COMP 249 Advanced Distributed Systems
Multimedia Networking

The Video Data Type
Coding & Compression Basics

Kevin Jeffay
Department of Computer Science
University of North Carolina at Chapel Hill
jeffay@cs.unc.edu
September 7, 1999

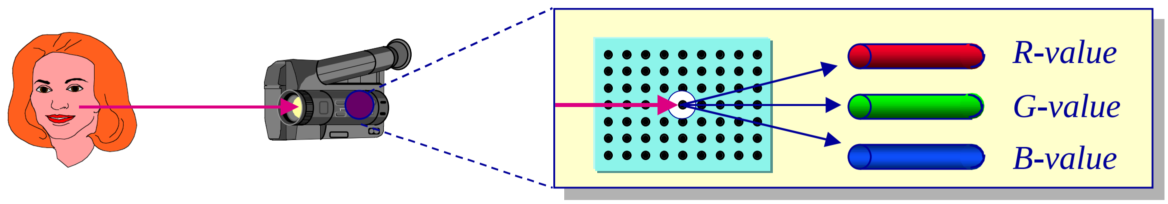
http://www.cs.unc.edu/~jeffay/courses/comp249f99

The Video Data Type
Outline

- What is video?
  - » Video components
  - » Representations of video signals
  - » Color spaces
- Digital Video
  - » Coding
- Compression basics
  - » Simple compression
  - » Interpolation-based techniques
  - » Predictive techniques
  - » Transforms
  - » Statistical techniques

Video Basics

The components of video

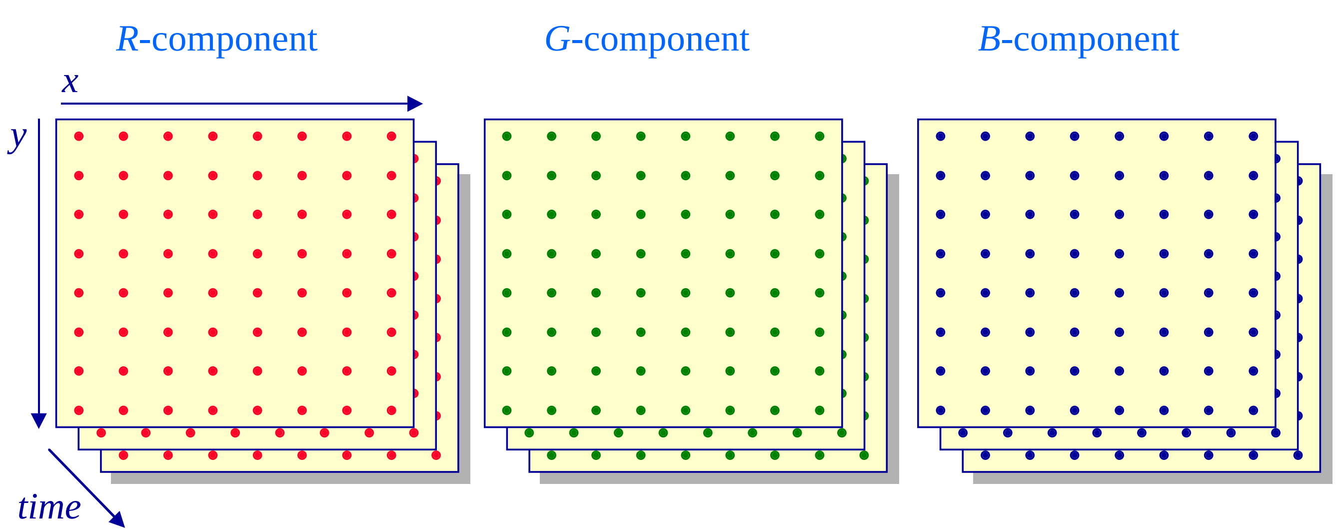


- Video deals with *absorbed* and *projected* light
  - » Cameras absorb light and monitors project light
- The primary colors in this domain are:
  - » red, green, and blue

Video Basics

The components of video transmission

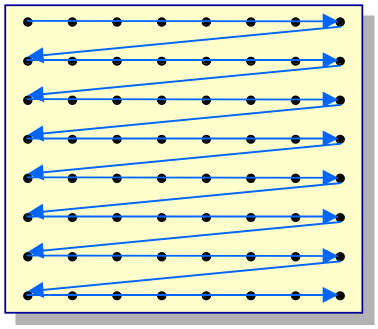
- Video is a multi-dimensional signal



Video Basics

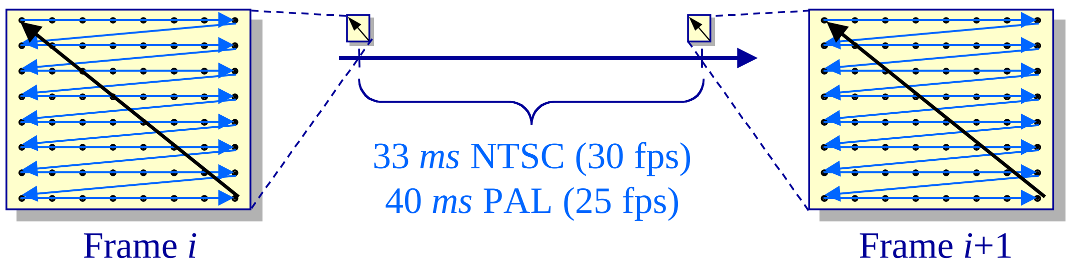
Video as a 1-dimensional signal

- Representation of a 2-dimensional image



$(R, G, B)_{11}, (R, G, B)_{12}, (R, G, B)_{13}, \ldots, (R, G, B)_{row,\, col}$

- Representation of motion (3-dimensional images)



Video Basics

Resolution

- Television broadcast standards
  - » NTSC — 525 lines
  - » PAL — 625 lines
- Computer graphics standards
  - » VGA — 640x480
  - » SVGA — 1024x768
- Multimedia standards
  - » CIF — 352x288
  - » QCIF — 176x144
- Digital video standards
  - » CCIR 601 — 720x480
  - » HDTV — 1440x1152

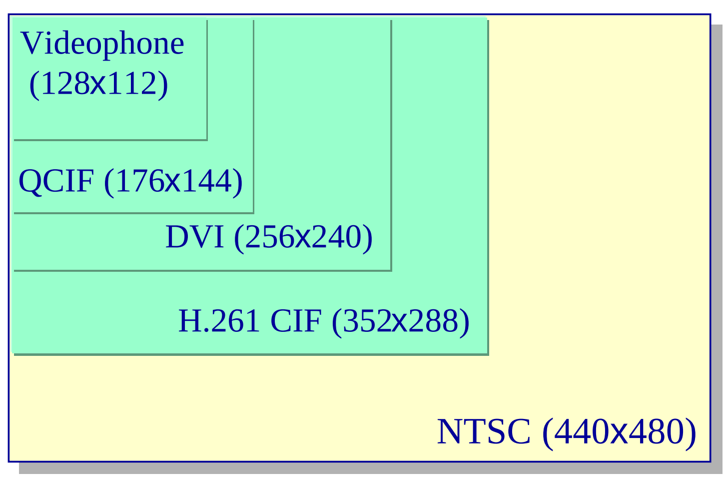


Image sizes
(in picture elements)

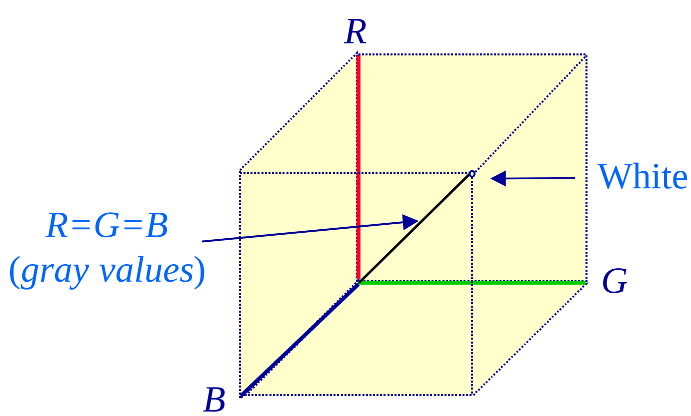
Video Basics

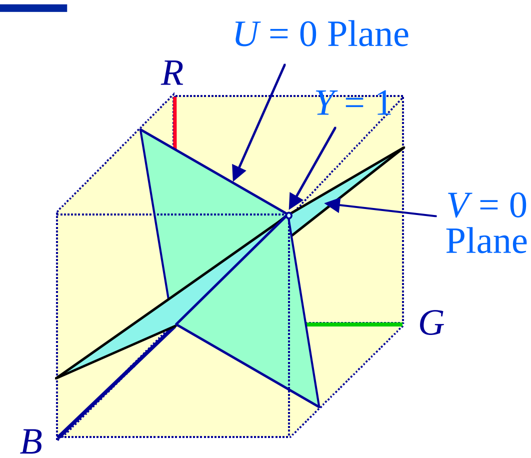
Color spaces

- *RGB* is not widely used for transmitting a signal between capture and display devices
  - » It's difficult to manage 3 separate inputs & outputs (and requires too much bandwidth)
- Composite formats are used instead
  - » Luminance ("*Y*") — the brightness of the monochrome signal
  - » Chrominance — the coloring information
  - » Chrominance is typically represented by two "color difference" signals:
    - "*U*" and "*V*" ("*hue* and *tint*") or
    - "*I*" and "*Q*" ("*saturation*" and "*color*")

Video Basics

Color spaces





- *NTSC* video
  - » $Y = 0.30R + 0.59G + 0.11B$
  - » $I = 0.60R - 0.28G - 0.32B$
  - » $Q = 0.21R - 0.52G + 0.31B$
- *PAL* video/Digital recorders
  - » $Y = 0.3R + 0.6G + 0.1B$
  - » $U = (B - Y) \times 0.493$
  - » $V = (R - Y) \times 0.877$

Video Basics
Digital video

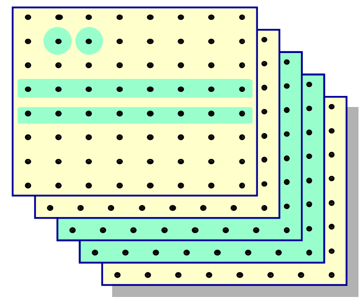
- Sample an analog representation of video (*RGB* or *YUV*) & quantize
  - » Two dimensions of video are already discretized
  - » Sample in the horizontal direction according to the resolution of the media
- 8-bits per component per sample is common
  - » 24 bits per picture element (pixel)
- Storage/transmission requirements
  - » NTSC — 440 x 480 x 30 x 24 = 152×10^6 bits/sec (19 MB/s or 24 bits/pixel (bpp))

The Video Data Type
Outline

- What is video?
  - » Video components
  - » Representations of video signals
  - » Color spaces
- Digital Video
  - » Coding
- Compression basics
  - » Simple compression
  - » Interpolation-based techniques
  - » Predictive techniques
  - » Transforms
  - » Statistical techniques

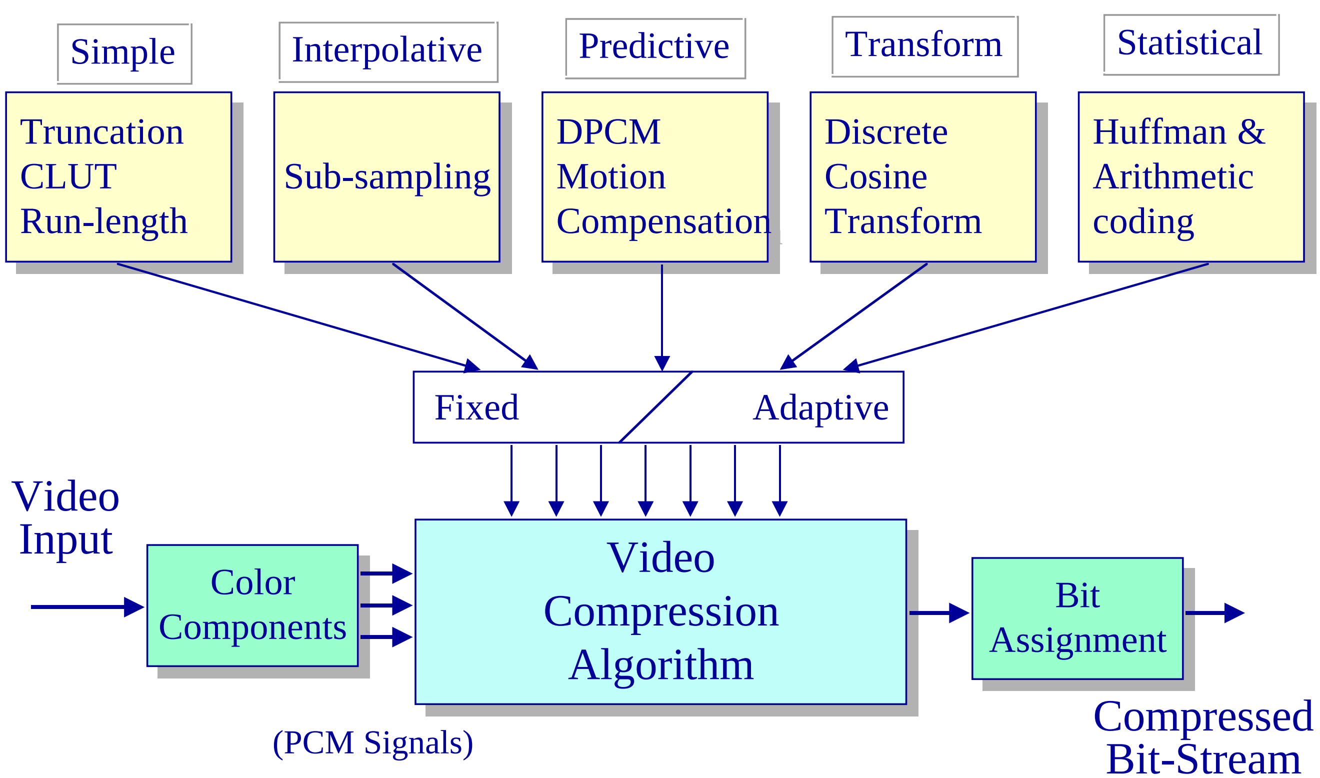
Digital Video
Compression Techniques

- Do we really need every “bit” of a video stream?
  - » Not if redundancy exists
  - » Not if we can’t perceive the effect of eliminating the bit



- Eliminating redundancy
  - » Spatial redundancy
  - » Temporal redundancy
- Eliminating imperceptible detail
  - » Coding
  - » Domain transformation

Digital Video
Compression Techniques



Adapted from Buford p.147

Video Compression
Issues

- Bandwidth requirements of resulting stream
  - » Bits per pixel (bpp)
- Image quality
- Compression/decompression speed
  - » Latency
  - » Cost
  - » Symmetry
- Robustness
  - » Tolerance of errors and loss
- Application requirements
  - » Live video
  - » Stored video

Simple Image Compression
Truncation

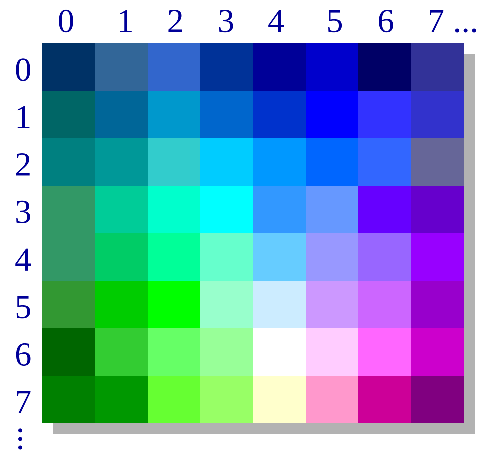
- Reducing the number of bits per pixel
  - » Throw away the least significant bits of each sample value
- Example
  - » Go from *RGB* at 8 bits/component sample (8:8:8) to 5 bits (5:5:5)
    - Go from 24 bpp to 15 bpp
    - This gives “acceptable results”
  - » Go from *YUV* at 8 bits/component sample 6:5:5 (16 bpp)
- Advantage — simple!

