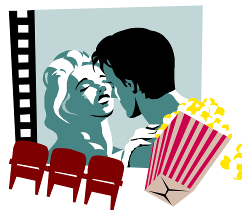
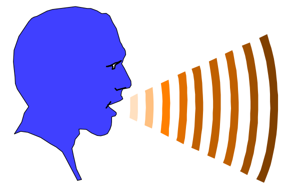





Multimedia Communication Systems II

Video Coding Basics

Yao Wang
Polytechnic University, Brooklyn, NY11201
http: //eeweb.poly.edu/~yao

Outline

- Video application
- Motivation for video coding
- Basic ideas in video coding
- Block diagram of a typical video codec
- Desirable features: scalability and error resilience

Video Applications

- Digital TV/HDTV broadcast over the air/cable/satellite
- Video-on-demand
  - On-line video store
  - CNN news video
- Internet Video broadcast/multicast
- DVD movies
- Home video capture and sharing
- …

Why Compress?

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

Multimedia Communication 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		

Components in a Coding System

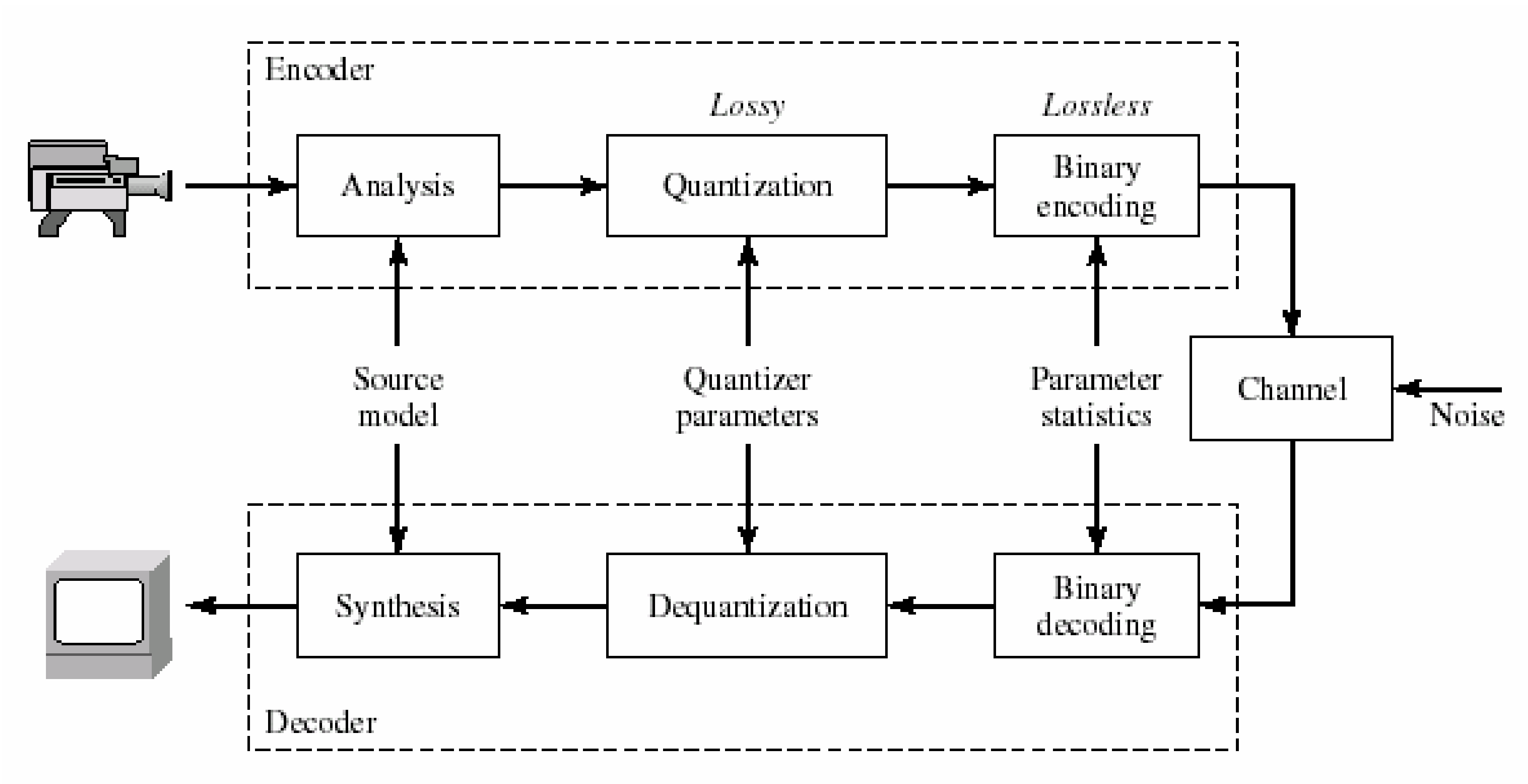
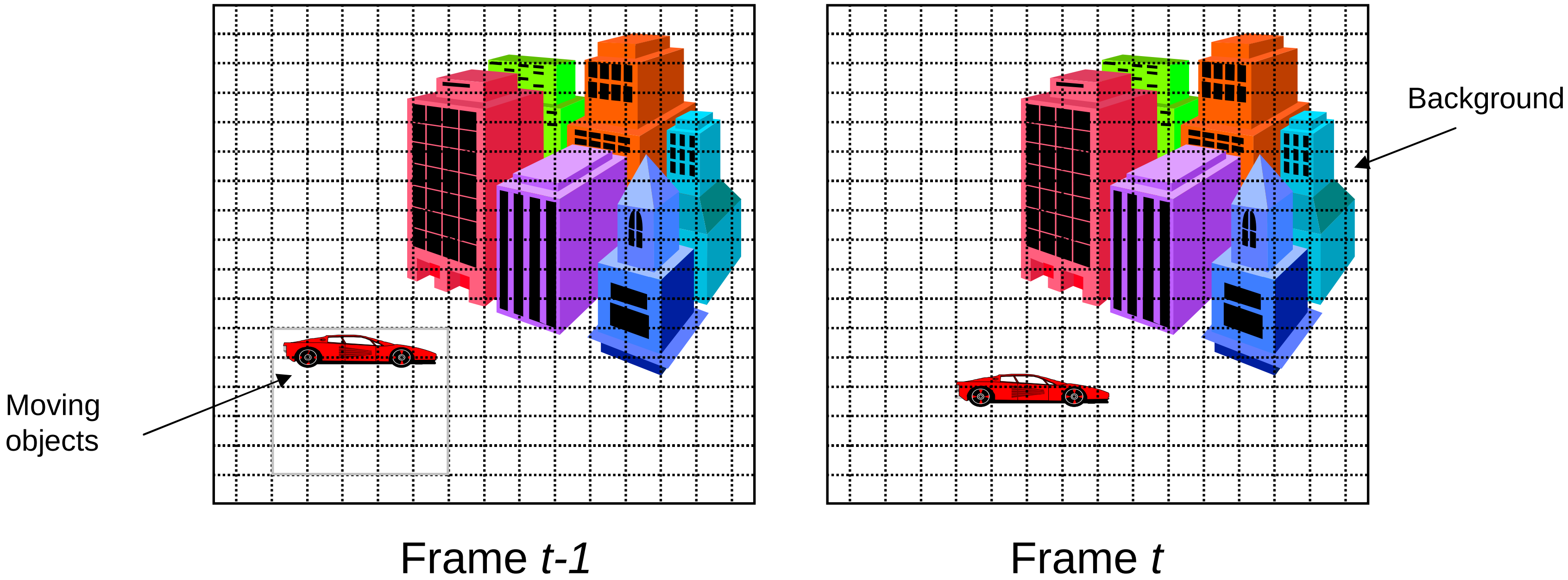


