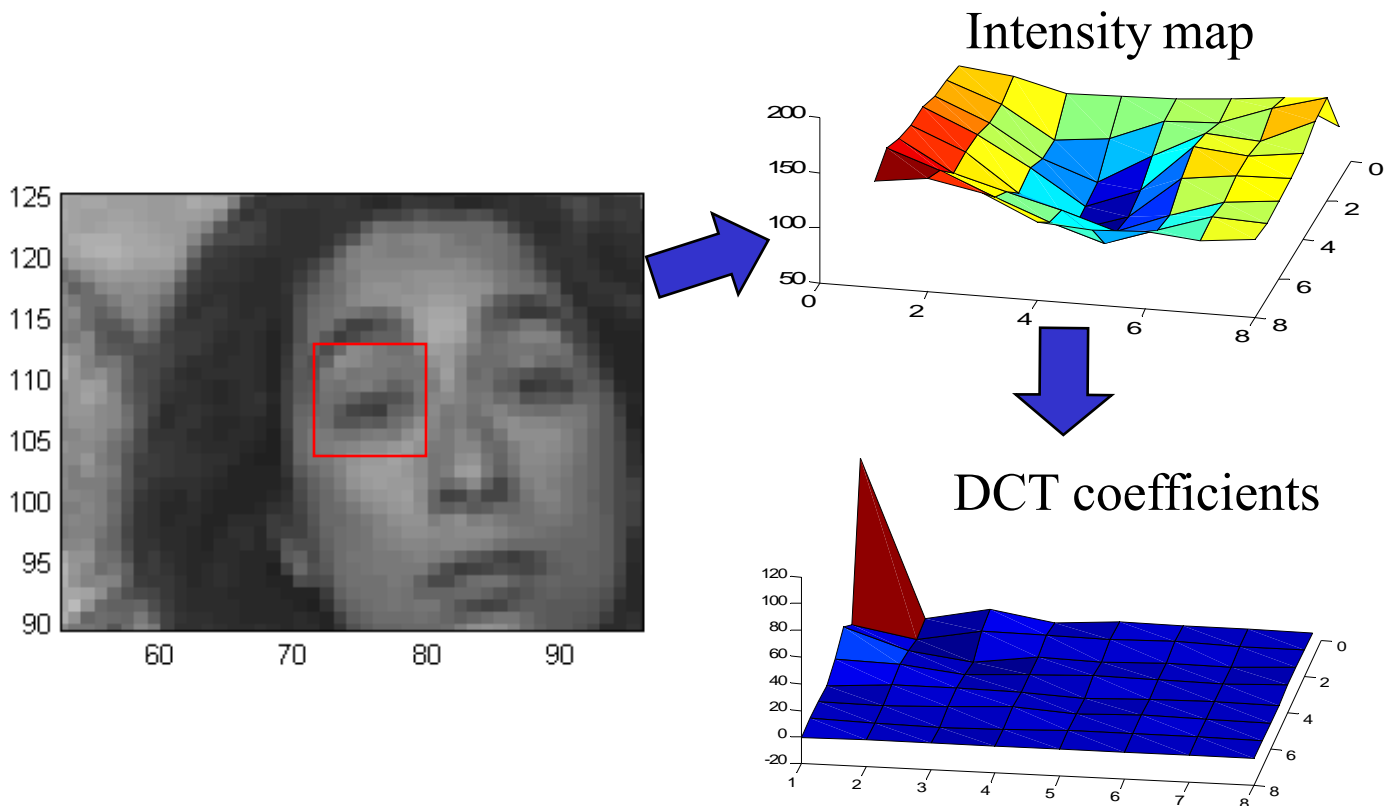
Blocks

- Process the data in blocks of 8x8 samples
- Convert Red-Green-Blue into Luminance (greyscale) and Chrominance (Blue colour difference and Red colour difference)
- Use half resolution for Chrominance (because eye is more sensitive to greyscale than to colour)