Simple Compression Schemes

Color-table lookup (CLUT)

- Quantize coarser in the color domain
  - » Pixel values represent indices into a color table
  - » Tables can be optimized for individual images

- Entries in color table stored at “full resolution” (*e.g.* 24 bits)

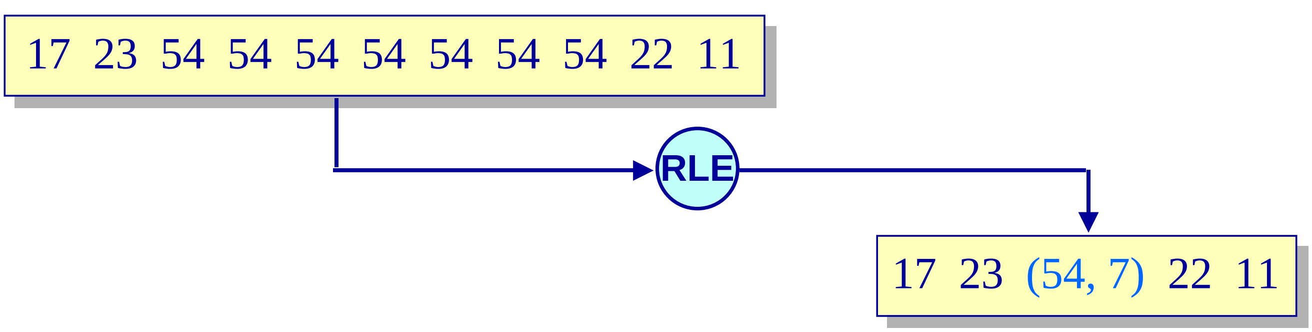


- Example:
  - » 8-bit indices (256 colors) gives

    $(440 \times 480) \times 8 + (24 \times 256) = 1.7 \times 10^6$ bits/sec

Simple Compression Schemes

Run-length encoding



- Replace sequences of pixel components with identical values with a pair (*value*, *count*)

- Works well for computer-generated images, cartoons. works less well for natural video

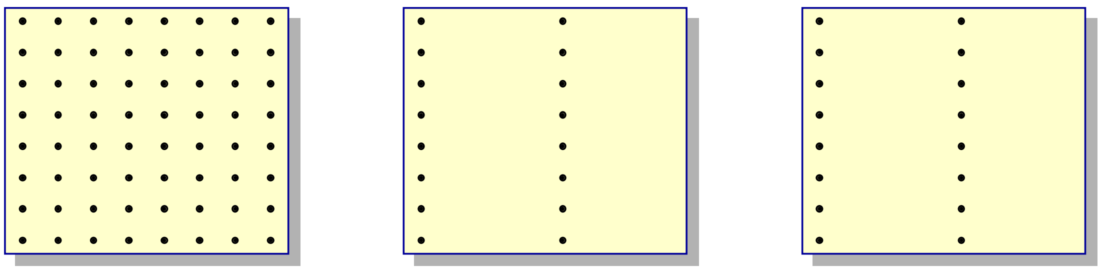
- Also works well with CLUT encoded images

  (*i.e., multiple techniques may be effectively combined*)

Interpolative Compression Schemes

Color sub-sampling

- Do not acquire chrominance component values at all sampling points
  - » Humans have poor acuity for color changes
  - » *UV* and *IQ* components were defined with this in mind
- <u>Example</u>: Color representation in digital tape recorders
  - » Subsampling by a factor of 4 horizontally is performed

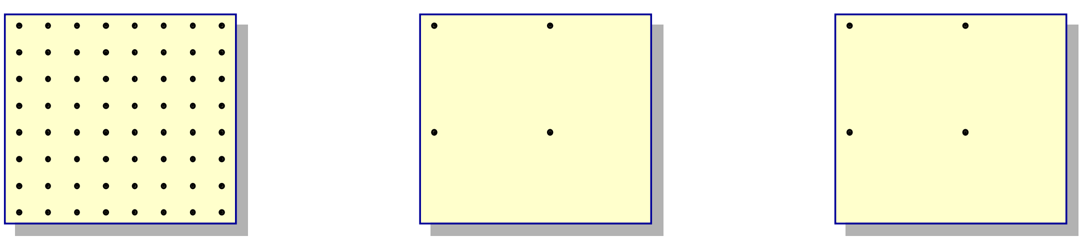


Y component *U* component *V* component

Interpolative Compression Schemes

Color sub-sampling

- Subsampling by a factor of 4 horizontally & vertically



Y component *U* component *V* component

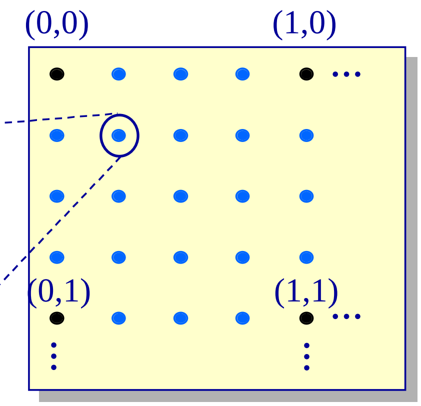
- Interpolating between samples provides “excellent” results
  - » Chrominance still sampled at 8 bpp

Interpolative Compression Schemes

Color sub-sampling

- Intermediate pixels either take on the value of nearest sampling point or their value is computed by interpolation
- Bi-linear interpolation:

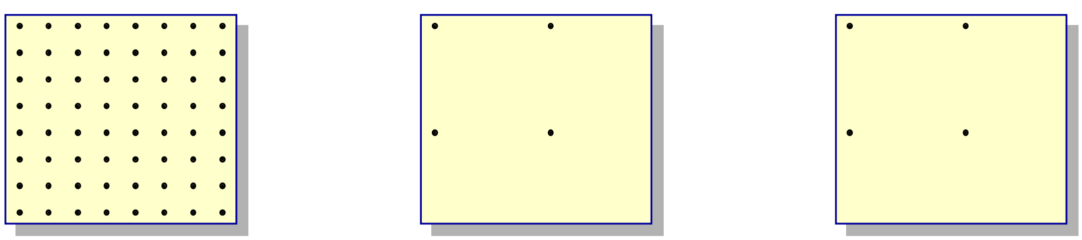
$$U(1,1) = U(0,0)\times 0.75 + U(1,0)\times 0.25 + U(0,1)\times 0.75 + U(1,1)\times 0.25$$



Sub-sampled
U or *V* component

Interpolative Compression Schemes

Color sub-sampling



Y component *U* component *V* component

- Storage/transmission requirements reduction:
  - » Within a 4x4 pixel block:

$$\text{bpp} = \frac{(8 \text{ bpp luminance})\times 16 \text{ samples} + (8 \text{ bpp chrominance})\times 2}{16}$$

$$= 9$$

  - » A 62.5% reduction overall

Predictive Compression Schemes

Exploiting spatial & temporal redundancy

- Adjacent pixels are frequently similar
  - » Do pixel-by-pixel DPCM compression
    - Leads to smearing of high-contrast edges
  - » ADPCM — a little better, a little worse
    - Introduces "edge quantization" noise
- Motion Estimation — If the future is the similar to the past, encode only the difference between frames
  - » This assumes we can store a previous frame to compare with a future one

Transform-Based Compression

Exploiting redundancy in other domains

- A simple linear transformation

2 x 2 array of pixels

A	B
C	D

1-D array of differences

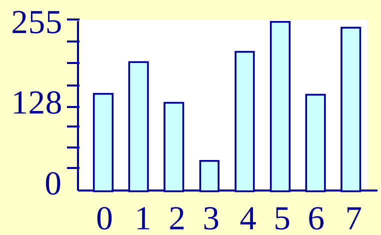
A	B–A	C–A	D–A

  - » Encode differences with less precision
- Storage savings
  - » Original array: 4 pixels x 8 bpp = 32 bits
  - » Transformed array: 8 bits + (3 pixels x 4 bpp) = 20 bits

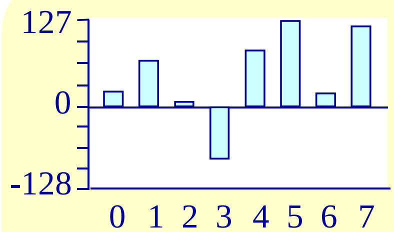
Transform-Based Compression

The Discrete Cosine Transform (DCT)

- A transformation into the frequency domain
- Example: 8 adjacent pixel values (*e.g.*, luminance)



Sample values



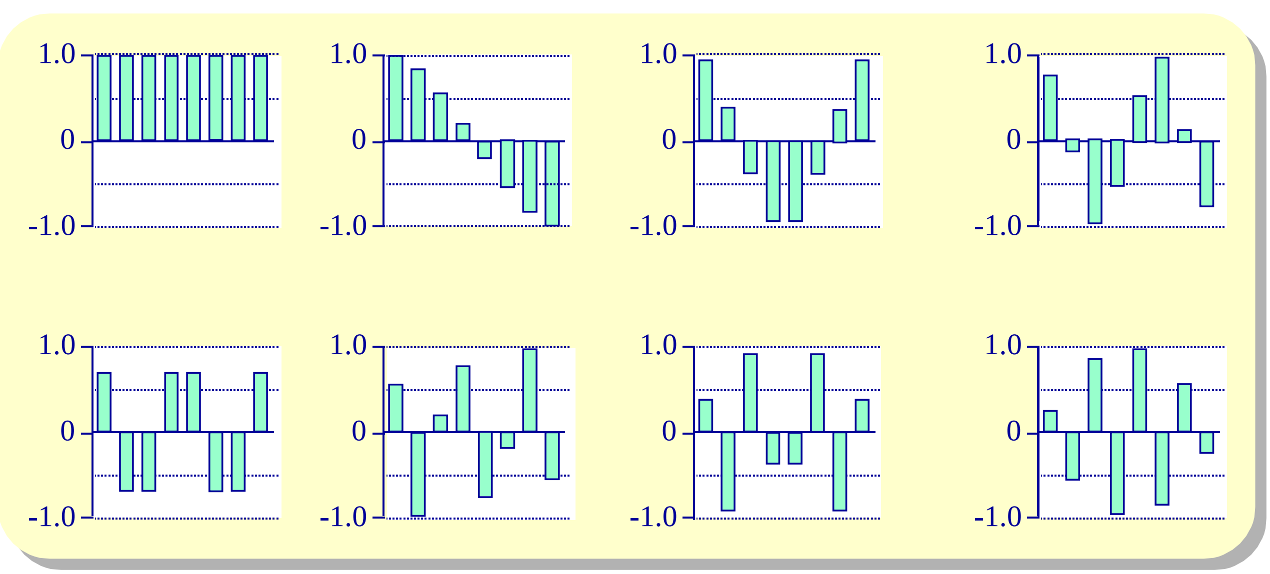
Level-shifted values

- What is the most compact way to represent this signal?

Transform-Based Compression

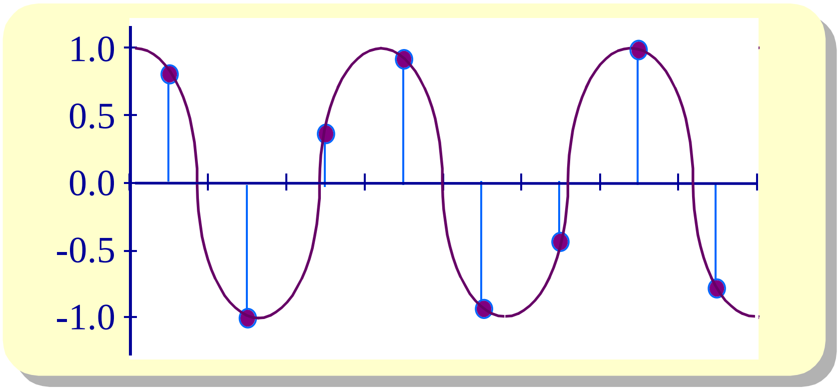
The Discrete Cosine Transform (DCT)

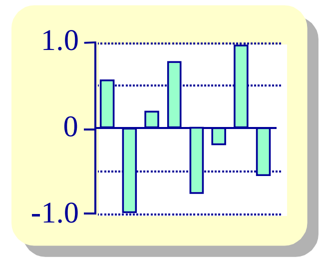
- Represent the signal in terms of a set of *cosine basis functions*



Transform-Based Compression

The Discrete Cosine Transform (DCT)