Image Coding Revisited

- 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
    - Zigzag ordering
    - Huffman coding each pair (zero runlength, non-zero value)
- 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!

Characteristics of Typical Videos



Adjacent frames are similar and changes are due to object or camera motion

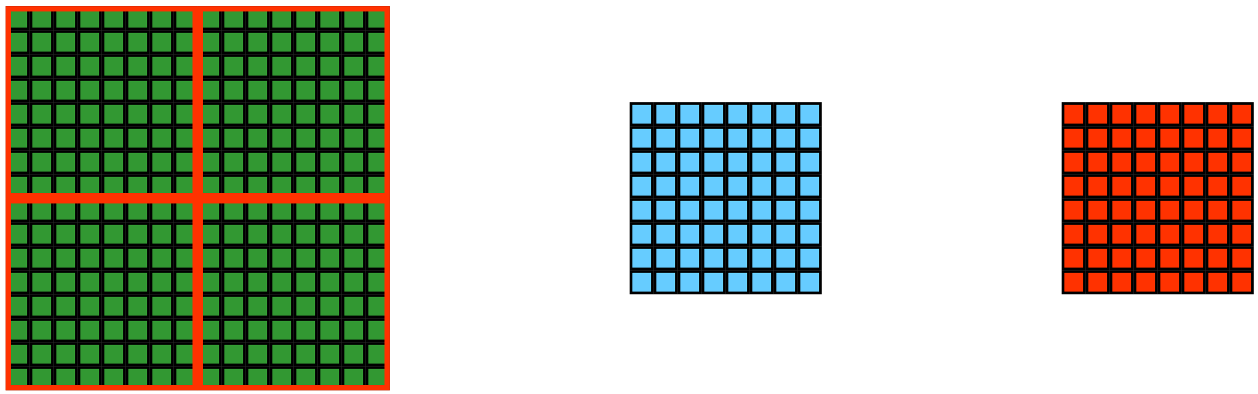
--- Temporal correlation

Frame 66
Frame 69
SIEMENS
SIEMENS
Absolute Difference w/o Motion Compensation
Absolute Difference with Motion Compensation