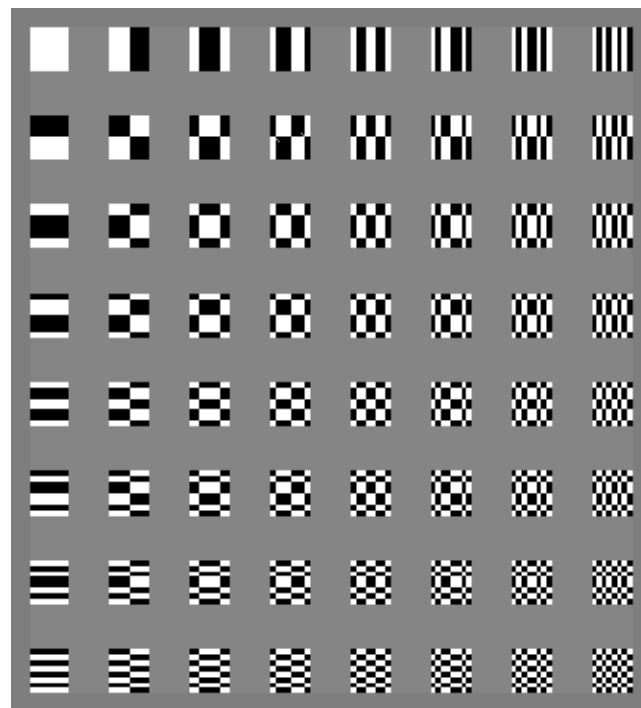
Discrete Cosine Transform

- Transform each block of 8x8 samples into a block of 8x8 spatial frequency coefficients
 - energy tends to be concentrated into a few significant coefficients
 - other coefficients are close to zero / insignificant



Discrete Cosine Transform

- Any 8x8 block of pixels can be represented as a sum of 64 **basis patterns** (black and white patterns)
- Output of the DCT is the set of **weights** for these basis patterns (the DCT **coefficients**)
 - multiply each basis pattern by its weight and add them together
 - result is the original image block



DCT

- Most image blocks only contain a few significant coefficients (usually the lowest “frequencies”)



1 Top-left
coefficient
per block



3 top-left
coefficients



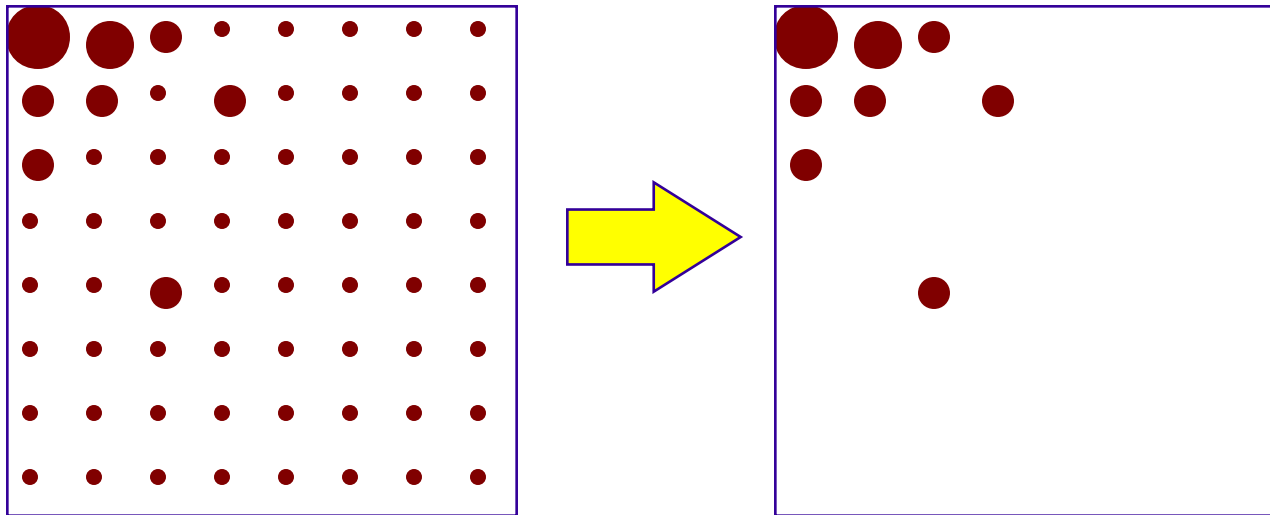
6 top-left
coefficients



All
coefficients

Quantize

- ◆ Divide each DCT coefficient by an integer, discard remainder
- ◆ Result: loss of precision
- ◆ Typically, a few non-zero coefficients are left



Run-Level Encoding

- ◆ Encode each coefficient value as a (run,level) pair
 - run = number of zeros preceding value
 - level = non-zero value
- Usually, the block data is reduced to a short sequence of (run,level) pairs.
 - This is now easy to compress using an Entropy Encoder.

