Lecture 6: Compression II

Reading: book chapter 8, Section 1, 2, 3, 4

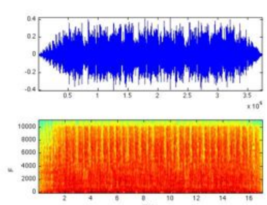
This Week's Schedule

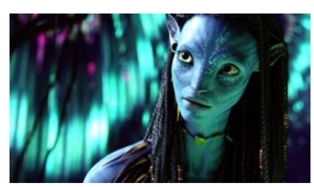
- Monday
  - The concept behind compression
  - Rate distortion theory
  - Image compression via DCT
- Today
  - Speech compression via Prediction
  - Video compression via IPB and motion estimation/compensation

RECAP

Redundancy in Media Data

- Medias (speech, audio, image, video) are not random collection of signals, but exhibit a similar structure in local neighborhood
  - **Temporal redundancy**: current and next signals are very similar (smooth media: speech, audio, video)
  - **Spatial redundancy**: the pixels' intensities and colors in local regions are very similar
  - **Spectral redundancy**: When the data is mapped into the frequency domain, a few frequencies dominate over the others





Lossless Compression

- Lossless compression
  - Compress the signal but can reproduce the exact original signal
  - Used for archival purposes and often medical imaging, technical drawings
  - Assign new binary codes to represent the symbols based on the frequency of occurrence of the symbols in the message
  - Example 1: Run Length Encoding (BMP, PCX) BBBBEEEEEEEECCCCDAAAAA → 4B8E4C1D5A
  - Example 2: Lempel-Ziv-Welch (LZW): adaptive dictionary, dynamically create a dictionary of strings to efficiently represent messages, used in GIF & TIFF
  - Example 3: Huffman coding: the length of the codeword to present a symbol (or a value) scales inversely with the probability of the symbol's appearance, used in PNG, MNG, TIFF

Lossy Compression

- The compressed signal after de-compressed, does not match the original signal
  - Compression leads to some signal distortion
  - Suitable for natural images such as photos in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a **substantial reduction** in bit rate.
- Types
  - **Color space reduction**: reduce 24→8bits via color lookup table
  - **Chrominance subsampling**: from 4:4:4 to 4:2:2, 4:1:1, 4:2:0, eye perceives spatial changes of brightness more sharply than those of color, by averaging or dropping some of the chrominance information
  - **Transform coding** (or perceptual coding): Fourier transform (DCT, wavelet) followed by quantization and entropy coding

Monday's focus

A Typical Compression System

Transform original data into a new representation that is easier to compress

Use a limited number of levels to represent the signal values

Find an efficient way to represent these levels using binary bits

→ Transformation → Quantization → Binary Encoding →

Temporal Prediction
DCT for images
Model fitting

Scalar quantization
Vector quantization
Uniform quantization
Nonuniform quantization

Fixed length
Variable length
(Run-length coding
Huffman coding…)

A Typical Image Compression System

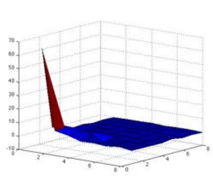
Transform original data into a new representation that is easier to compress

Use a limited number of levels to represent the signal values

Find an efficient way to represent these levels using binary bits

→ Transformation → Quantization → Binary Encoding →

DCT for images
+Zigzag ordering



Scalar quantization

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

(Run-length coding
Huffman coding…)

DC: prediction + Huffman
AC: run-length + Huffman

Summary of Monday's Learning

- The concept behind compression and transformation
- How to perform 2D DCT: forward and inverse transform
  - Manual calculation for small sizes, using inner product notation
  - Using Matlab: dct2, idct2
- Why DCT is good for image coding
  - Real transform, easier than DFT
  - Most high frequency coefficients are nearly zero and can be ignored
  - Different coefficients can be quantized with different accuracy based on human sensitivity
- How to quantize & code DCT coefficients
  - Varying step sizes for different DCT coefficients based on visual sensitivity to different frequencies; A quantization matrix specifies the default quantization stepsize for each coefficient; The matrix can be scaled using a user chosen parameter (QP) to obtain different trade-offs between quality and size
  - DC: prediction + huffman; AC: run-length + huffman

Today's Journey

- Speech/Audio/Video Compression!
- What is the similarity between Speed/Audio/Video?
  - How they are different from image?

A video consists of a time-ordered sequence of frames, i.e., images.