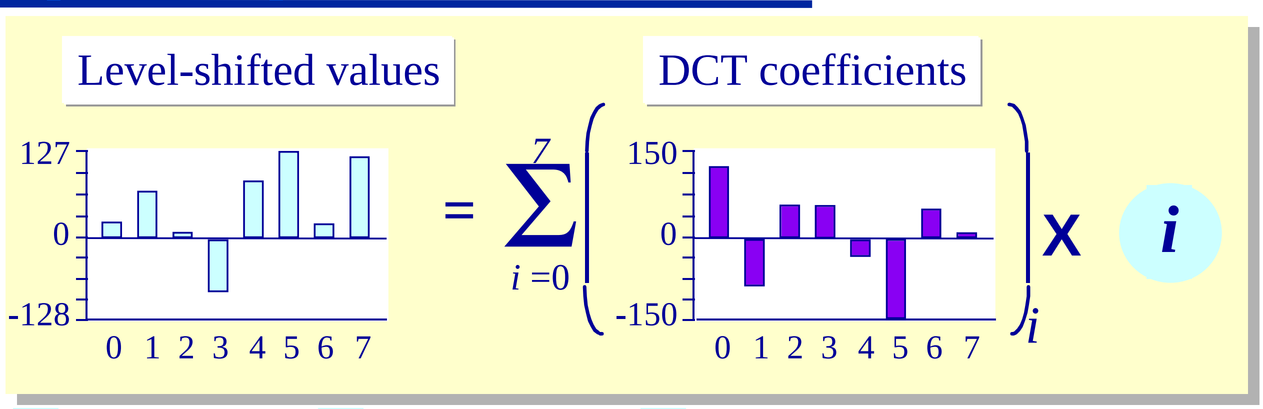


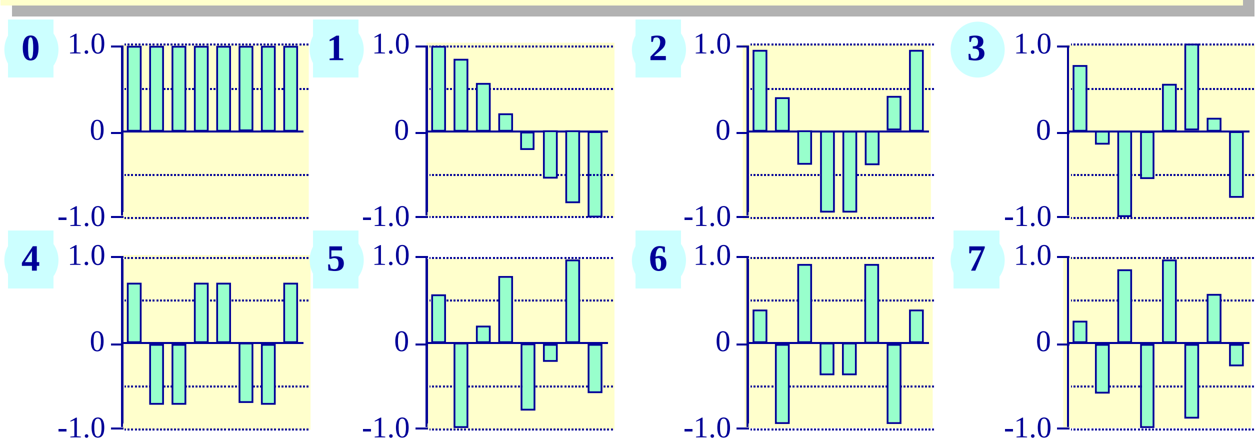
Sampled 2.5 Hz cosine wave

- The basis functions derive from sampling cosine functions of increasing frequency
  - » From 0-3.5 Hz
  - » Basis functions sampled at 8 discrete points

The Discrete Cosine Transform

Represent input as a sum of scaled basis functions





Transform-Based Compression

The Discrete Cosine Transform (DCT)

- The 1-dimensional transform:

$$F(\mu) = \frac{C(\mu)}{2} \sum_{x=1}^{7} f(x) \cos \frac{(2x+1)\mu\pi}{16}$$

  - » $F(\mu)$ is the DCT coefficient for $\mu = 0..7$
  - » $f(x)$ is the x^{th} input sample fot $x = 0..7$
  - » $C(\mu)$ is a constant (equal to $2^{-0.5}$ if $\mu = 0$ and 1 otherwise)

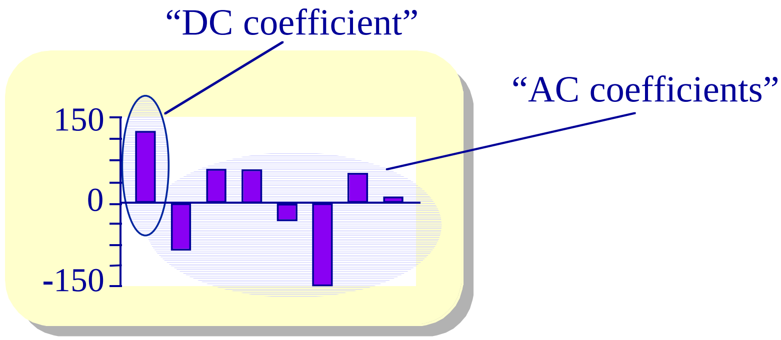
- The 2-dimensional (spatial) transform:

$$F(\mu,\nu) = \frac{C(\mu)}{2}\frac{C(\nu)}{2} \sum_{y=1}^{7} \sum_{x=1}^{7} f(x,y) \cos \frac{(2x+1)\mu\pi}{16} \cos \frac{(2y+1)\nu\pi}{16}$$

Transform-Based Compression

The Discrete Cosine Transform (DCT)

- DCT coefficients encode the spatial frequency of the input signal
  - » DC coefficient — zero spatial frequency (the "average" sample value)
  - » AC coefficients — higher spatial frequencies



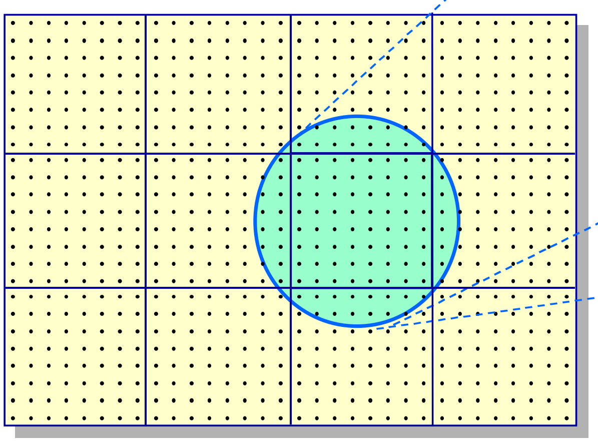
- Claim: Higher frequency coefficients will be zero and can be ignored

Transform-Based Compression

The two-dimensional DCT

- Apply the DCT in *x* and *y* dimensions simultaneously to 8x8 pixel blocks
  - » Code coefficients individually with fewer bits

Video Frame



172	-18	15	-8	23	-9	-14	19
21	-34	24	-8	-10	11	14	7
-9	-8	-4	6	-5	4	3	-1
-10	6	-5	4	-4	4	2	1
-8	-2	-3	5	-3	3	4	6
4	-2	-4	6	-4	4	2	-1
4	-3	-4	5	6	3	1	1
0	-8	-4	3	2	1	4	0

DCT Coefficients

Statistical Compression

Huffman coding

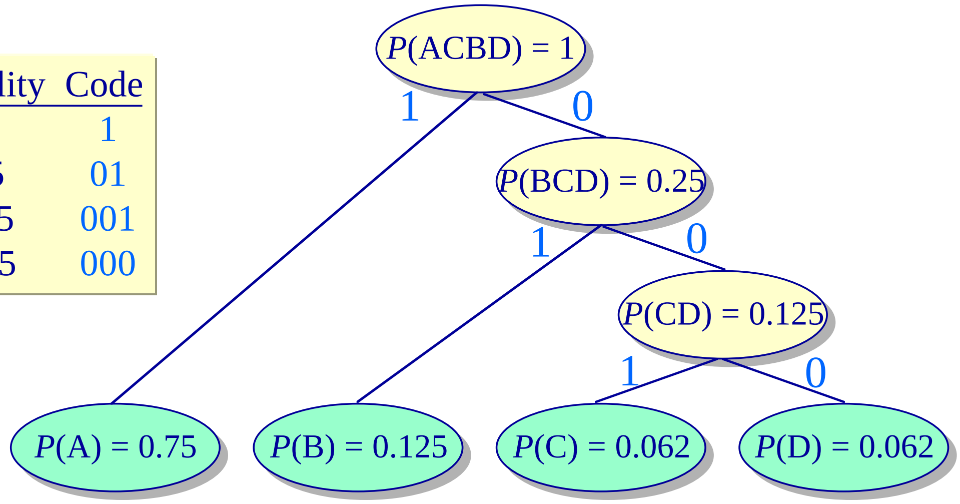
- Exploit the fact that not all sample values are equally likely
  - » Samples values are non-uniformly distributed
  - » Encode "common" values with fewer bits and less common values with more bits

- Process each image to determine the statistical distribution of sample values
  - » Generate a *codebook* — a table used by the decoder to interpret variable length codes
  - » Codebook becomes part of the compressed image

Statistical Compression

Huffman coding

Symbol	Probability	Code
A	0.75	1
B	0.125	01
C	0.0625	001
D	0.0625	000



- Order all possible sample values in a binary tree by combining the least likely samples into a sub-tree
- Label the branches of the tree with 1’s and 0’s
  - » Huffman code is the sequence of 1’s and 0’s on the path from the root to the leaf node for the symbol

Video Compression Standards

Kevin Jeffay
Department of Computer Science
University of North Carolina at Chapel Hill
jeffay@cs.unc.edu
September 14, 1999

http://www.cs.unc.edu/~jeffay/courses/comp249f99

The Video Data Type
Compression Standards

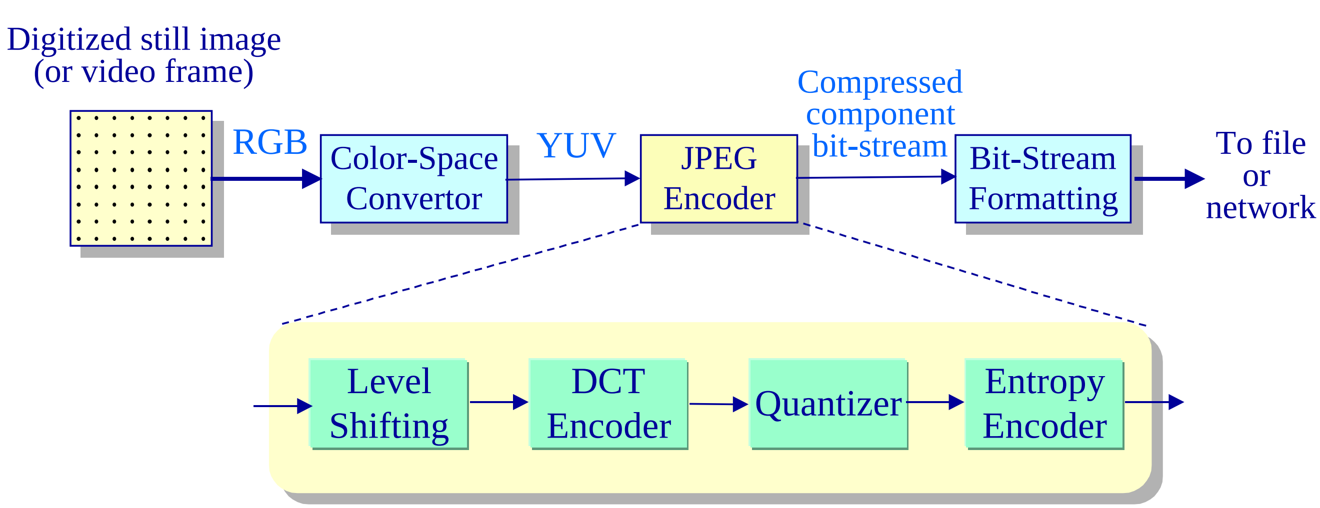
- Basic compression techniques
  - » Truncation, CLUT, run-length coding
  - » sub-sampling & interpolation
  - » DPCM
  - » DCT
  - » Huffman coding
- Common algorithms
  - » JPEG/MJPEG
  - » H.261/H.263
  - » MPEG-1,-2

Compression Algorithms
JPEG

- A still image ("continuous tone") compression standard
  - » DCT-based
- 4 Modes of compression
  - » *sequential* — image components coded in order scanned
    - ❖ Baseline — "default compression"
  - » *progressive* — image coded in multiple passes so partial images can be displayed during decoding
  - » *lossless* — guaranteed no loss
  - » *hierarchical* — image encoded at multiple resolutions
- Typical results
  - » 24:1 compression (1 bpp)

JPEG Compression
Encoder architecture — sequential mode

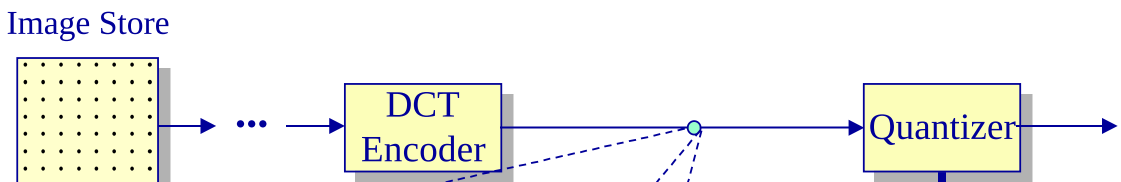
- Inputs are 8 or 12-bit samples
  - » baseline = 8-bit samples
- Image components are compressed separately
  - » DCT operates on 8x8 pixel blocks



JPEG Compression

Quantization

- DCT coefficient quantization is the key to compression
  - » Quantize according to the visual important of each coefficient
- The application specifies a *quantization table*
  - » A table of step-sizes from 1-255
  - » Default tables specified for the baseline coder



DCT coefficients

172	-18	15	-8	23	-9	-14	19
21	-34	24	-8	-10	11	14	7
-9	-8	-4	6	-5	4	3	-1
-10	6	-5	4	-4	4	2	1
-8	-2	-3	5	-3	3	4	6
4	-2	-4	6	-4	4	2	-1
4	-3	-4	5	6	3	1	1
0	-8	-4	3	2	1	4	0

Step-size table

4	7	10	13	16	19	22	25
7	10	13	16	19	22	25	28
10	13	16	19	22	25	28	31
13	16	19	22	25	28	31	34
16	19	22	25	28	31	34	37
19	22	25	28	31	34	37	40
22	25	28	31	34	37	40	43
25	28	31	34	37	40	43	46

JPEG Compression

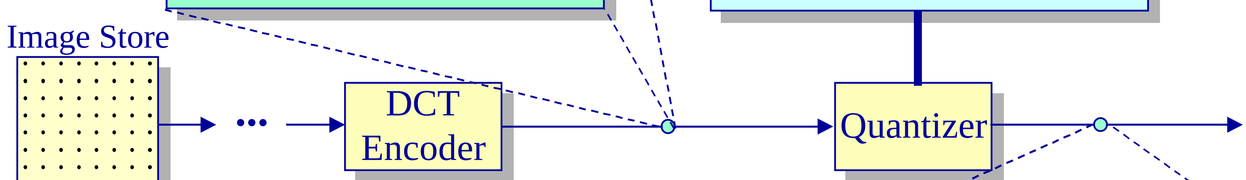
Quantization example

DCT coefficients

172	-18	15	-8	23	-9	-14	19
21	-34	24	-8	-10	11	14	7
-9	-8	-4	6	-5	4	3	-1
-10	6	-5	4	-4	4	2	1
-8	-2	-3	5	-3	3	4	6
4	-2	-4	6	-4	4	2	-1
4	-3	-4	5	6	3	1	1
0	-8	-4	3	2	1	4	0