Example:

Original data	14,3,4,0,0,-3,0,0,0,0,0,14,...
(Run,level)	(0,14)(0,3)(0,4)(2,-3)(5,14)...

Variable-Length Coding

- ◆ Encode each (run,level) pair using a variable-length code
- ◆ Frequently occurring groups
 - assign a short code
- ◆ Infrequently occurring groups
 - assign a long code
- Result: compressed version of image.

Image CODEC

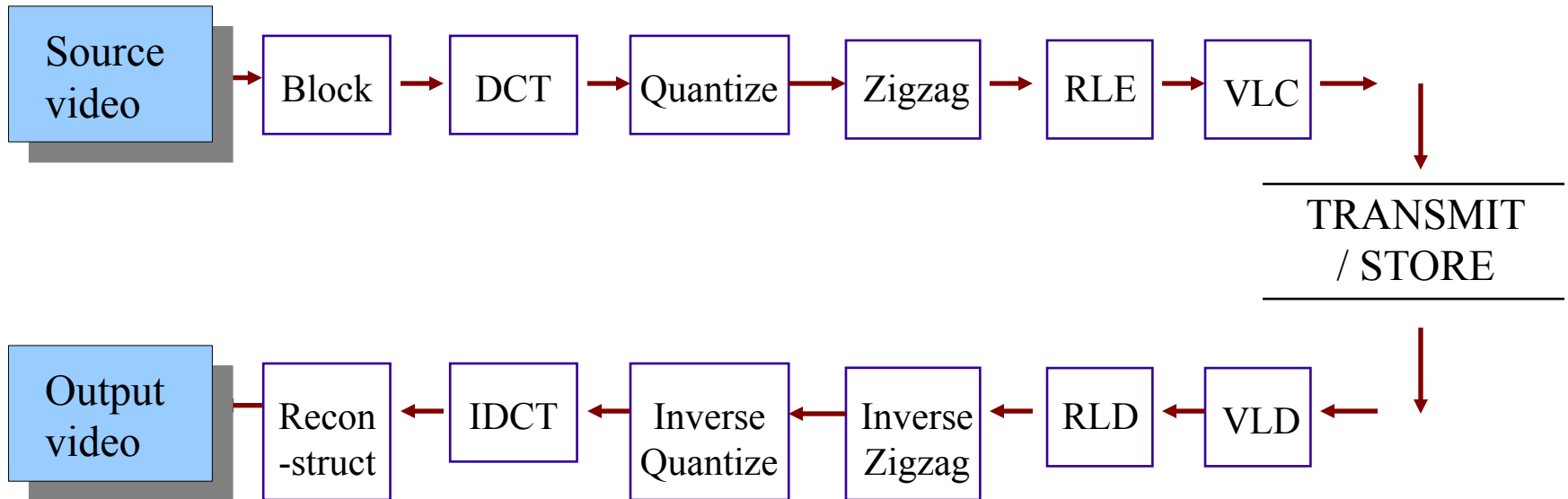


Image Decoding

- Reverse the stages to recover the image
- Information was thrown away during Quantization
 - decoded image will not be identical to the original
- In general:
 - more compression = more quality loss
- Too much compression:
 - block edges start to show (“blockiness”)
 - high-frequency patterns start to appear (“mosquito noise”)

JPEG: examples



Original



9x
compression



15x
compression

Outline

- The need for compression
- The coding model
- Image coding
- Video coding
- Standards