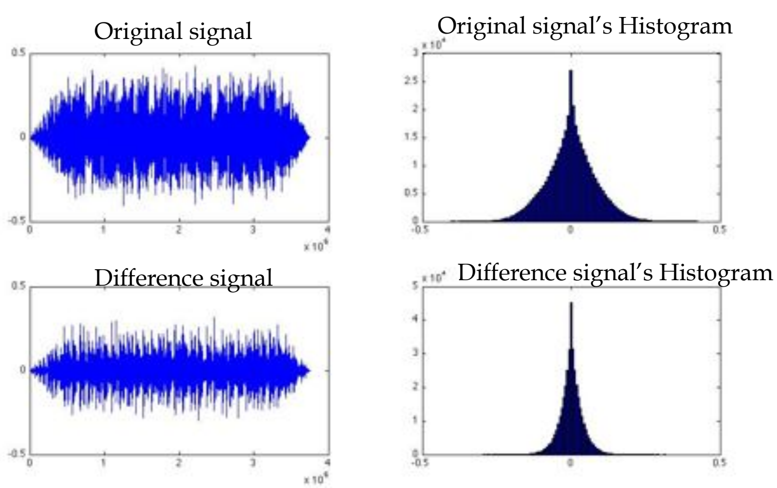
Compressing Speech via Temporal Prediction

- Consecutive speech (audio) samples are similar
- Simply transmitting differences between consecutive samples
  - predict the next sample as being equal to the current sample;
  - send not the sample itself but **the difference** between previous and next
  - It is often the case that some **function** of a few of the previous values, $f_{n-1}, f_{n-2}, f_{n-3}$, etc., provides a better prediction

- Matlab Demo

```
[x,fs]=wavread('funky.wav');
subplot(2,2,1); plot(x);  subplot(2,2,2); hist(x,100);
y=x(2:length(x))-x(1:length(x)-1);
subplot(2,2,3); plot(y); subplot(2,2,4); hist(y,100);
```

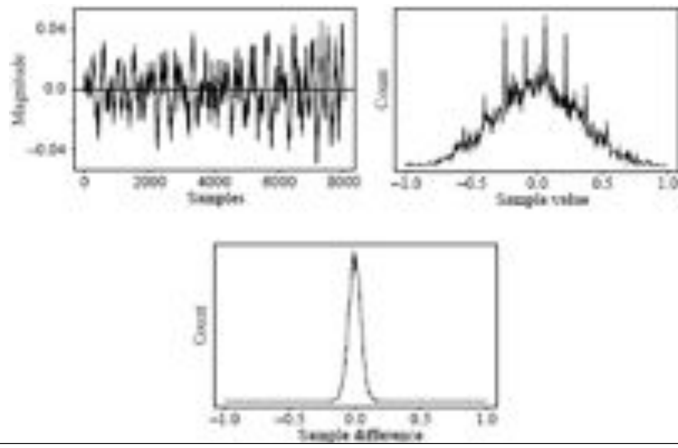
Demo Results



Much smaller range
→ easier to encode

Another Example

- Differencing concentrates the histogram.
  (a): Digital speech signal.
  (b): Histogram of digital speech signal values.
  (c): Histogram of digital speech signal differences.



Compression Process

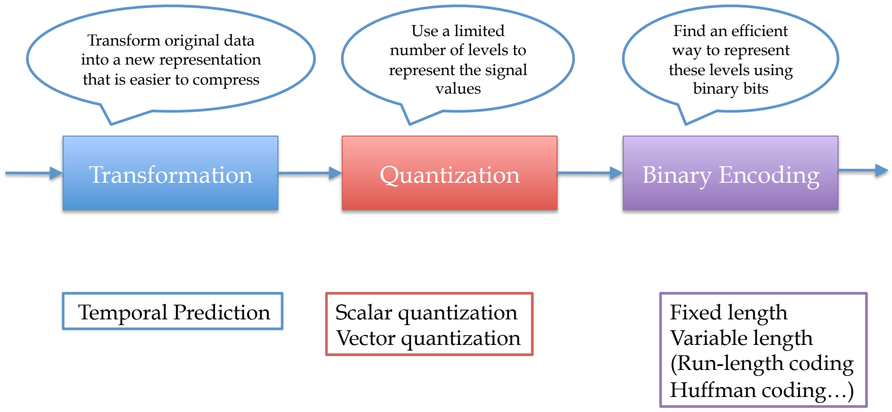
- Suppose we wish to code the sequence $f_1; f_2; f_3; f_4; f_5$ = 21, 22, 27, 25, 22.
- Instead of transmitting f_n, transmit f_n-f_{n-1}