Step-size table

4	7	10	13	16	19	22	25
7	10	13	16	19	22	25	28
10	13	16	19	22	25	28	31
13	16	19	22	25	28	31	34
16	19	22	25	28	31	34	37
19	22	25	28	31	34	37	40
22	25	28	31	34	37	40	43
25	28	31	34	37	40	43	46

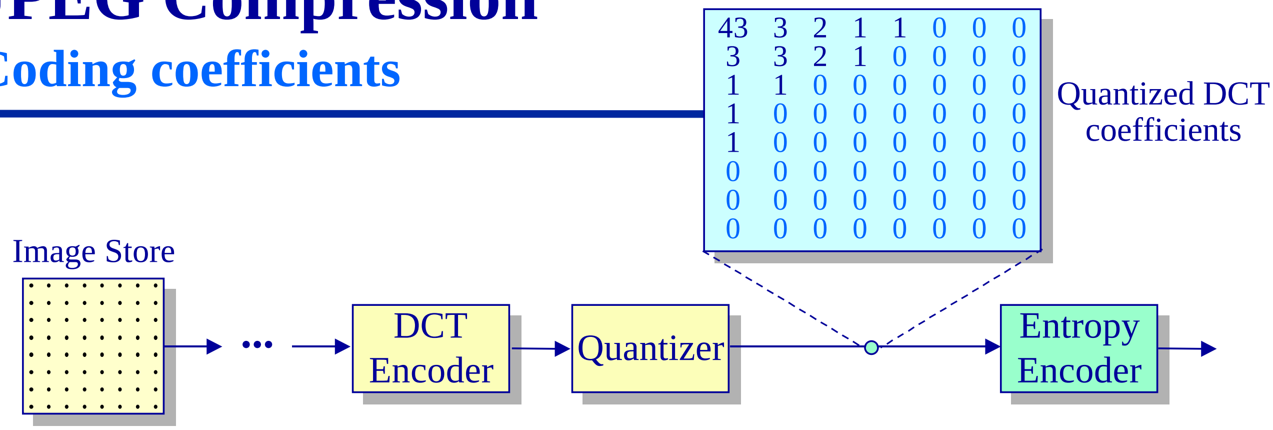


43	3	2	1	1	0	0	0
3	3	2	1	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

- Step-size table can be scaled to control degree of compression
  - » Scaling factor called the "*q-factor*"

JPEG Compression

Coding coefficients

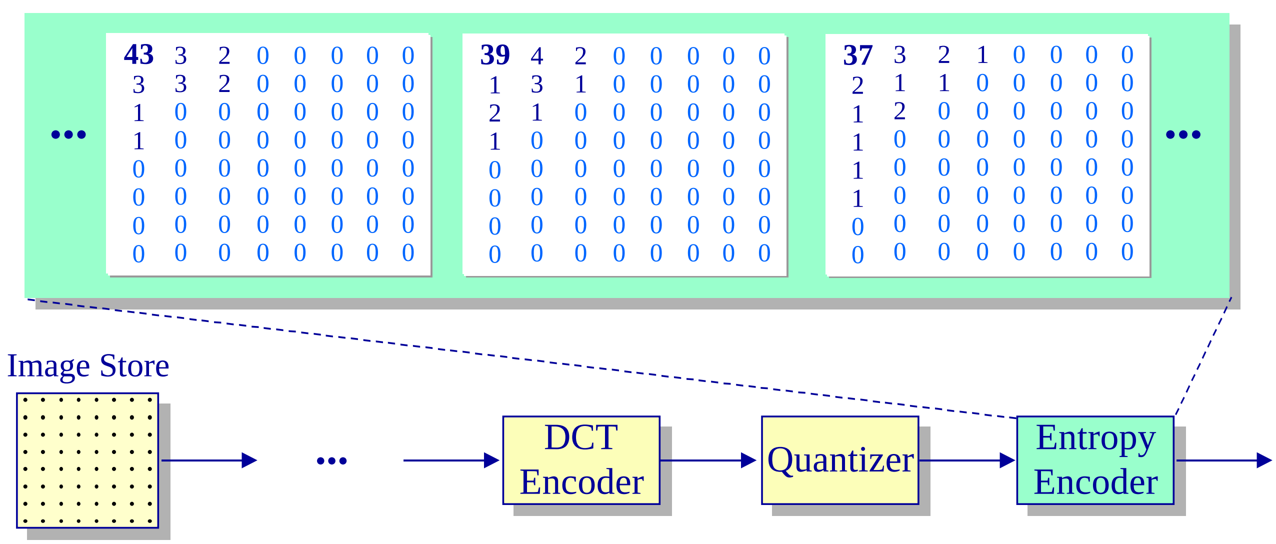


- DC coefficients difference-coded
  - » DC coefficients from adjacent 8x8 blocks strongly correlated
- AC coefficients run-length and Huffman coded

JPEG Compression

Coding DC coefficients

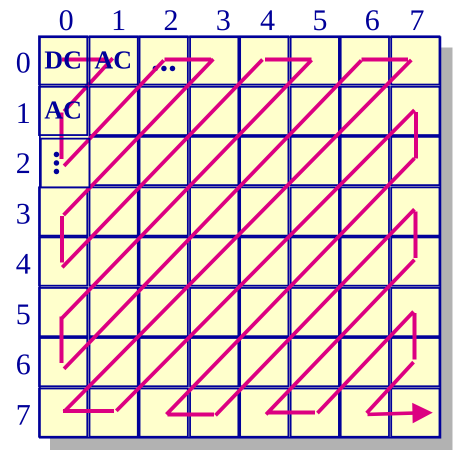
... (43-39), *ac*, *ac*, *ac*, ..., (39-37), *ac*, *ac*, *ac*, ...



- DC coefficients DPCM coded and recoded using a variable length entropy (Huffman) code

JPEG Compression
Coding AC coefficients

- "Zig-zag" order the AC coefficients to increase effectiveness of run-length coding
  - » DC,
    AC_{01}, AC_{10},
    AC_{02}, AC_{11}, AC_{20},
    AC_{03}, AC_{12}, AC_{21}, AC_{30},
    ...



- Run-length code the stream
  - » Each non-zero coefficient encoded as a pair:

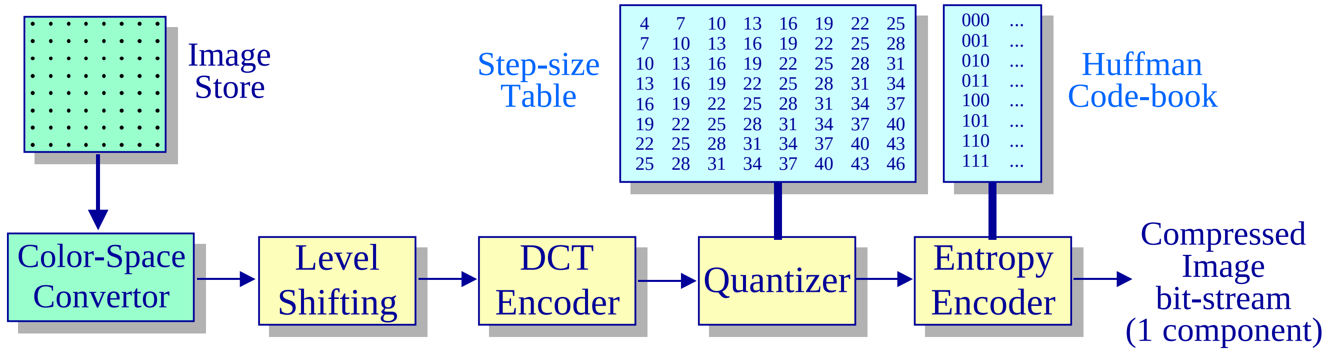
    (*run-length of preceding zero coefficients*, *amplitude of non-zero coefficient*)

JPEG Compression
Coding AC coefficients

- Each coefficient encoded as a variable length "pair"
  - » ([*run length*, *size*], *amplitude*)
- First element coded using a variable-length Huffman code
  - » A coding table ("code book") must be provided
    - Can be generated on-the-fly with an additional pass over the coefficients
    - Up to four code books per image may be specified
    - The codebook becomes part of the coded bit-stream
- Second element coded as a variable length integer
  - » whose length is specified in the previous "symbol"

Sequential JPEG Compression Summary

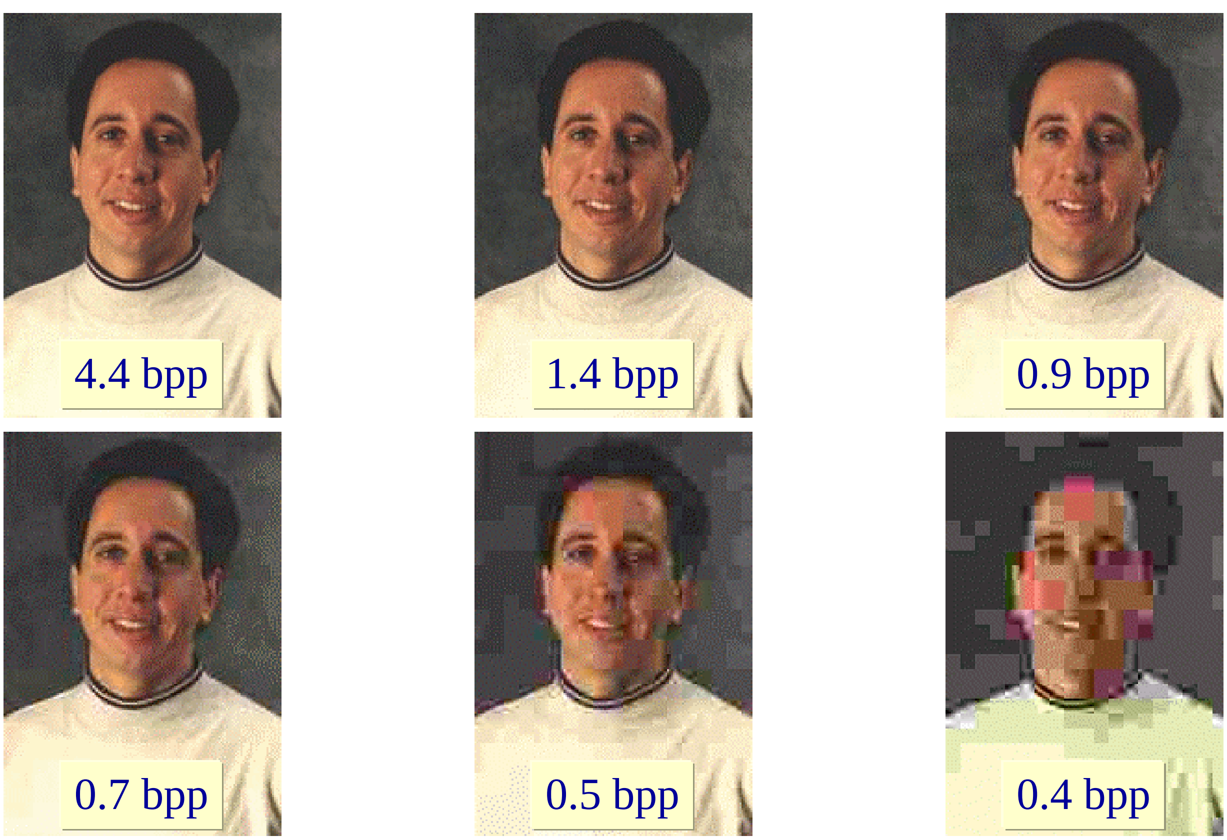
Complete compression pipeline



- Compression comes from:
  - » Chrominance subsampling
  - » DCT coefficient quantization
  - » Difference coding DC coefficients
  - » Statistical & run-length coding of AC coefficients
- Qualitative results:
  - » 0.25 - 0.5 bpp — ok for some applications
  - » 0.5 - 0.75 bpp — ok for many
  - » 0.75 - 1.5 bpp — excellent
  - » 1.5 - 2.0 — indistinguishable

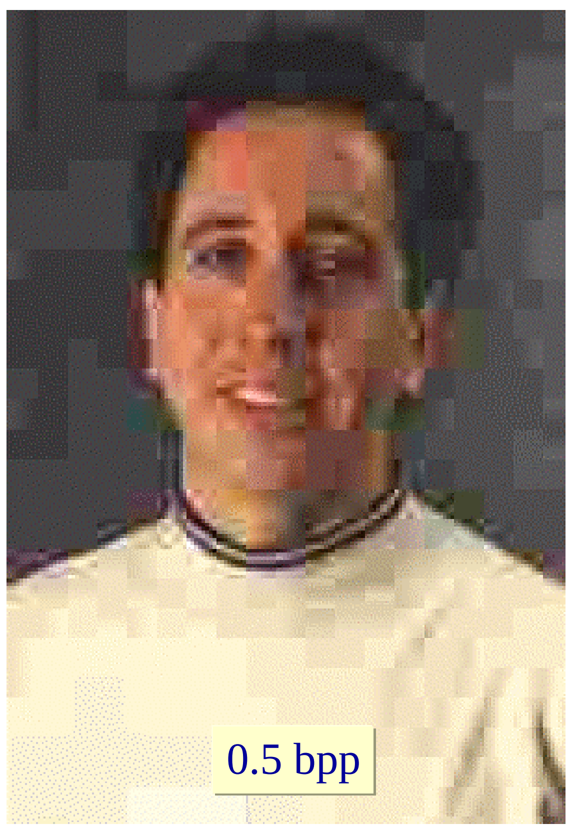
JPEG Compression

Examples of quality *v.* bpp



JPEG Compression
Examples of quality *v.* bpp

JPEG Compression
Examples of quality *v.* bpp



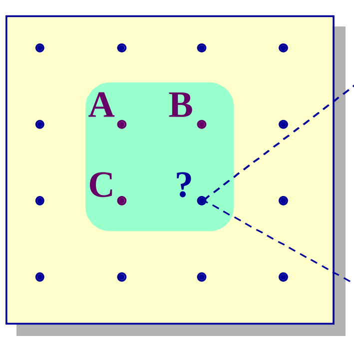
JPEG Compression

Other modes of operation

- Lossy compression modes
  - *sequential* — image components coded in order scanned
    - Default mode
  - *progressive* — image coded in multiple passes so partial images can be displayed during decoding
    - Useful for transmission of images over slow communications links
  - *hierarchical* — image encoded at multiple resolutions
    - Useful for images that will be displayed on heterogeneous displays
- Lossless mode
  - Guaranteed lossless
  - Uses DPCM encoding rather than DCT