Video Coding

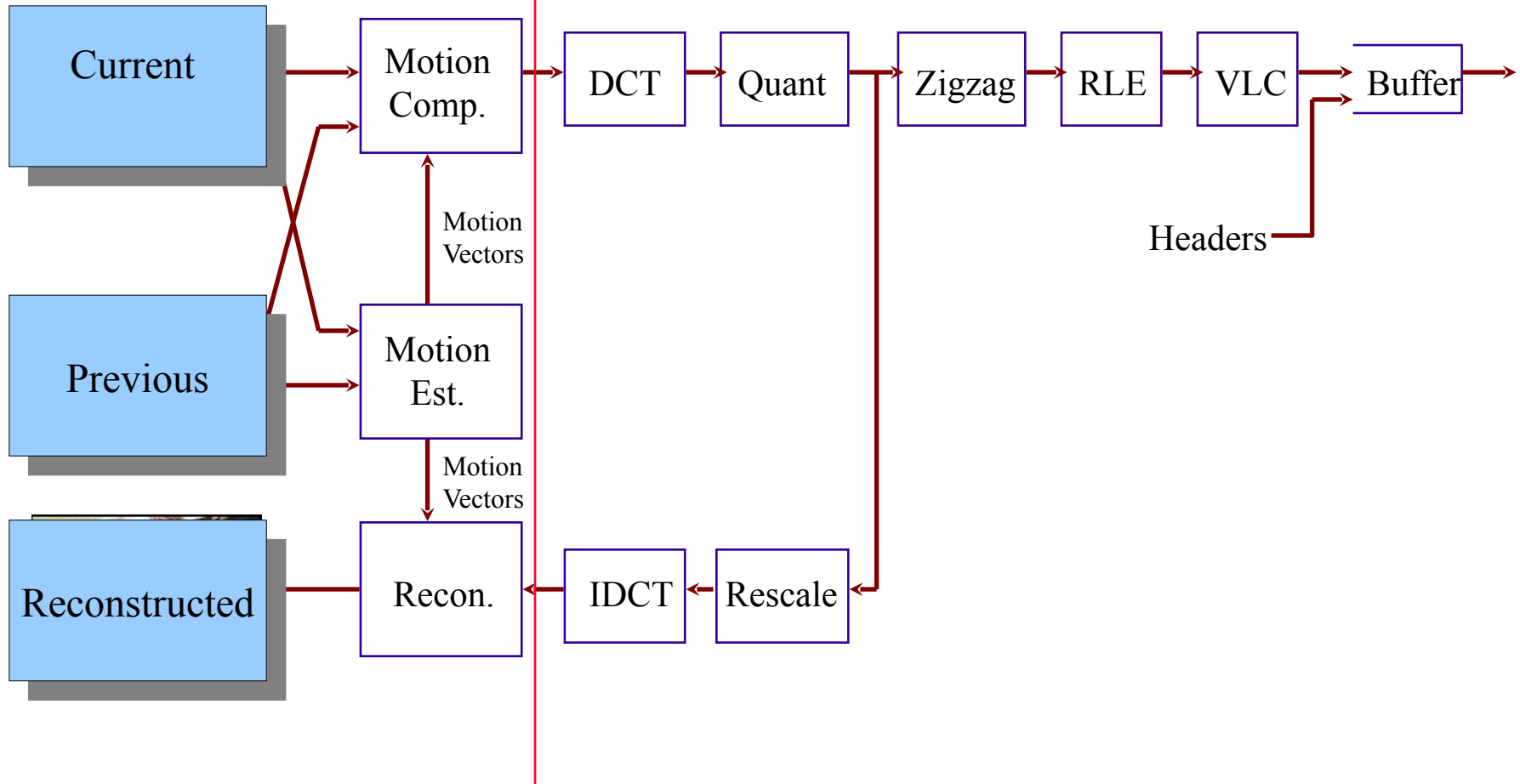
- Moving images contain significant temporal redundancy
 - successive frames are very similar
- Add an extra “motion model” at the “front end” of the image encoder.