Key Ideas in Video Coding

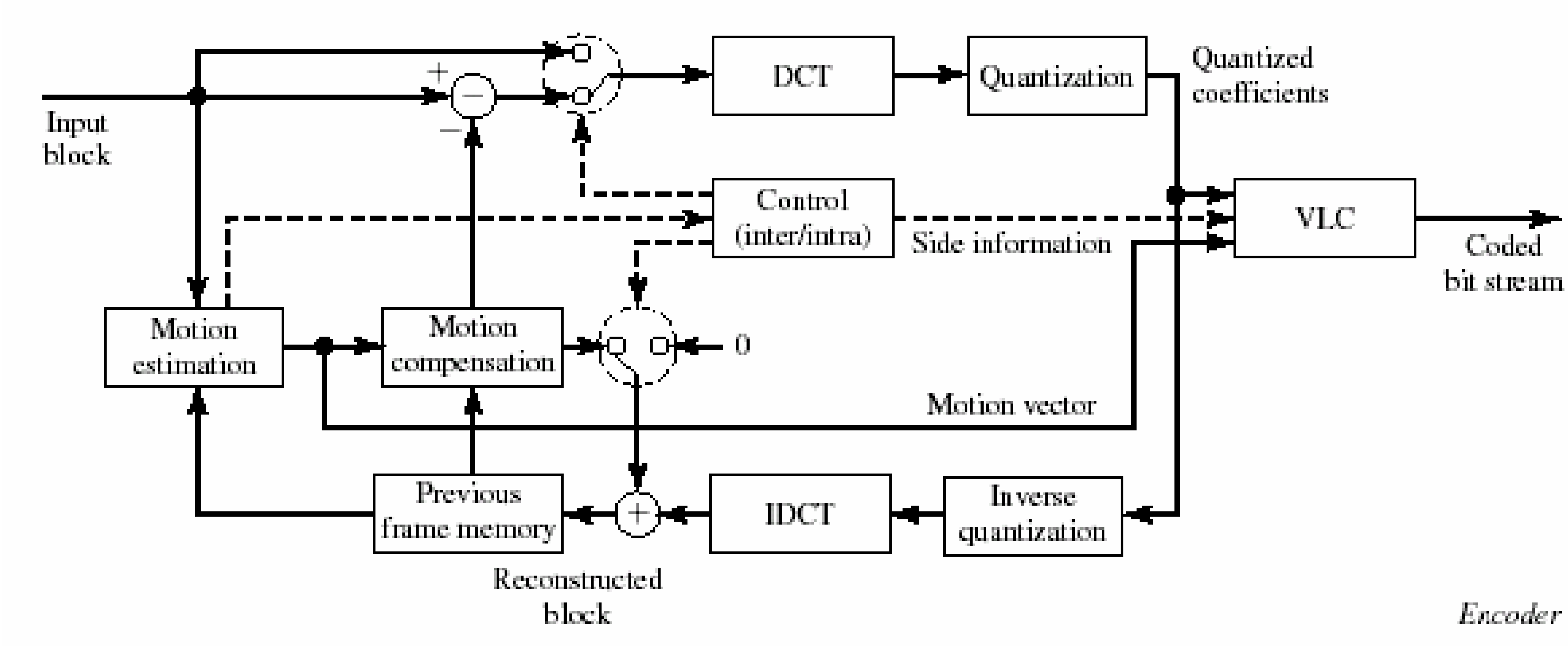
- Predict a new frame from a previous frame and only specify the prediction error (INTER mode)
- Prediction error will be coded using an image coding method (e.g., DCT-based as in JPEG)
- Prediction errors have smaller energy than the original pixel values and can be coded with fewer bits
- Those regions that cannot be predicted well will be coded directly using DCT-based method (INTRA mode)
- Use motion-compensated temporal prediction to account for object motion
- 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