JPEG Compression Modes

Loseless mode operation

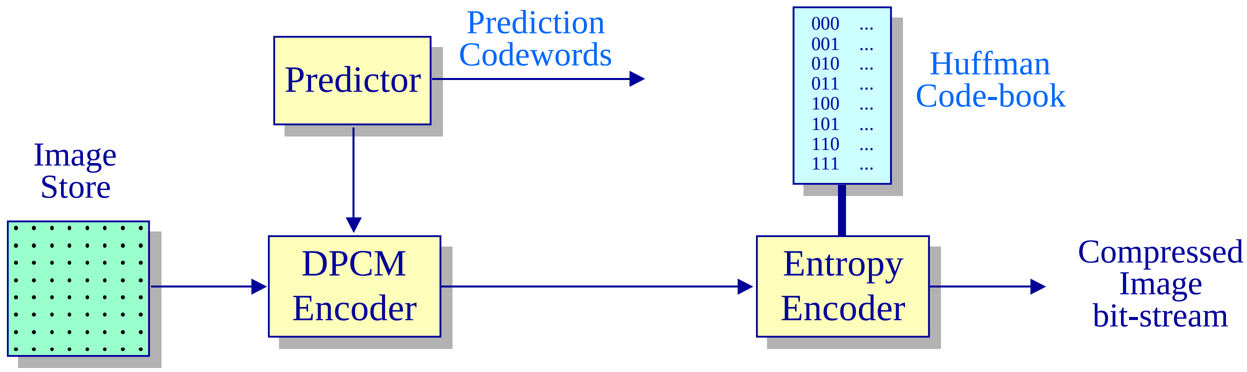


Predictor	Prediction Code
0	0
C	1
B	2
A	3
C + B – A	4
C + (B – A)/2	5
B + (C – A)/2	6
(C + B)/2	7

- Uses prediction instead of the DCT
  - Each pixel’s value is expressed as a function of neighboring pixels
  - A code word identifies the predictor being used

JPEG Compression Modes

Loseless mode operation

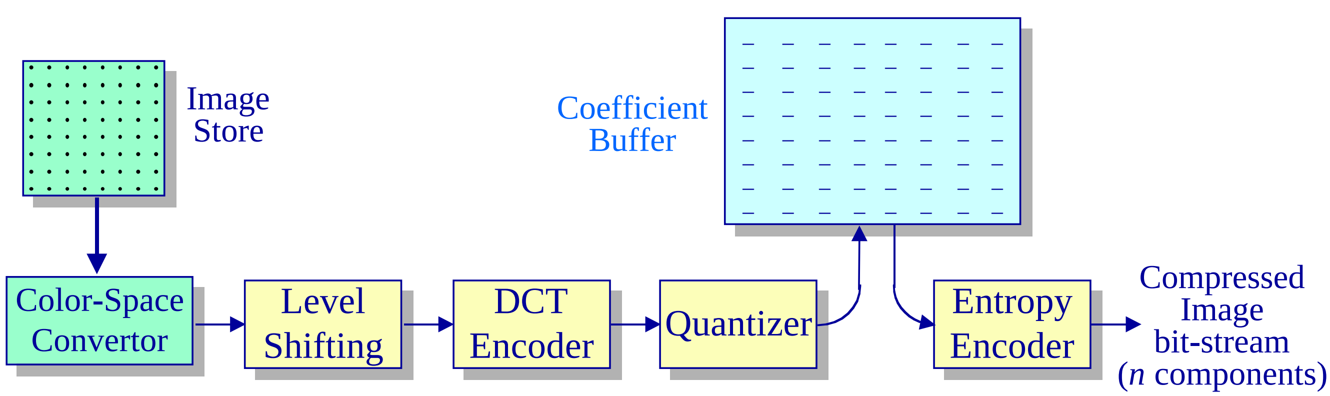


- Predicted samples are DPCM encoded
- Differences are entropy coded as before
- Achieves approximately 2:1 compression

JPEG Compression Modes

Progressive mode operation

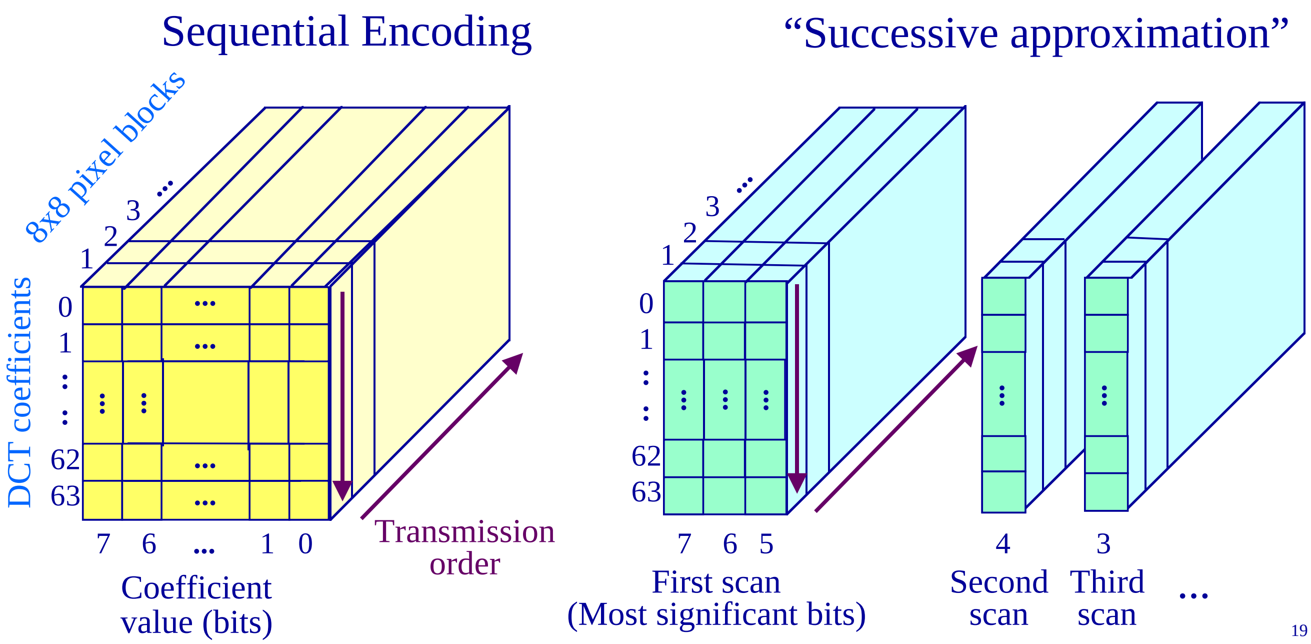
- Encode the image in scans to enable the display of a series of progressively refined images
  - » Requires an image-sized coefficient buffer between quantizer & entropy coder
  - » Scans of image components are also interleaved in bit-stream



JPEG Compression Modes

Progressive mode operation

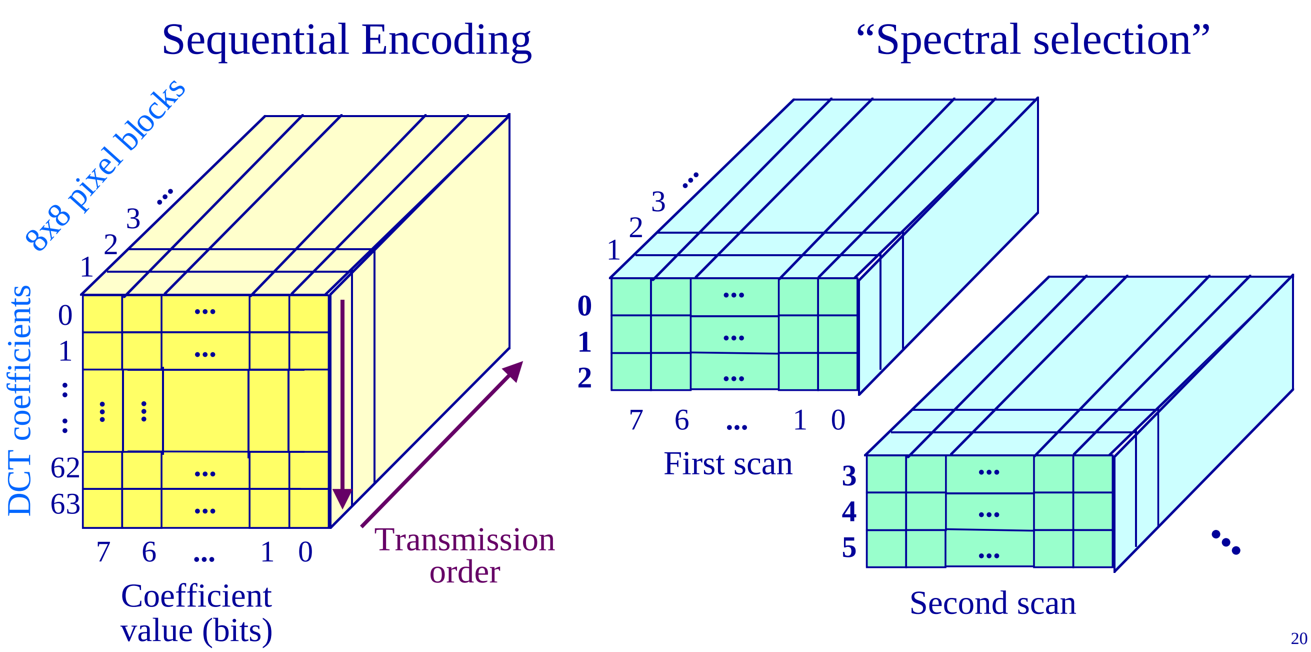
- Scan the coefficient buffer in multiple passes
  - » Transmit portions of each coefficient



JPEG Compression Modes

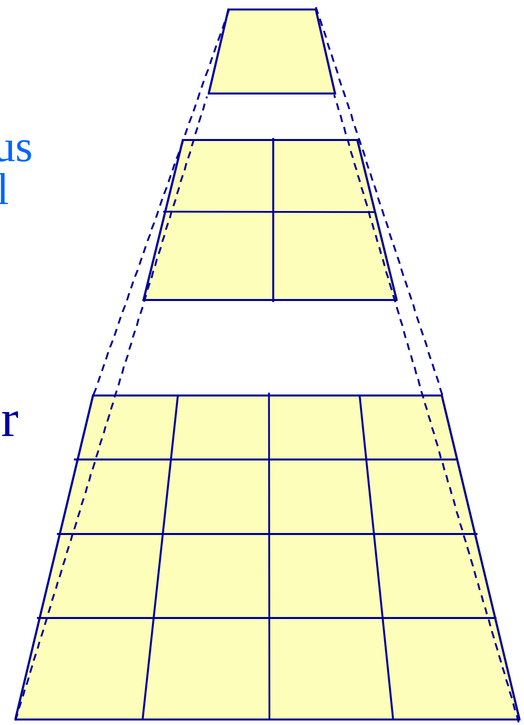
Progressive mode operation

- Scan the coefficient buffer in multiple passes
  - » Transmit portions of each coefficient



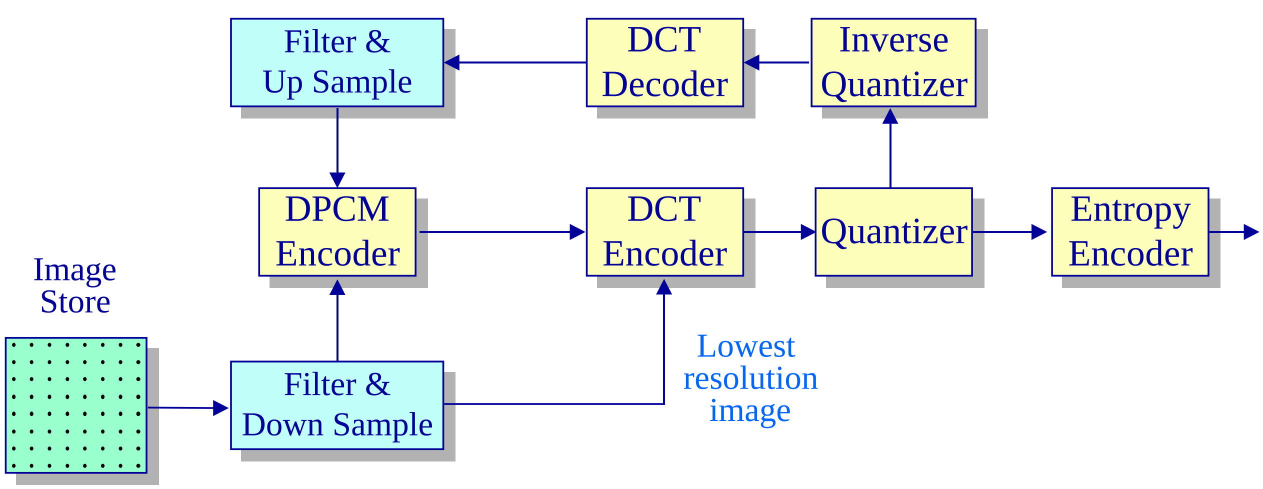
JPEG Compression Mode
Hierarchical mode operation

- Encode the image at multiple resolutions
  - » Each image differs from the previous by a factor of 2 in either the vertical or horizontal dimension
  - » Images created by filtering and subsampling
- Each resolution encoded by either the sequential or progressive algorithm



JPEG Compression Mode
Hierarchical mode operation

- Start with the lowest desired resolution & iteratively encode until the full image resolution has been coded
  - » Each iteration encodes an image with a factor of 2 higher resolution in one dimension



Motion JPEG
Applying JPEG to moving images

- Video can be (trivially) encoded as a sequence of stills
  - » This practice is routine in the digital video editing world
- The issue is how to encode and transmit "side information"
  - » Quantization tables, Huffman code-book may/may not change between frames