Video Encoder

motion model

Video frames



Motion Estimation and Compensation

- The amount of data to be coded can be reduced significantly if the previous frame is **subtracted** from the current frame:



Residual Image

Difference frames

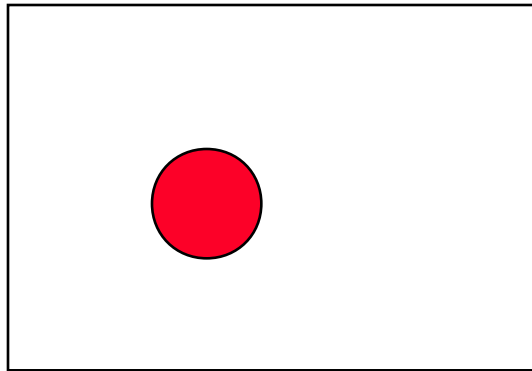


Frame n

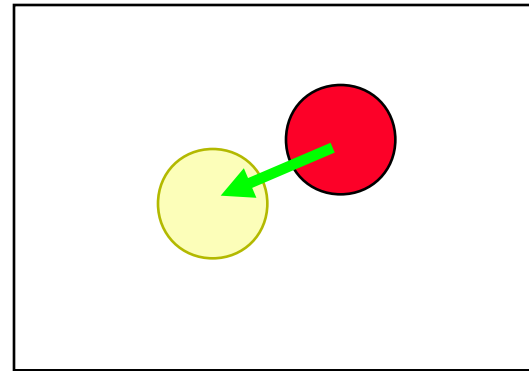
Frame n - Frame (n-1)

Motion Estimation

- Process 16x16 luminance samples at a time (“macroblock”)
- Compare with neighbouring areas in previous frame
- Find closest matching area
 - prediction reference
- Calculate offset between current macroblock and prediction reference area
 - motion vector