Encoding:	Decoding:
initialize send f_0'=21,	initialize receive f_0'=21,
$e_1=f_1-f_0'=0$, send $e_1=0$,	Receive $e_1=0$, recover $f_1'=f_0'+e_1=21$
$e_2=f_2-f_1'= 22 - 21= 1$	Receive $e_2=1$, recover $f_2'=f_1'+e_2= 22$
$e_3=f_3-f_2'= 27 - 22 = 5$	Receive $e_3=5$, recover $f_3'=f_2'+e_3= 27$
$e_4=f_4-f_3'= 25 - 27 =-2$	Receive $e_4=-2$, recover $f_4'=f_3'+e_4= 25$

- Instead of sending 21, 22, 27, 25, …, now send 0, 1, 5, -2 (much smaller range)
- Much smaller range → better quantization efficiency
- Can use run-length, or huffman coding to efficiently store e_n

- See DPCM in Book Chapter 6, Section 5 (Differential PCM)

A Typical Speech Compression System



VIDEO COMPRESSION

Various Video Formats

Video Format	Y Size	Color Sampling	Frame Rate (Hz)	Raw Data Rate (Mbps)
HDTV Over air. cable, satellite, MPEG2 video, 20-45 Mbps				
SMPTE296M	1280x720	4:2:0	24P/30P/60P	265/332/664
SMPTE295M	1920x1080	4:2:0	24P/30P/60I	597/746/746
Video production, MPEG2, 15-50 Mbps				
BT.601	720x480/576	4:4:4	60I/50I	249
BT.601	720x480/576	4:2:2	60I/50I	166
High quality video distribution (DVD, SDTV), MPEG2, 4-10 Mbps				
BT.601	720x480/576	4:2:0	60I/50I	124
Intermediate quality video distribution (VCD, WWW), MPEG1, 1.5 Mbps				
SIF	352x240/288	4:2:0	30P/25P	30
Video conferencing over ISDN/Internet, H.261/H.263/MPEG4, 128-384 Kbps				
CIF	352x288	4:2:0	30P	37
Video telephony over wired/wireless modem, H.263/MPEG4, 20-64 Kbps				
QCIF	176x144	4:2:0	30P	9.1

Video Compression =?= Image Compression

- Why can we compress an image
  - Adjacent pixels are correlated (have similar color values)
- How to compress (the JPEG way)
  - Use transform to decorrelate the signal (DCT)
  - Quantize the DCT coefficients
  - Runlength code the quantized indices
- What is different with video?
  - We can apply JPEG to each video frame (Motion-JPEG)
  - But we can do more than that to achieve higher compression!

Motivation

- A video consists of a time-ordered sequence of frames, i.e., images.
  - Adjacent frames are similar
  - Changes are due to object or camera motion

Frame 66

Frame 69

An Obvious Way to Compress

- An obvious solution: **Temporal Prediction**
  - Predictive coding based on previous frames.
  - Compression proceeds by subtracting images: subtract in time order and code the residual error.
- It can be done even better by
  - **searching for just the right parts of the image to subtract from the previous frame.**

Key Concepts of Video Compression

- **Temporal Prediction**: (INTER mode)
  - Predict a new frame from a previous frame and only specify the prediction error
  - Prediction error will be coded using an image coding method (e.g., DCT-based JPEG)
  - Prediction errors have smaller energy than the original pixel values and can be coded with fewer bits
- **Motion-compensation** to improve prediction:
  - Use motion-compensated temporal prediction to account for object motion
- **INTRA frame coding**: (INTRA mode)
  - Those regions that cannot be predicted well are coded **directly** using DCT-based method
- **Spatial prediction**:
  - Use spatial directional prediction to exploit spatial correlation (H.264)
- **Work on each macroblock (MB) (16x16 pixels)** independently for reduced complexity
  - Motion compensation done at the MB level
  - DCT coding of error at the block level (8x8 pixels or smaller)
  - Block-based hybrid video coding

Key Concepts of Video Compression

- **Temporal Prediction**: (INTER mode)
  - Predict a new frame from a previous frame and only specify the prediction error
  - Prediction error will be coded using an image coding method (e.g., DCT-based JPEG)
  - Prediction errors have smaller energy than the original pixel values and can be coded with fewer bits
- **Motion-compensation** to improve prediction:
  - Use motion-compensated temporal prediction to account for object motion
- **INTRA frame coding**: (INTRA mode)
  - Those regions that cannot be predicted well are coded **directly** using DCT-based method
- **Spatial prediction**:
  - Use spatial directional prediction to exploit spatial correlation (H.264)
- **Work on each macroblock (MB) (16x16 pixels)** independently for reduced complexity
  - Motion compensation done at the MB level
  - DCT coding of error at the block level (8x8 pixels or smaller)
  - Block-based hybrid video coding