The Video Data Type
Compression Standards

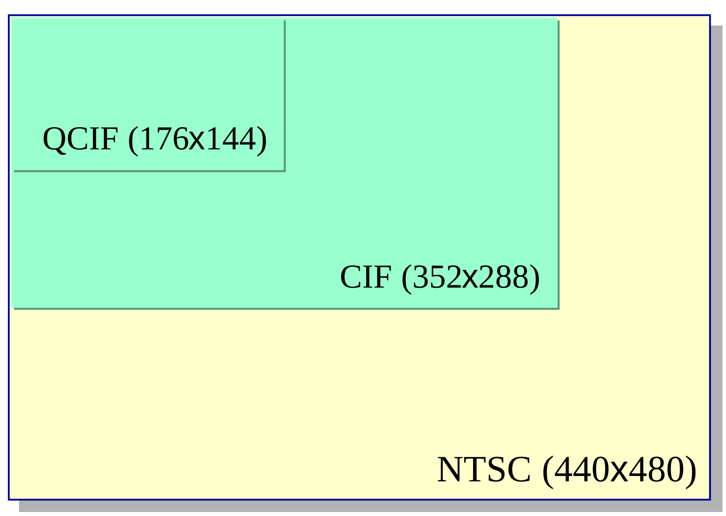
- Basic compression techniques
  - » Truncation, CLUT, run-length coding
  - » sub-sampling & interpolation
  - » DPCM
  - » DCT
  - » Huffman coding
- Common algorithms
  - » JPEG/MJPEG
  - » H.261/H.263
  - » MPEG-1,-2

Compression Algorithms
H.261 (*p*x64)

- A telecommunications (ITU) standard for audio & video transmission over digital phone lines (ISDN)
- H.261 primarily intended for interactive video applications
  - » Design of the standard driven by a 150 ms maximum encoding/decoding delay goal
- A scalable coding architecture capable of generating bit streams from 64 kbps (“1x64”) to 1,920 kbps (“30x64”) in 64 kbps increments
  - » p = 1, 2 produces a low res “videophone” (Common use is for ISDN BRI — 112 kbps video, 16 kbps audio)
  - » $p \geq 6$ produces an acceptable videoconference and allows multipoint communication

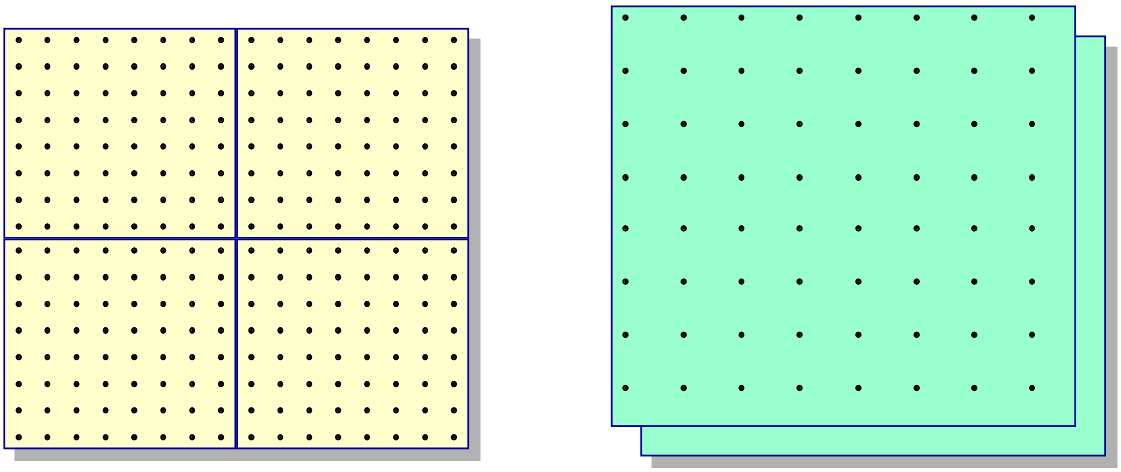
H.261
Video formats

- Inputs
  - » 525 or 625 line composite video
  - » 8 bits/sample
  - » 30 frames/second
- Color space
  - » *Y, Cr, Cb*
- Outputs
  - » CIF or QCIF
  - » 30, 15, 10, or 7.5 frames/second



H.261
Video frame representation

- Chrominance components are subsampled 2:1 horizontally & vertically
- Each video frame is subdivided into 16x16 *macroblocks*

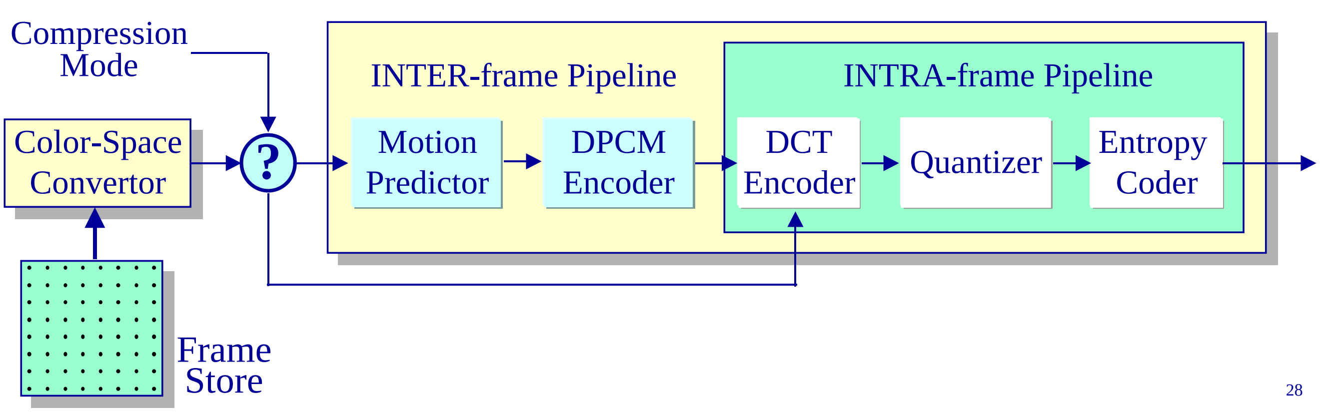


Y component: 4 8x8 blocks

Cr component: 1 8x8 block
(same for *Cb*)

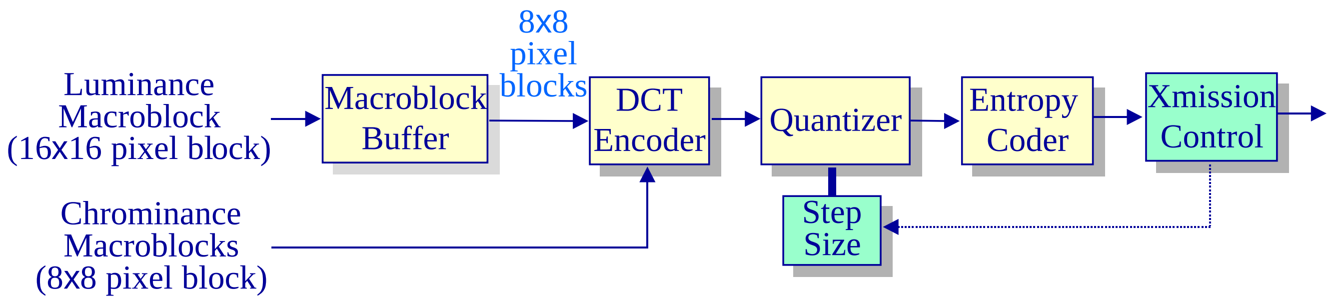
H.261
Video compression pipeline

- Two compression modes, selectable on a frame-by-frame basis
  - » INTRA-frame mode — DCT-based compression *á la* JPEG
    - video is treated as a sequence of stills
  - » INTER-frame mode — Incorporates motion estimation & DPCM prediction
    - temporal redundancy is eliminated to further improve compression



H.261 Video Compression

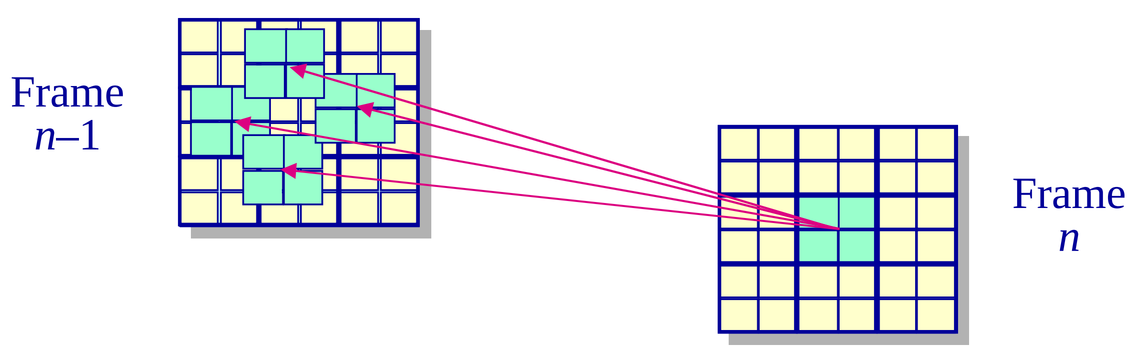
INTRA-frame mode



- Compression is similar to JPEG
  - » DCT encoding
  - » linear quantization
  - » entropy coding
- Quantization is uniform across all AC coefficients
  - » But is adaptive and driven by the space remaining in a transmission buffer

H.261 INTER-Frame Mode

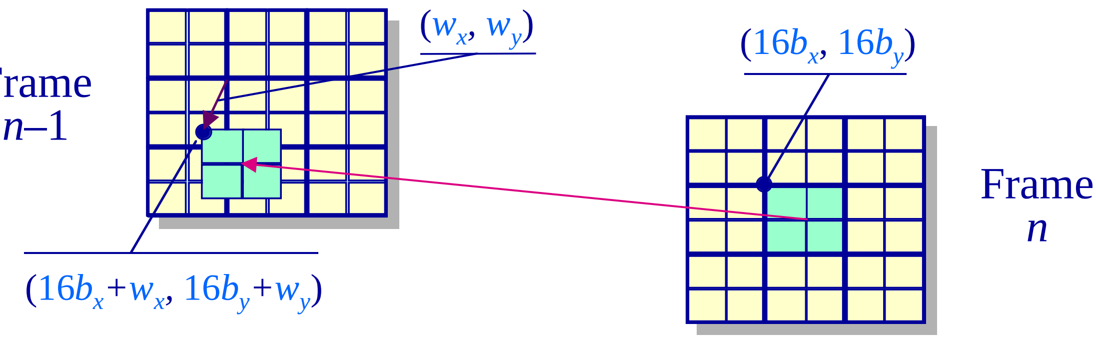
Motion estimation & prediction



- Motion estimation is performed only on luminance macroblocks
  - » Compare a luminance macroblock with its neighbors in the previous frame
  - » If the difference is small, do not compress the block, only record location of matching block
  - » If the difference is “large” send the *difference* between this macroblock and a previous neighboring macroblock into the DCT compression pipeline

H.261 INTER-Frame Mode

Motion estimation & prediction

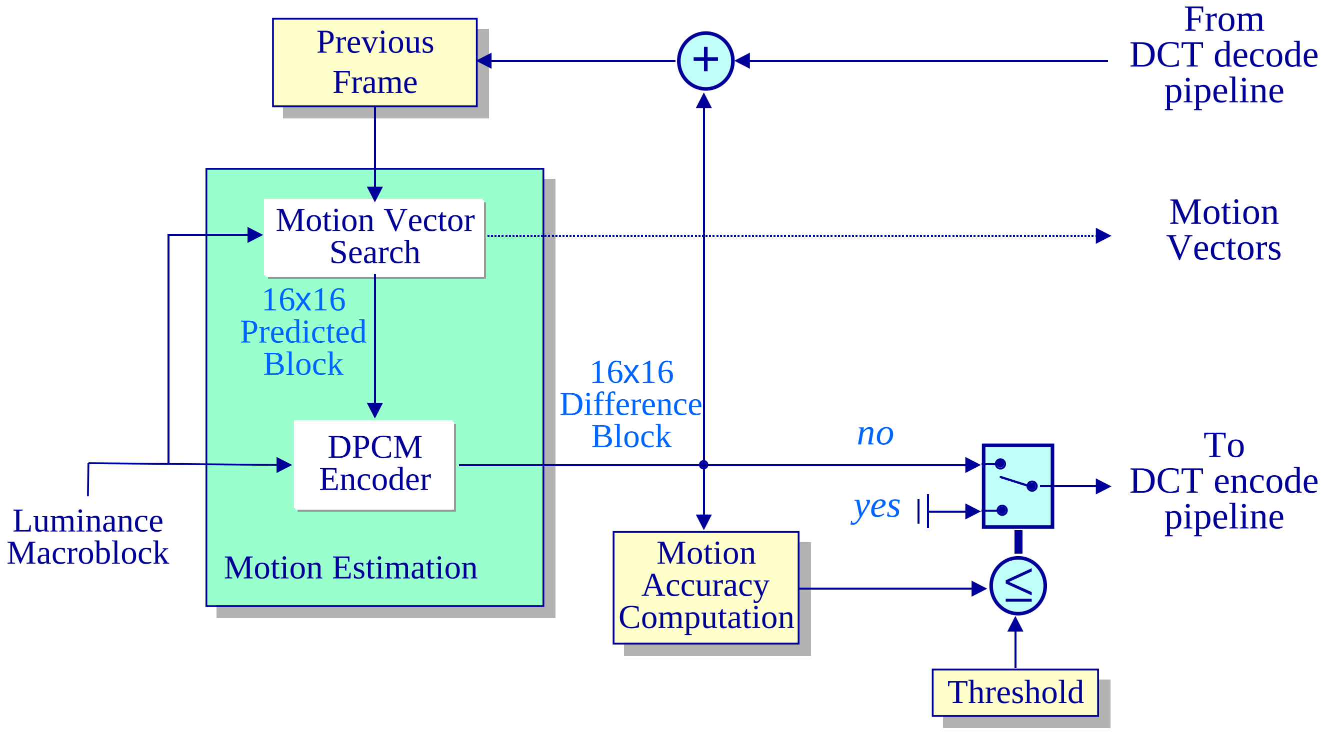


- Finding a predictor is the process of finding the minimally different adjacent 16x16 block in the previous frame
  - » Construct a "motion vector" — a relative displacement w for block b that minimizes the *mean absolute distortion* (*MAD*):

$$\frac{1}{256}\sum_{j=0}^{15}\sum_{i=0}^{15}\left|frame_n[16b_x+i,\ 16b_y+j]-frame_{n-1}[(16b_x+w_x)+i,\ (16b_y+w_y)+j]\right|$$