MB Structure in 4:2:0 Color Format



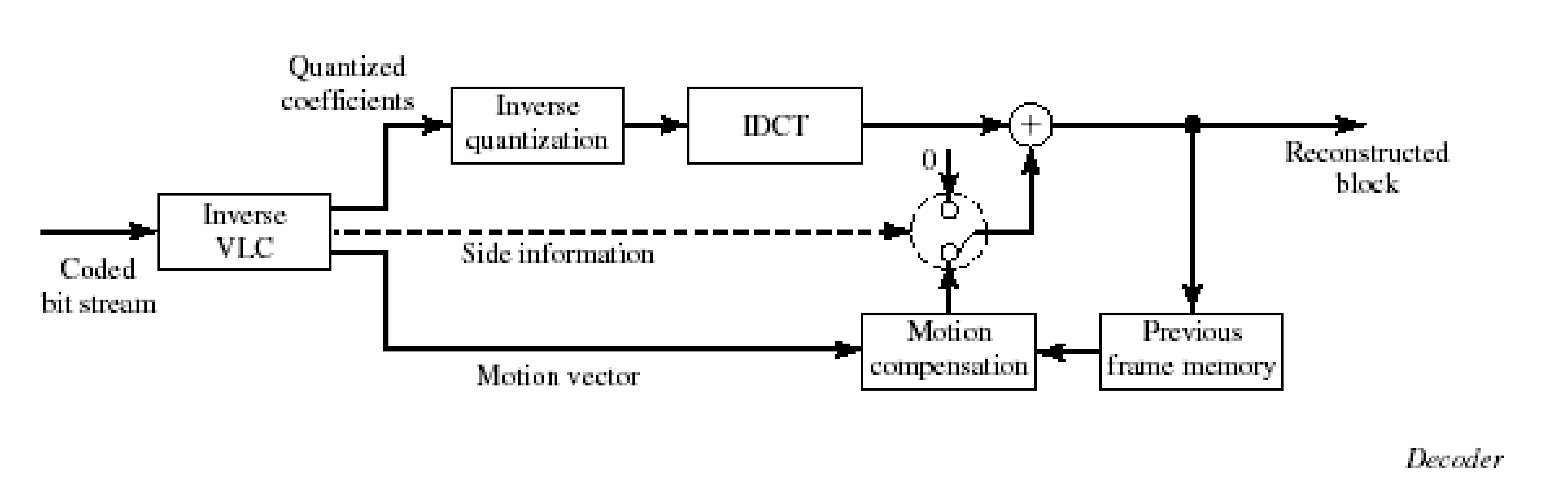
4 8x8 Y blocks   1 8x8 Cb blocks   1 8x8 Cr blocks

Encoder Block Diagram of a Typical Block-Based Hybrid Coder



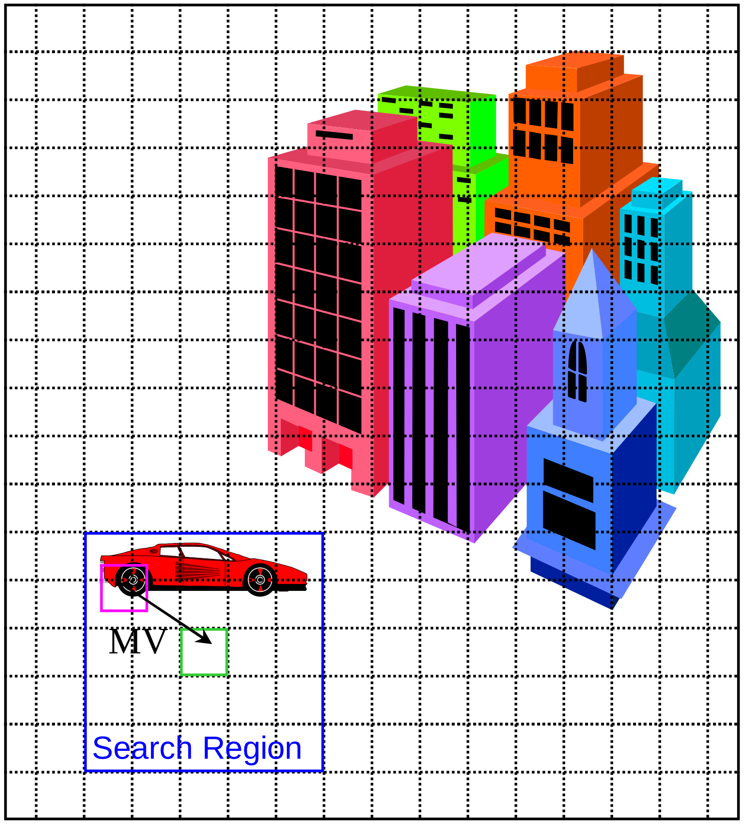
From [Wang02]