Motion Estimation/Compensation

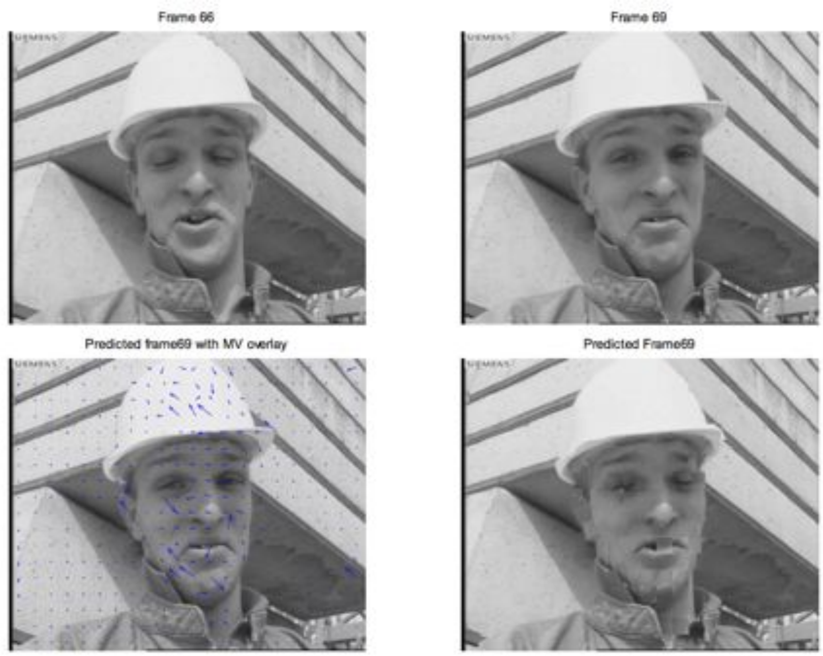
- Each image is divided into *macroblocks* of size *NxN*.
  - By default, *N* = 16 for luminance images.
  - For chrominance images, *N* = 8 if 4:2:0 chroma subsampling is adopted.

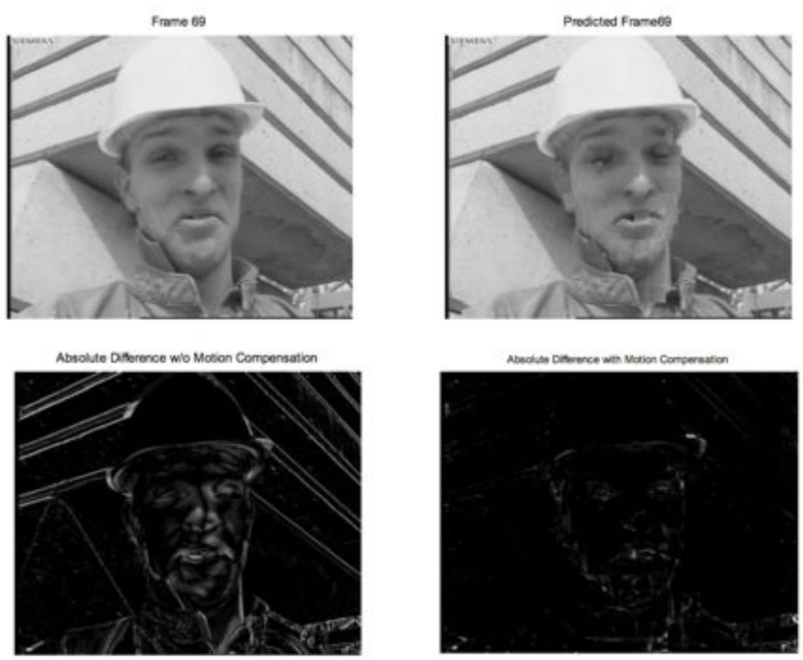
- Motion compensation operates at the macroblock level, using Y
  - The current image frame is referred to as *Target Frame*.
  - **Search:** A match is sought between the macroblock in the Target Frame and the most similar macroblock in previous and/or future frame(s) (referred to as *Reference frame(s)*)
    - Using sum of absolute differences (SAD) between corresponding pixels
  - **Estimation:** The displacement of the reference macroblock to the target macroblock is called a *motion vector* **MV**
  - **Compensation**: Current MB is replaced by the best matching MB (motion- compensated prediction or motion compensation), plus the errors → only code the errors and the MV