H.261 INTER-Frame Mode

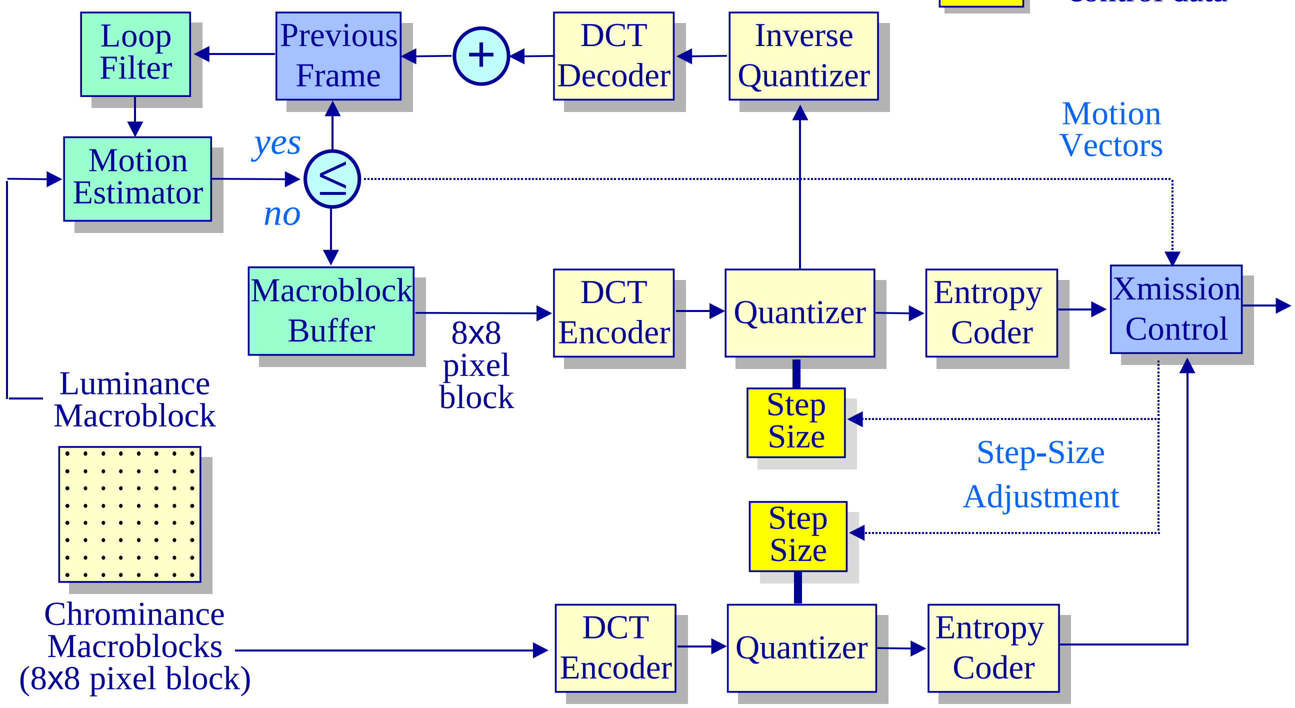
Motion compensated prediction



H.261 INTER-frame Mode

Complete pipeline

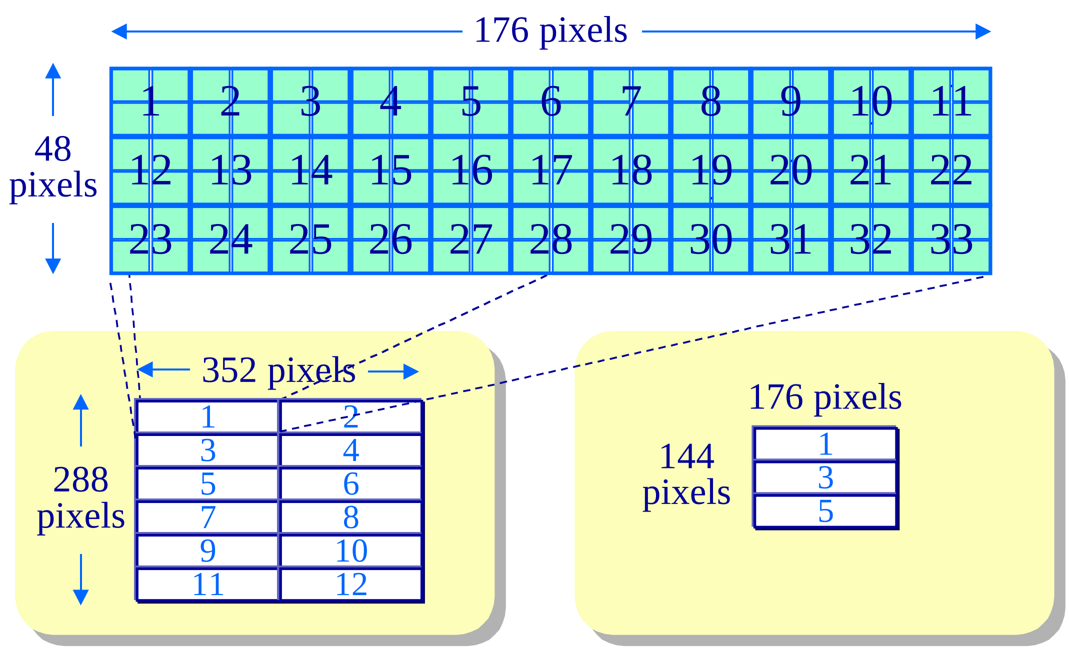
= video frame buffer
= 16x16 pixel block
= 8x8 pixel block
= control data



H.261

Video frame representation

- Macroblocks combined into *groups of blocks* (GOBs)
  - » An 11 by 3 array of macroblocks

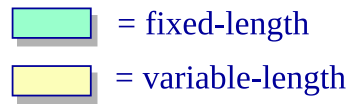


CIF Image: 12 GOBs

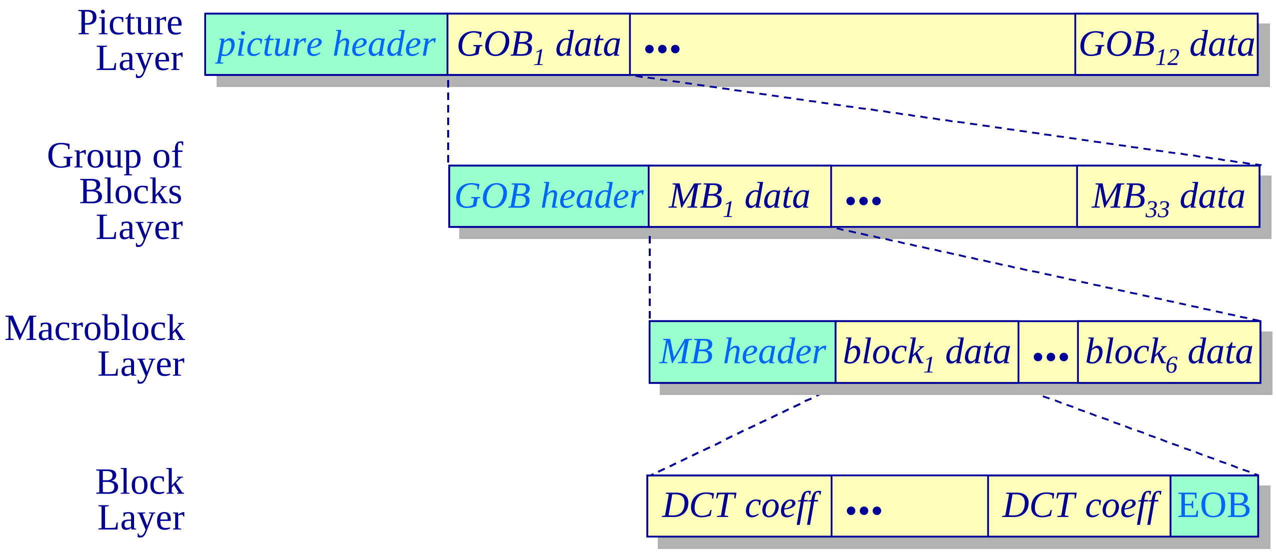
QCIF Image: 3 GOBs

H.261 Data Transmission

Bit-stream format



- Picture data is hierarchically transmitted



ITU H.320 Teleconferencing Standards

Teleconferencing over ISDN

- H.261 — Video communications at *p*x64 kbps
- H.221 — Syntax for multiplexing audio and video packets
- H.230 — Protocol for call setup and negotiation of end-system (“terminal”) capabilities
- H.242 — Conference control protocol
- G.711 — ISDN audio coding standard at 64 kbps
- G.722 — High-quality audio at 64 kbps
- G.728 — Reduced quality speech at 16 kbps

H.263 Video Compression

Low-bitrate video compression for data networks

- Based on H.261 (& MPEG-1, -2)
- Includes new image formats:

Format	Image Size	Maximum Number of coded bits/picture
sub-QCIF	128 x 96	64
QCIF	176 x 144	64
CIF	352 x 288	256
4CIF	704 x 576	512
16CIF	1,408 x 1,152	1024

- Added coding efficiency from:
  - » Unrestricted motion vectors
  - » Bi-directional motion estimation/prediction
  - » Arithmetic coding of AC coefficients

H.263 Video Compression

Companion standards

- H.263 — "Low bit-rate" video coding
- H.324 — Terminal systems
- H.245 — Conference control
- H.223 — Audio/video multiplexing
- G.723 — Audio coding 5.3 and 6.3 kbps

- For Internet conferencing there is also the related T.120 Document Conferencing standards family

The Video Data Type

Compression Standards

- Basic compression techniques
  - Truncation, CLUT, run-length coding
  - sub-sampling & interpolation
  - DPCM
  - DCT
  - Huffman coding
- Common algorithms
  - JPEG/MJPEG
  - H.261/H.263
  - MPEG-1,-2

Compression Algorithms

MPEG

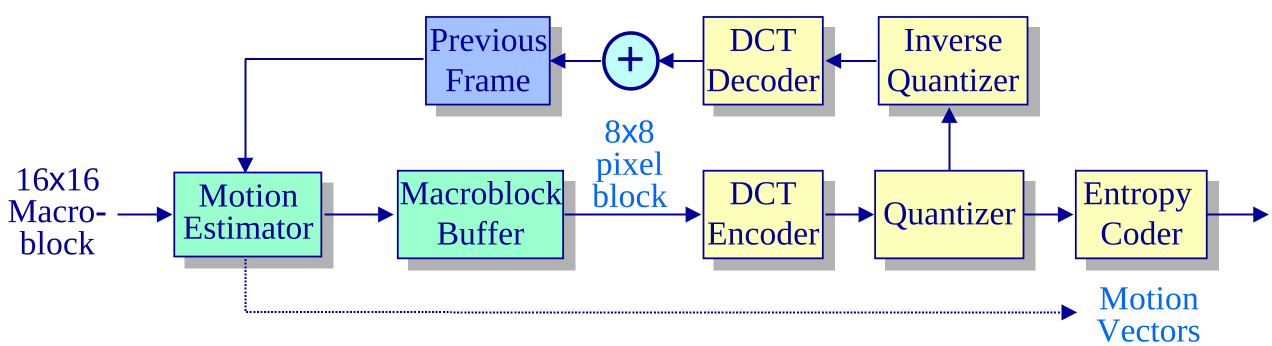
- A family of audio/video coding schemes
  - MPEG-1 —A video coding standard for digital storage/ retrieval devices
    - “VHS quality” video coded at approximately 1.5 Mbps
  - MPEG-2 — Video coding for digital television
    - SIF/CIF to HDTV resolutions at data rates up to 100 Mbps
  - MPEG-4 — Coding of audio/visual “objects” for multimedia applications
    - Coding of natural & synthetic images
    - Object-based encoding for content access & manipulation
  - MPEG-7 — A content/meta-data representation standard for content search and retrieval

MPEG-1 Video Compression
Requirements

- MPEG intended primarily for stored video applications
  - » A "generic" standard
  - » But a basic assumption is that video will be coded once and played multiple times
- Support for VCR-like operations
  - » Fast forward/forward scan
  - » Rewind/reverse scan
  - » Direct random access
  - » ...

MPEG-1 Video Compression
Relation to H.261

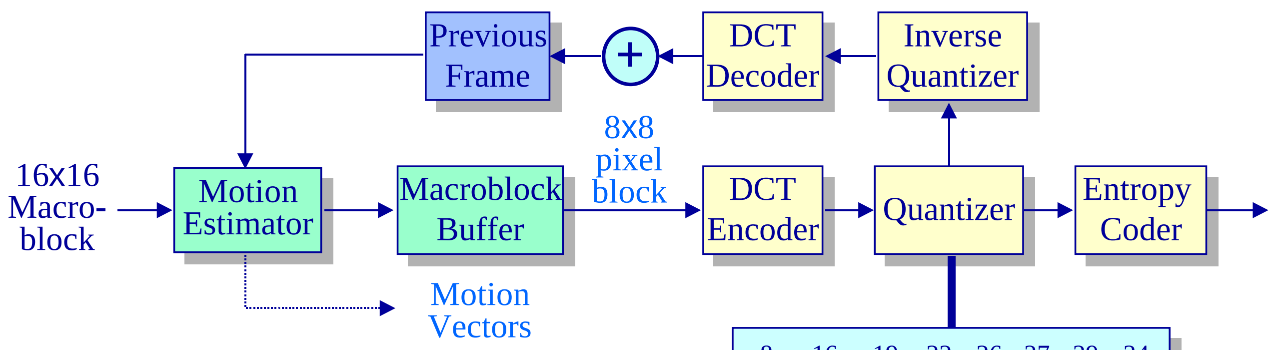
- Similar to H.261...
  - » INTER and INTRA picture types, entropy encoded, motion compensated, DCT-based compression...