Decoder Block Diagram

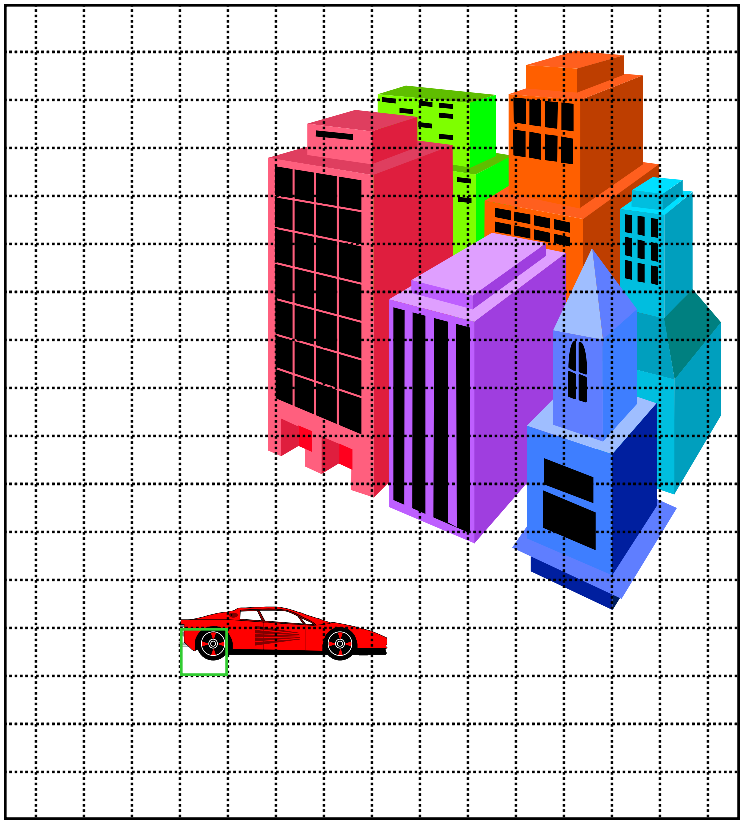


From [Wang02]

Block Matching Algorithm for Motion Estimation



Frame *t-1*
(Reference Frame)



Frame *t*
(Predicted frame)

Block Matching Algorithm Overview

- For each MB in a new (predicted) frame
  - Search for a block in a reference frame that has the lowest matching error
    - Using sum of absolute differences (SAD) between corresponding pels
    - Search range: depends on the anticipated motion range

$$E_{\mathrm{DFD}}(\mathbf{d}_{\mathrm{m}}) = \sum_{\mathbf{x} \in B_m} \left| \psi_2(\mathbf{x} + \mathbf{d}_m) - \psi_1(\mathbf{x}) \right|^p \rightarrow \min$$

  - Displacement between the current MB and the best matching MB is the MV
  - Current MB is replaced by the best matching MB (motion-compensated prediction or motion compensation)
- This subject will be discussed in more detail in a separate lecture

Temporal Prediction

- No Motion Compensation:
  - Work well in stationary regions

$$\hat{f}(t,m,n) = f(t-1,m,n)$$

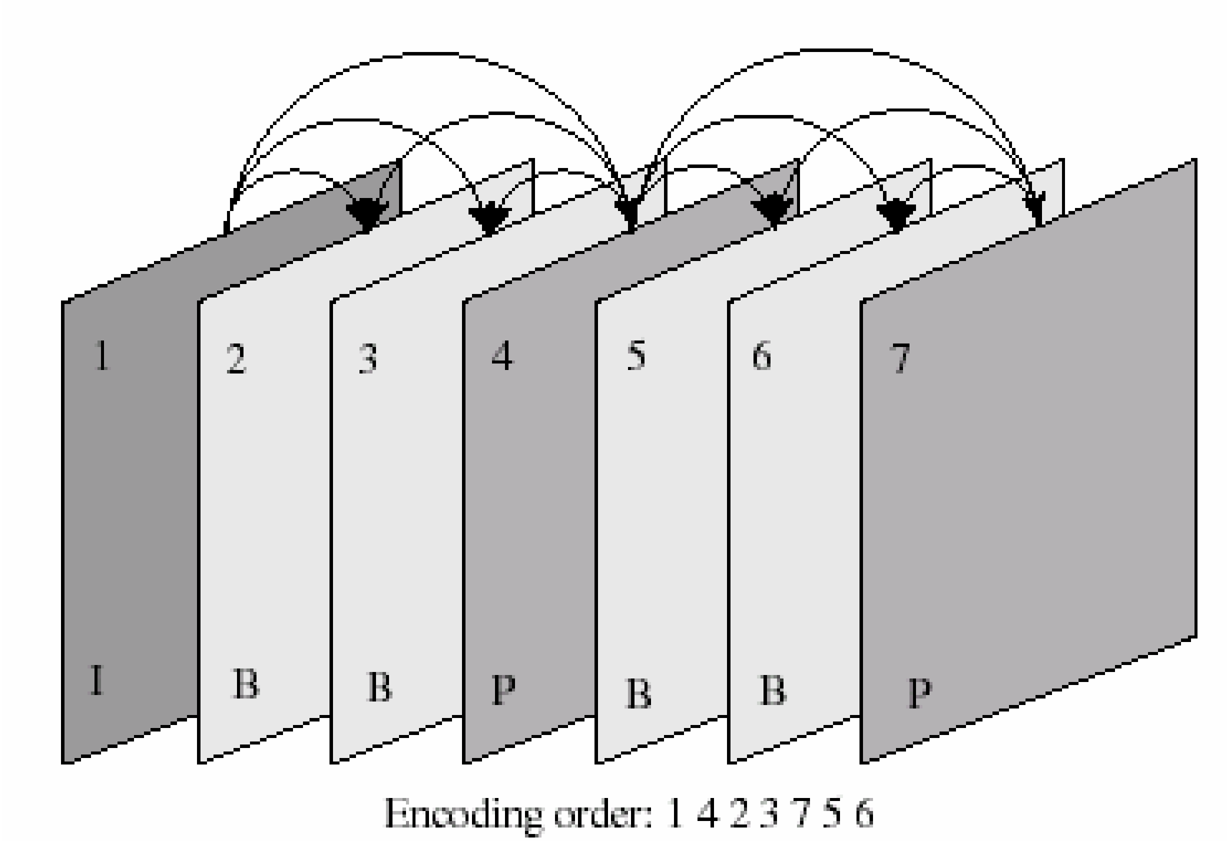
- Uni-directional Motion Compensation:
  - Does not work well for uncovered regions due to object motion or newly appeared objects