An Illustrative Example

Frame *t-1*
(Reference Frame)

Frame *t*
(Predicted frame)


Frame 66
Frame 69
Predicted frame69 with MV overlay
Predicted Frame69

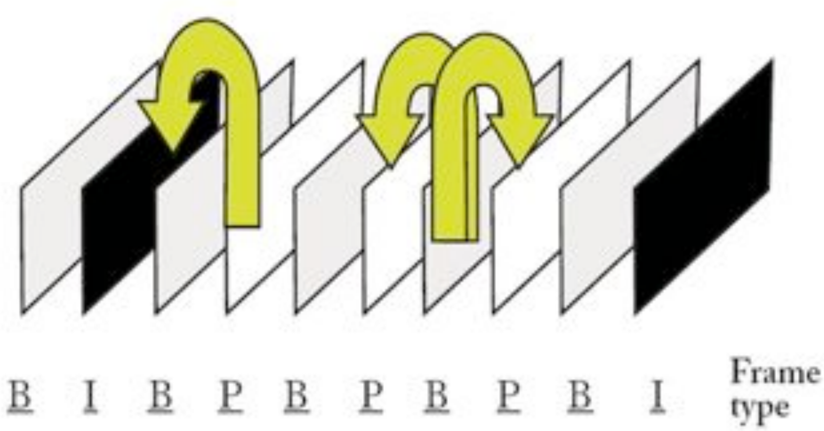

Frame 69
Predicted Frame69
Absolute Difference w/o Motion Compensation
Absolute Difference with Motion Compensation

Again: Temporal Prediction

- No Motion Compensation:
  - Work well in stationary regions
    - $f'(t,m,n) = f(t-1,m,n)$
- *Uni-directional Motion Compensation:*
  - *Does not work well for uncovered regions due to object motion or newly appeared objects*
    - $f'(t,m,n)= f(t-1,m-dx,n-dy)$
- *Bi-directional Motion Compensation*
  - *Can handle better covered/uncovered regions*
    - $f'(t,m,n) = w_b f(t-1,m-db,x,n-db,y) + w_f f(t+1,m-df,x,n-df,y)$

Code: e(t)= $f(t,m,n)-f'(t,m,n)$

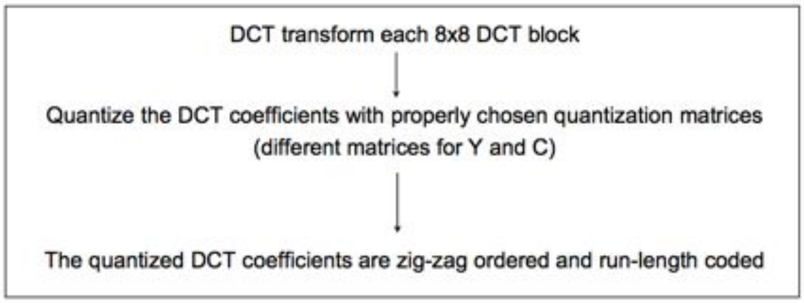
Different Prediction Modes



Intra: coded directly;
Predictive: predicted from a previous frame;
Bidirectional: predicted from a previous frame and a following frame.
Can be done at frame or block levels

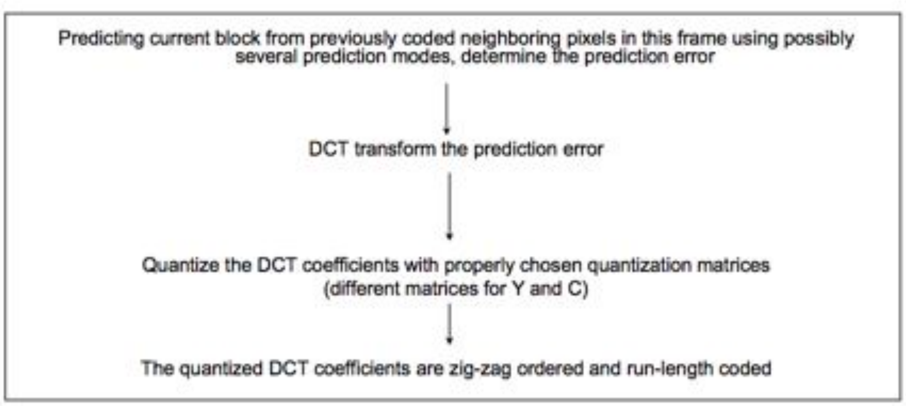
DCT on I frames/blocks

- For I-blocks, DCT is applied to original image values



DCT on I frames/blocks

- For I-blocks, the new H.264 standard applies **intra-prediction (spatial prediction within the same frame)** and DCT is applied to intra-prediction error