- … but with more aggressive motion compensation:
  - » prediction — approximately the same as in H.261
  - » interpolation (bi-directional prediction)
  - » DPCM encoding of motion vectors

MPEG-1 Video Compression
Relation to JPEG

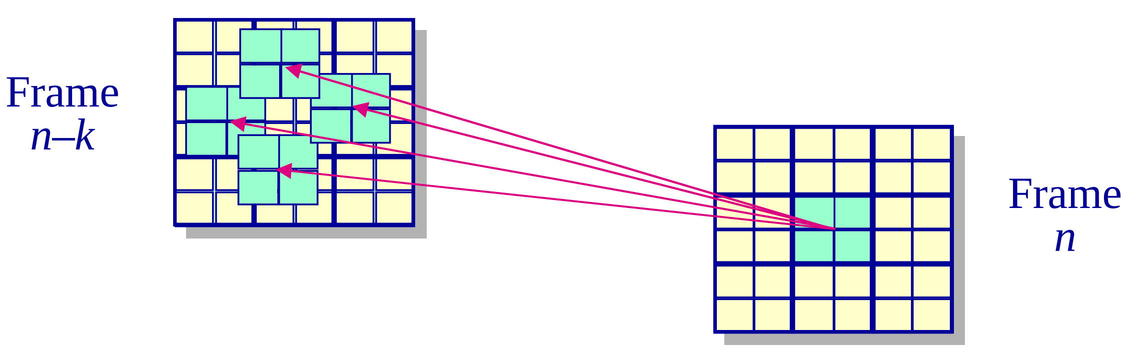
- Non-uniform quantization for intra-coded pictures
  - » Uniform quantization for inter-coded pictures



Step-size table

8	16	19	22	26	27	29	34
16	19	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

MPEG Video Compression
Motion compensated prediction

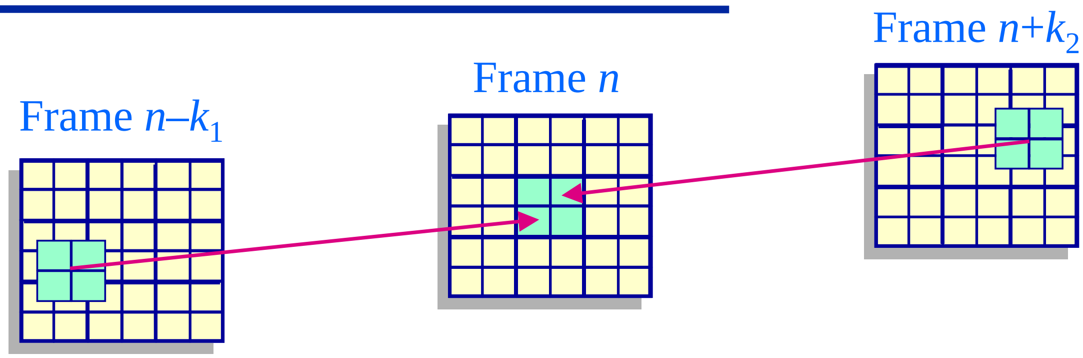


- The predictor search space is not specified in the standard
  - » Implementations can perform as exhaustive a search as they desire
- Find the motion vector w that minimizes some cost function f

$$\sum_{j=0}^{15}\sum_{i=0}^{15} f(frame_n[16b_x+i,\ 16b_y+j] - frame_{n-1}[16b_x+w_x+i,\ 16b_x+w_y+j])$$

MPEG Motion Compensated Prediction

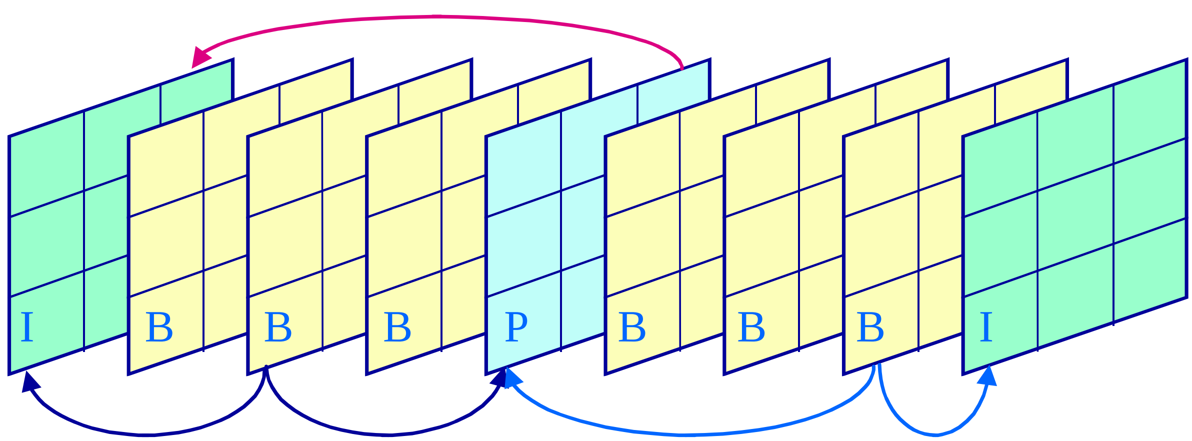
Bi-directional prediction



- Besides simple prediction, *interpolation* (bi-directional prediction) is used to achieve further compression
- A future frame *and* a past frame are used to predict the current frame
  - » Deals effectively with scene changes and new object appearances
  - » Produces predictors (pairs) with better statistical properties

MPEG Motion Compensated Prediction

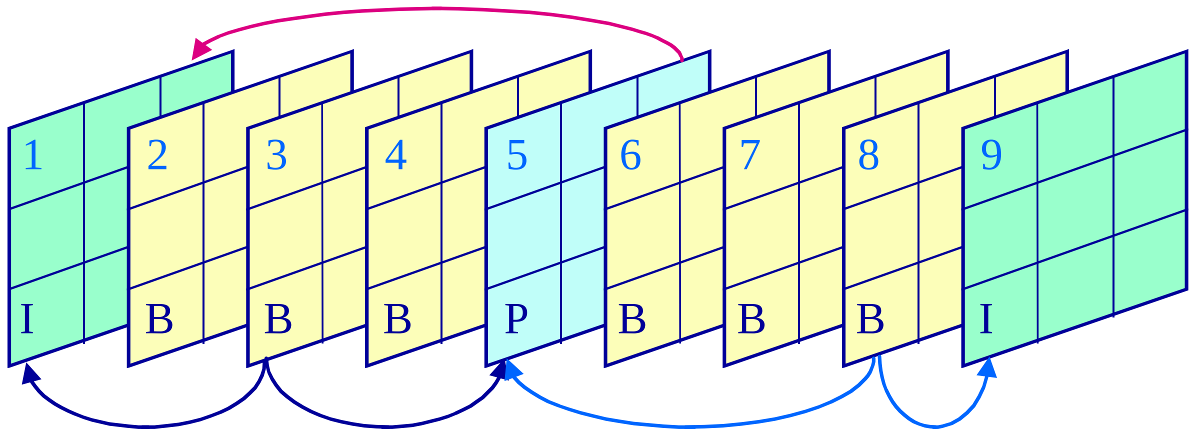
Bi-directional prediction



- MPEG defines three picture (frame) types:
  - » I — intracoded pictures coded as a still image
  - » P — predicted pictures predicted from the previous I or P picture
  - » B — interpolated pictures predicted from the previous I or P picture and the next I or P picture

MPEG Motion Compensated Prediction

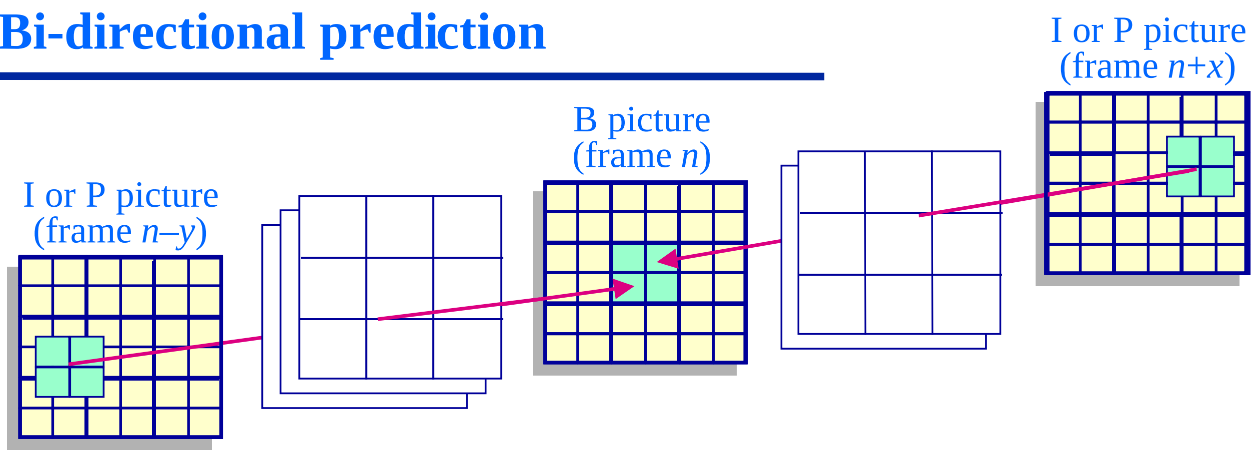
Bi-directional prediction



- Directional prediction implies that frames cannot be encoded or transmitted in the order they are scanned
- Encoding & transmission order:
  - $I_1\ P_5\ B_2\ B_3\ B_4\ I_9\ B_6\ B_7\ B_8\ P_{13}\ B_{10}\ B_{11}\ B_{12}\ I_{17}\ B_{14}\ B_{15}\ B_{16}$ ...

MPEG Motion Compensated Prediction

Bi-directional prediction



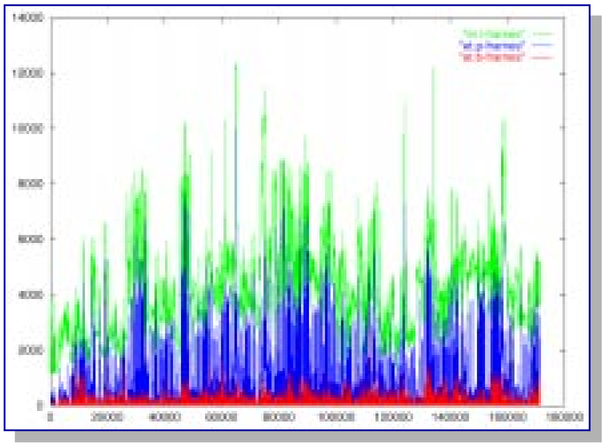
- Bi-directional prediction modes are selectable on a macroblock by macroblock basis within a B picture
- Macroblocks can be predicted by:
  - themselves: $frame'_n[i, j] = frame_n[i, j]$
  - a previous frame: $frame'_n[i, j] = frame_{n-y}[i+w_x, j+w_y]$
  - a future frame: $frame'_n[i, j] = frame_{n+x}[i+w_x, j+w_y]$
  - a previous & future frame: $frame'_n[i, j] = (frame_{n-y}[i+w_x, j+w_y] + frame_{n+x}[i+w'_x, j+w'_y])/2$

Bi-Directional Motion Compensation

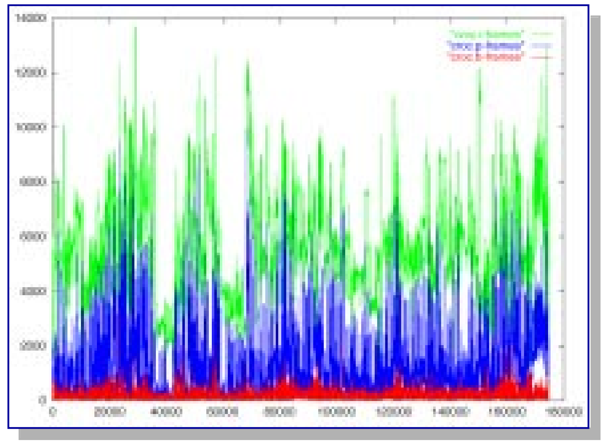
Compression rates

- Some prototypical results for 2 movies encoded at 320x240, 30 fps and constant quality

“Crocodile Dundee”



“ET”



MPEG Video Compression

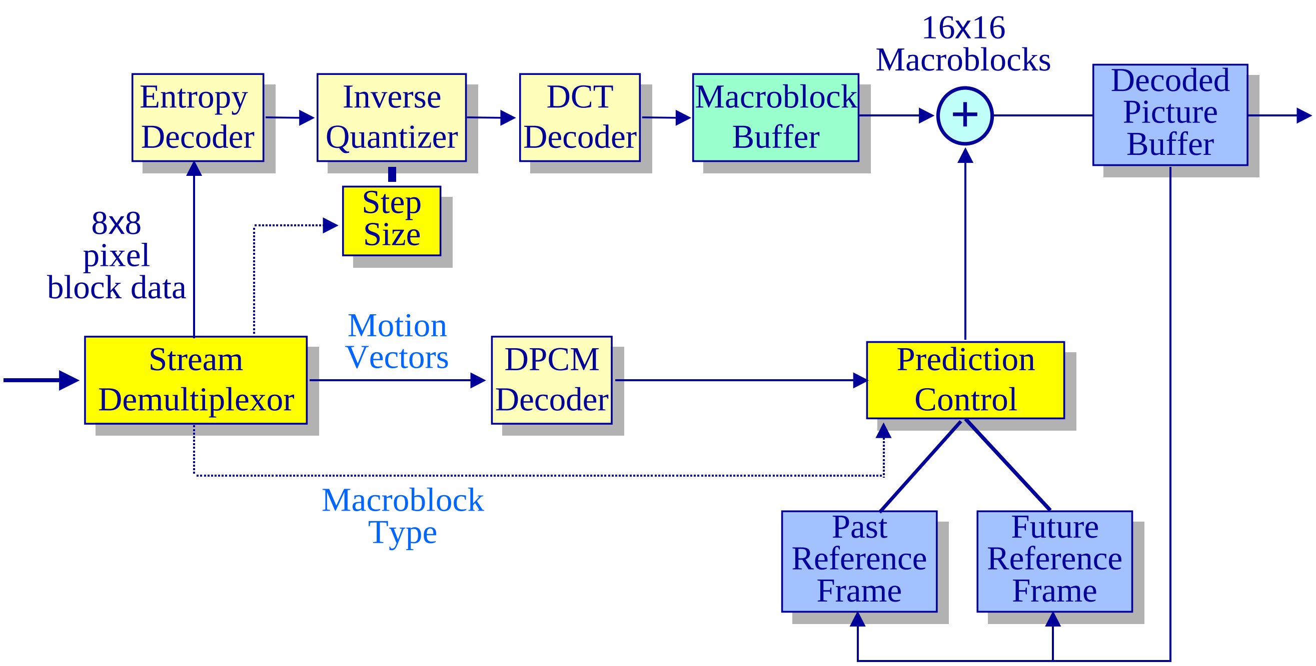
Decoder architecture

= video frame

= 16x16 pixel block

= 8x8 pixel block

= control data



MPEG Video Compression

Coded bit-stream

- MPEG has a layered bit-stream similar to H.261
- There are seven layers:
  - » Sequence Layer
    - ❖ decoding parameters (bit-rate, buffer size, picture resolution, frame rate, ...)
  - » Group of Pictures Layer
    - ❖ a random access point
  - » Picture Layer
    - ❖ picture type and reference picture information
  - » Slice Layer
    - ❖ position and state information for decoder resynchronization
  - » Macroblock Layer
    - ❖ coded motion vectors
  - » Block Layer
    - ❖ coded DCT coefficients, quantizer step size, *etc.*

MPEG-2

"New & Improved" MPEG-1

- A coding standard for the broadcast industry
  - » Coding for video that originates from cameras
  - » Offers little benefit for material originally recorded on film
- But included is support for:
  - » Higher (chrominance) sampling rates
  - » Resilience to transmission errors
  - » ...
- More mature and powerful coding/compression technology is used
  - » Unrestricted motion search with 1/2 pel resolution for motion vectors