$$\hat{f}(t,m,n) = f(t-1,m-d_x,n-d_y)$$

- Bi-directional Motion Compensation
  - Can handle better covered/uncovered regions

$$\hat{f}(t,m,n) = w_b f(t-1,m-d_{b,x},n-d_{b,y}) + w_f f(t+1,m-d_{f,x},n-d_{f,y})$$

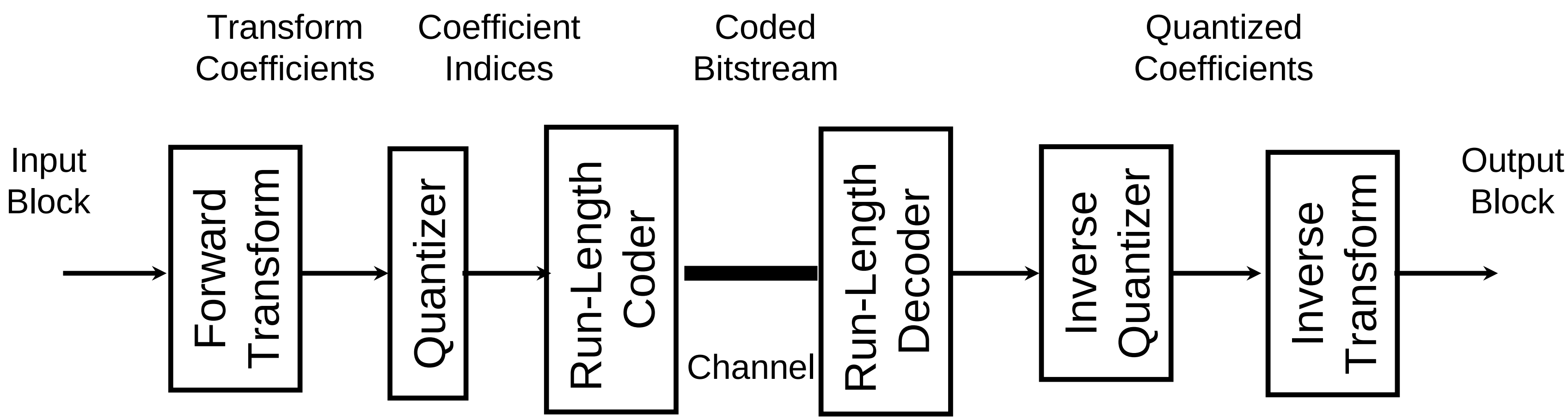
Different Coding Modes



Intra: coded directly; **P**redictive: predicted from a previous frame;
Bidirectional: predicted from a previous frame and a following frame.
Can be done at the block or frame level.