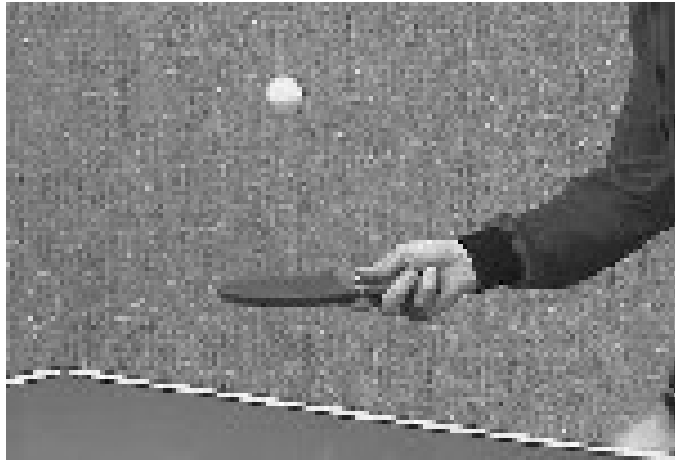


Frame 1

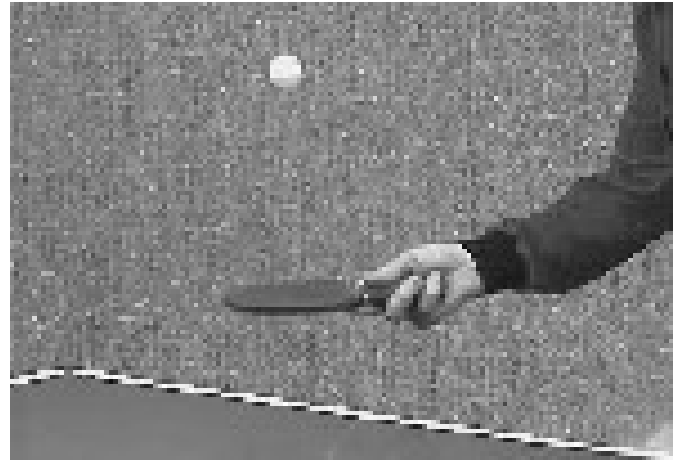


Frame 2

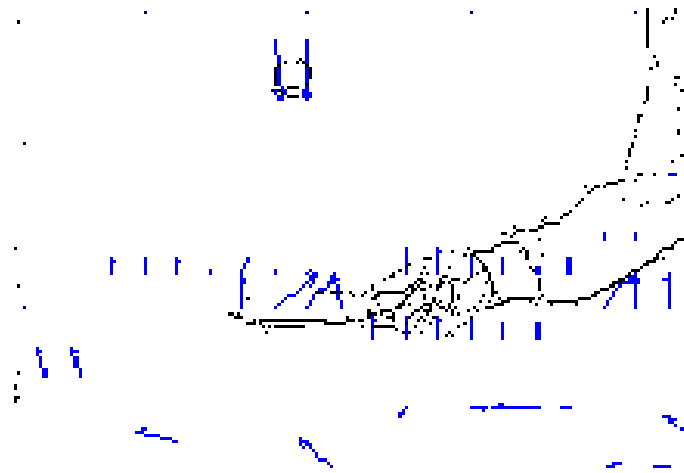
Motion Estimation



frame 1



frame 2

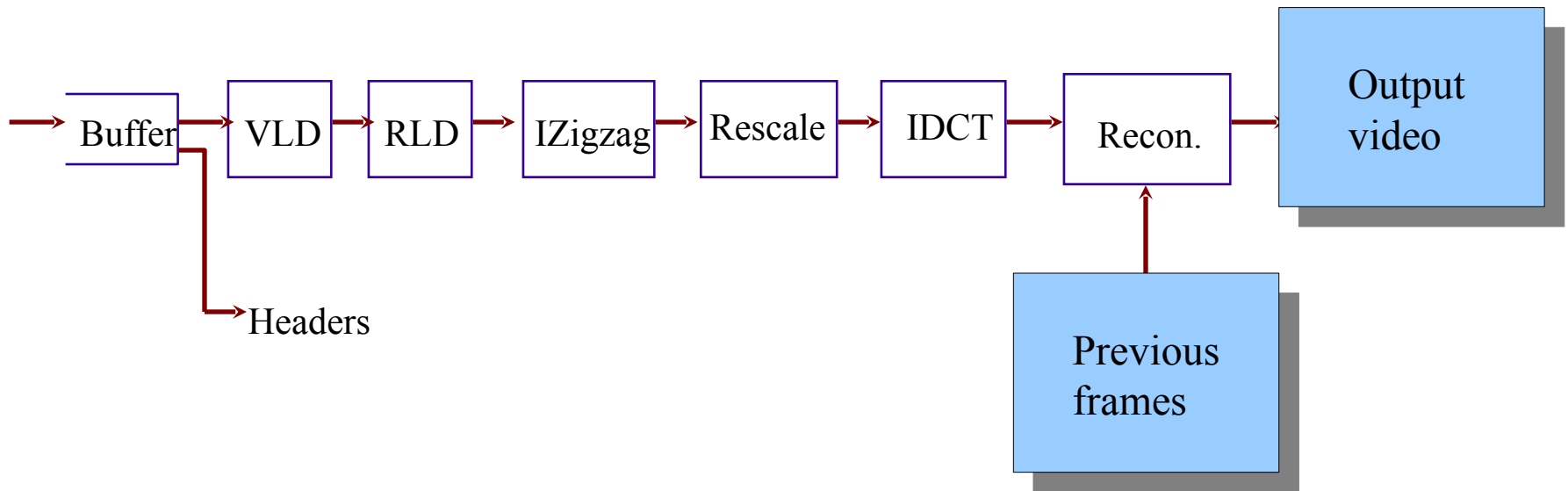


motion vectors 29

Motion Compensation

- Subtract the reference area from the current macroblock
 - difference macroblock
- Encode the difference macroblock with an image encoder
- If motion estimation was effective
 - little data left in difference macroblock
 - more efficient compression.

Video Decoder



Outline

- The need for compression
- The coding model
- Image coding
- Video coding
- Standards

Coding Standards

- JPEG
 - Joint Photographic Experts Group
 - Still image compression
- MPEG1
 - Moving Picture Experts Group
 - Video compression for CD storage / Internet
- MPEG2
 - Video compression for digital TV
- MPEG4
 - General purpose video compression
- H.261, H.263
 - Video compression for video conferencing

Video coding examples



MPEG1, 13x compression



30x compression



56x compression

References

- “Video Codec Design”, Iain Richardson, John Wiley & Sons 2002
- “Digital Video Communications”, Riley and Richardson, pub. Artech House, 1997.
- Bhaskaran, V, Konstantinides, K, “Image and video compression standards - algorithms and architectures”, Kluwer academic publishers, 1996
- Netravali, A N and Haskell, B G, “Digital Pictures: Representation, Compression and Standards”, 2nd Edition, Plenum Press, 1995.
- Ghanbari, “Video coding: an introduction to standard codecs”, IEE Press, 1999.