DCT on P frames/blocks

- First predict the current frame/block from the previous video frame
- DCT is applied to temporal prediction errors

For each macroblock (16x16), find the best matching block in a previous frame, and calculate the prediction errors

↓

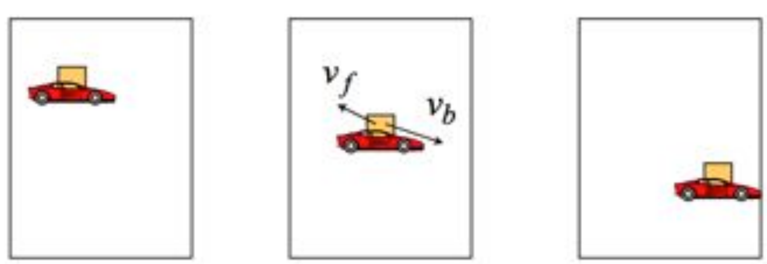
The prediction errors in each of the DCT blocks (8x8) are DCT transformed, quantized (according to specified QP), zig-zag scanned, and run-length coded

↓

1 pair of motion vector (MV) also needs to be coded

DCT on B Frames/blocks

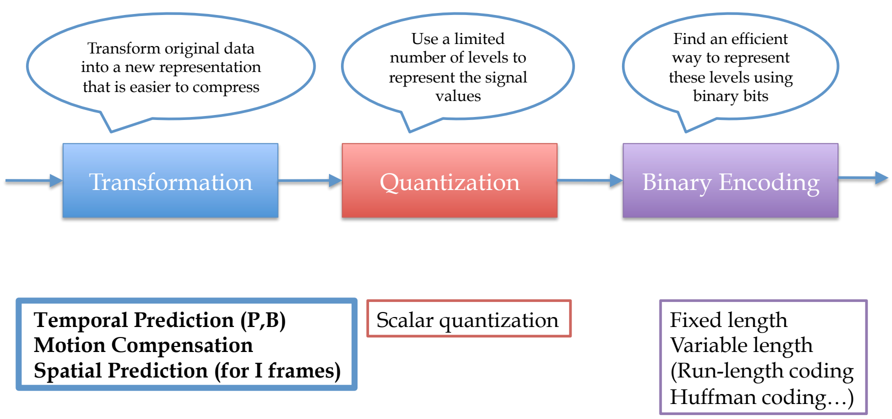
- Same as for the P-mode, except that a macroblock is predicted from both a previous picture frame and a following one
- 2 sets of MVs needed to be coded.



Choosing the Mode for a MB

- Frame-level decision
  - I frame use only I-mode
  - P-frame use P-mode, except when prediction does not work (back to I-mode)
  - B-frame use B-mode (but can switch to P-mode and I-mode)
- Block-level decision
  - A MB is coded using the mode that leads to the lowest bit rate for the same distortion
  - I-mode is used for the first frame, and **is inserted periodically in following frames**, to stop transmission error propagation
  - Mode information is coded in MB header

Summary: A Typical Video Compression System



Various Video Standards

Standards	Application	Video Format	Raw Data Rate	Compressed Data Rate
H.320 (H.261)	Video conferencing over ISDN	CIF QCIF	37 Mbps 9.1 Mbps	>=384 Kbps >=64 Kbps
H.323 (H.263)	Video conferencing over Internet	4CIF/ CIF/ QCIF		>=64 Kbps
H.324 (H.263)	Video over phone lines/ wireless	QCIF	9.1 Mbps	>=18 Kbps
MPEG-1	Video distribution on CD/ WWW	CIF	30 Mbps	1.5 Mbps
MPEG-2	Video distribution on DVD / digital TV	CCIR601 4:2:0	128 Mbps	3-10 Mbps
MPEG-4	Multimedia distribution over Inter/Intra net	QCIF/CIF		28-1024 Kbps
GA-HDTV	HDTV broadcasting	SMPTE296/295	<=700 Mbps	18--45 Mbps
H.264/AVC	Newest video coding standard	All		