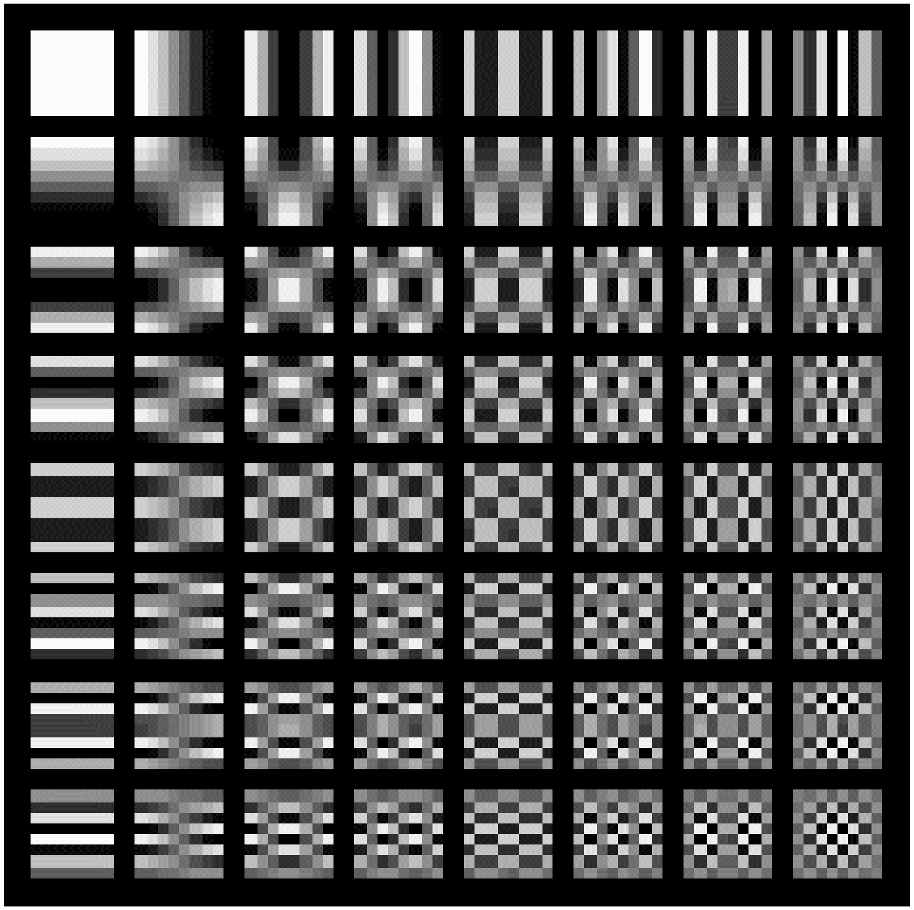
From [Wang02]

DCT-Based Coding Revisited



- Why do we use DCT:
  - To exploit the correlation between adjacent pixels
  - Typically only low frequency DCT coefficients are significant
- For I-blocks, DCT is applied to original image values
  - The new H.264 standard applies intra-prediction and DCT is applied to intra-prediction error
- For P/B-blocks, DCT is applied to prediction errors

Basis Images of 8x8 DCT



DCT on a Real Image Block

```
>>imblock = lena256(128:135,128:135)-128
imblock=
 54   68   71   73   75   73   71   45
 47   52   48   14   20   24   20   -8
 20  -10   -5  -13  -14  -21  -20  -21
-13  -18  -18  -16  -23  -19  -27  -28
-24  -22  -22  -26  -24  -33  -30  -23
-29  -13    3  -24  -10  -42  -41    5
-16   26   26  -21   12  -31  -40   23
 17   30   50   -5    4   12   10    5
```

```
>>dctblock =dct2(imblock)
dctblock=
 31.0000   51.7034    1.1673  -24.5837  -12.0000  -25.7508   11.9640   23.2873
113.5766    6.9743  -13.9045   43.2054   -6.0959   35.5931  -13.3692  -13.0005
195.5804   10.1395   -8.6657   -2.9380  -28.9833   -7.9396    0.8750    9.5585
 35.8733  -24.3038  -15.5776  -20.7924   11.6485  -19.1072   -8.5366    0.5125
 40.7500  -20.5573  -13.6629   17.0615  -14.2500   22.3828   -4.8940  -11.3606
  7.1918  -13.5722   -7.5971  -11.9452   18.2597  -16.2618   -1.4197   -3.5087
 -1.4562  -13.3225   -0.8750    1.3248   10.3817   16.0762    4.4157    1.1041
 -6.7720   -2.8384    4.1187    1.1118   10.5527   -2.7348   -3.2327    1.5799
```

Note that most DCT coefficients are close to zero except those at the low-low range

Quantization Matrices

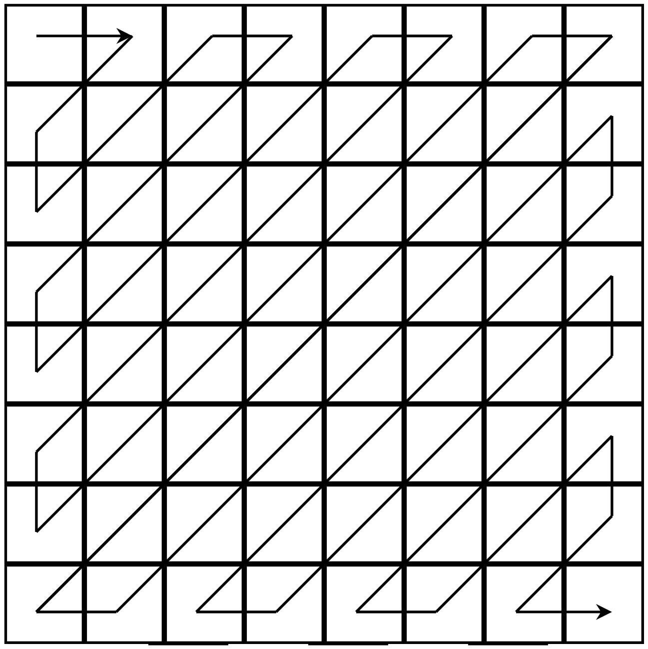
- For I-blocks: non-uniform scaling is used (as in JPEG)

$w_{u,v}$	0	1	2	3	4	5	6	8
0	8	16	19	22	26	27	29	34
1	16	16	22	24	27	29	34	37
2	19	22	26	27	29	34	34	38
3	22	22	26	27	29	34	37	40
4	22	26	27	29	32	35	40	48
5	26	27	29	32	35	40	48	58
6	26	27	29	34	38	46	56	69
7	27	29	35	38	46	56	69	83

Figure 13.13 Default weights for quantization of I-blocks in MPEG-1. Weights for horizontal and vertical frequencies differ.