H.261

- An earlier digital video compression standard, its principle of **Motion Compensation**-based compression is retained in all later video compression standards.
- Designed for videophone, video conferencing and other audiovisual services over ISDN.
- The video codec supports bit-rates of *p*x64 kbps, where *p* ranges from 1 to 30 (Hence also known as *p*x64).
- Supports QCIF (176x144 for luminance, 88x72 for chrominance)
- Support **I-frames and P-frames but not B-frames, why??**
- Measuring motion vector (MV) in the unit of **pixels**; limited range [-15, 15]

H.261 Coding Process

- I-frames
  - Take an image block of 16x16 pixels → Y 4 (8x8) blocks, Cr 1 (8x8) block, Cb 1 (8x8) block
  - For each of the 6 blocks → apply 8x8 DCT → quantization and zig-zag ordering → entropy coding
- P-frames
  - For each of the 6 blocks, search for a motion vector
  - Measure the difference as prediction error
  - For each of the 6 **difference** blocks → apply 8x8 DCT → quantization and zig-zag ordering → entropy coding
  - If prediction is not helping, code the block as Intra without any prediction
  - Code motion vector using **prediction + entropy coding**
    - MVD = MV_previous – MV_current

H.263 (1995, 1997, 2000)

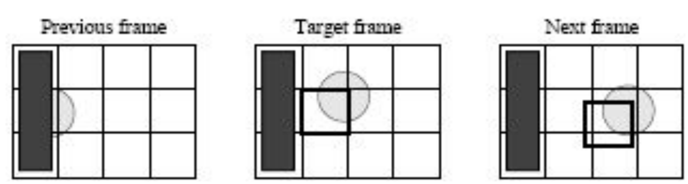
- An improved video coding standard for video conferencing
  - Aims at **low bit-rate** communications at bit-rates of **less than 64 kbps** (aka: improved quality at lower rates)
    - Better video at 18-24 Kbps than H.261 at 64 Kbps
  - Better motion estimation
    - **Half-pixel** precision in motion vector
    - Larger motion search range [-31.5, 31]
    - Use bidirectional temporal prediction (P,B frames) (optional)
  - Uses transform coding for the remaining signal to reduce spatial redundancy (for both Intra-frames and inter-frame prediction).

MPEG-1

- Audio/video on CD-ROM (1.5 Mbps, CIF: 352x240)
  - Maximum: 1.856 Mbps, 768x576 pels
- Start late 1988, test in 10/89, Committee Draft 9/90
- Prompted explosion of digital video applications:
  - MPEG1 video CD and downloadable video over Internet
- MPEG-1 Audio
  - Offers 3 coding options (3 layers), higher layer have higher coding efficiency with more computations
  - MP3 = MPEG1 layer 3 audio

MPEG-1

- Developed at about the same time
- Difference from H.261:
  - Using B-frames/blocks

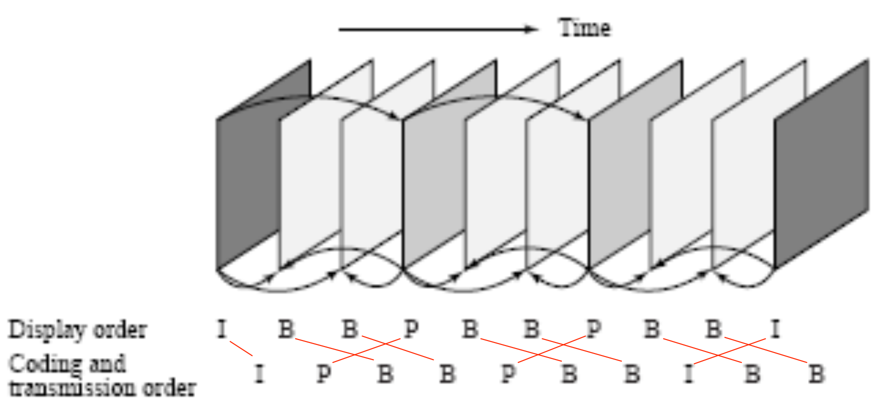


  - MPEG-1 supports SIF (352x240 for NTSC, 352x288 for PAL), H. 261 only supports CIF (352x288) and QCIF (176x144) source formats
  - Higher bandwidth, not for interactive application
  - Use perceptual-based quantization for I-blocks (JPEG like)

B-frames in MPEG

- Each MB from a B-frame will have up to *two* motion vectors (MVs)
  - one from the forward prediction
  - one from the backward prediction.
- If matching in both directions is successful, then two MVs will be sent
  - the two corresponding matching MBs are averaged before comparing to the Target MB for generating the prediction error.
- If an acceptable match can be found in only one of the reference frames, then only one MV and its corresponding MB will be used from either the forward or backward prediction.

MPEG Frame Arrangement



Additional Differences from H.261

- Quantization:
  - MPEG-1 quantization uses different quantization tables for its Intra and Inter coding
- MPEG-1 allows motion vectors to be sub-pixel precision (1/2 pixel).
  - The technique of "bilinear interpolation" (H.263) is used to generate the values at half-pixel locations.
- Compared to the maximum of 15 pixels for motion vectors in H.261, MPEG-1 supports a range of
  - [−512, 511.5] for half-pixel precision
  - [−1024, 1023] for full-pixel precision motion vectors.

MPEG-2

- A/V broadcast (TV, HDTV, Terrestrial, Cable, Satellite, High Speed Inter/Intranet) as well as DVD video
  - 4~8 Mbps for TV quality
  - 10-15Mbps for better quality at SDTV resolutions
  - 18-45 Mbps for HDTV applications
- Test in 11/91, Committee Draft 11/93
- Backward compatible with MPEG-1
- MPEG-2 audio
  - Support 5.1 channel
  - MPEG2 AAC: requires 30% fewer bits than MP3

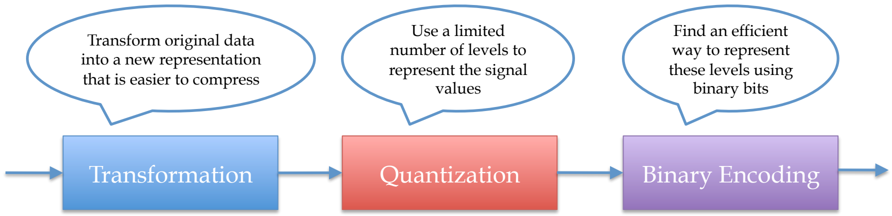
Comparing MPEG 2 to MPEG 1

- MPEG1 only handles progressive sequences (SIF).
- MPEG2 is targeted primarily at higher resolution (BT.601 = 4CIF and HDTV), can handle both progressive and interlaced sequences.
- More sophisticated motion estimation methods are developed to improve estimation accuracy for interlaced sequences.
- Different DCT modes and scanning methods are developed for interlaced sequences.
- MPEG2 has various scalability modes
- MPEG2 has various profiles and levels, each combination targeted for a different application

Summary of Various Video Standards

- H.261:
  - First video coding standard, targeted for video conferencing over ISDN
  - Uses block-based hybrid coding framework with integer-pel MC
- H.263:
  - Improved quality at lower bit rate, to enable video conferencing/telephony below 54 bkps (modems, desktop conferencing)
  - Half-pel MC and other improvement
- MPEG-1 video
  - Video on CD and video on the Internet (good quality at 1.5 mbps)
  - Half-pel MC and bidirectional MC
- MPEG-2 video
  - SDTV/HDTV/DVD (4-15 mbps)
  - Extended from MPEG-1, considering interlaced video

A Typical Video Compression System

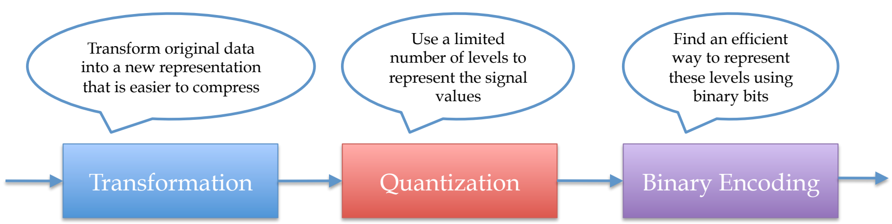


Temporal Prediction (P,B)
Motion Compensation
Spatial Prediction (for I frames)

Scalar quantization
Vector quantization

Fixed length
Variable length
(Run-length coding
Huffman coding…)

A Typical Speech Compression System



Temporal Prediction

Scalar quantization
Vector quantization

Fixed length
Variable length
(Run-length coding
Huffman coding…)

Next Week

- Media Distribution

- Today: Homework #2 assigned
  - Edge detection essay
  - Compression questions/programming tasks

  See Homework/Lab page in the class website or facebook classjournal

  Or directly at http://www.cs.ucsb.edu/~htzheng/teach/cs182/schedule/pdf/hw2.pdf