- For P/B blocks: the same stepsize (8) is used for all coefficients, and this stepsize can be scaled by a user-selectable parameter (quantization parameter or QP) that controls the trade-off between bit-rate and quality

Zig-Zag Ordering



Zig-Zag ordering: converting a 2D matrix into a 1D array, so that the frequency (horizontal+vertical) increases in this order, and the coefficient variance (average of magnitude square) decreases in this order.

Run-length Coding

- Runlength coding
  - Many coefficients are zero after quantization
  - Runlength Representation:
    - Ordering coefficients in the zig-zag order
    - Specify how many zeros before a non-zero value
    - Each symbol=(length-of-zero, non-zero-value)
    - For I-blocks, the DC coefficient is specified directly
  - Code all possible symbols using Huffman coding
    - More frequently appearing symbols are given shorter codewords
    - One can use default Huffman tables or specify its own tables.
    - Instead of Huffman coding, arithmetic coding can be used to achieve higher coding efficiency at an added complexity.
  - More efficient entropy coding methods can be used
    - Context-based arithmetic coding

Example of Runlength Coding

Quantized DCT indices for an I block =

2	5	0	-2	0	-1	0	0
9	1	-1	2	0	1	0	0
14	1	-1	0	-1	0	0	0
3	-1	-1	-1	0	0	0	0
2	-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

Run-length symbol representation:
{2,(0,5),(0,9),(0,14),(0,1),(1,-2),(0,-1),(0,1),(0,3),(0,2),(0,-1),(0,-1),(0,2),(1,-1),(2,-1), (0,-1),(4,-1),(0,-1),(0,1),EOB}

EOB: End of block, one of the symbol that is assigned a short Huffman codeword

Macroblock Coding in I-Mode (without Intra Prediction)

DCT transform each 8x8 DCT block

↓

Quantize the DCT coefficients with properly chosen quantization matrices (different matrices for Y and C)

↓

The quantized DCT coefficients are zig-zag ordered and run-length coded

Macroblock Coding in I-Mode (with Intra Prediction)

Predicting current block from previously coded neighboring pixels in this frame using possibly several prediction modes, determine the prediction error

↓

DCT transform the prediction error

↓

Quantize the DCT coefficients with properly chosen quantization matrices (different matrices for Y and C)

↓

The quantized DCT coefficients are zig-zag ordered and run-length coded

Macroblock Coding in P-Mode

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

↓

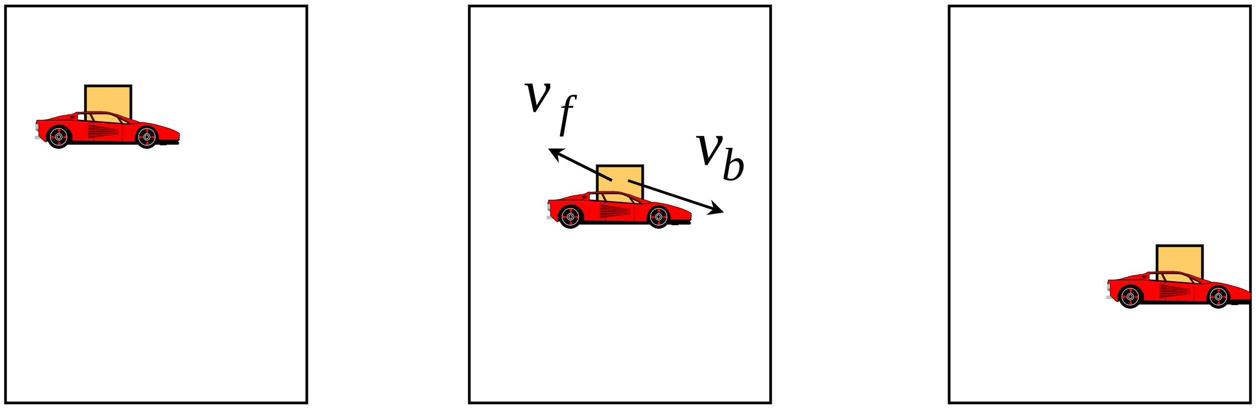
The prediction errors in each of the DCT blocks (8x8) are DCT transformed, quantized (according to specified QP), zig-zag scanned, and run-length coded

↓

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

Macroblock Coding in B-Mode

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



Coding Mode Selection

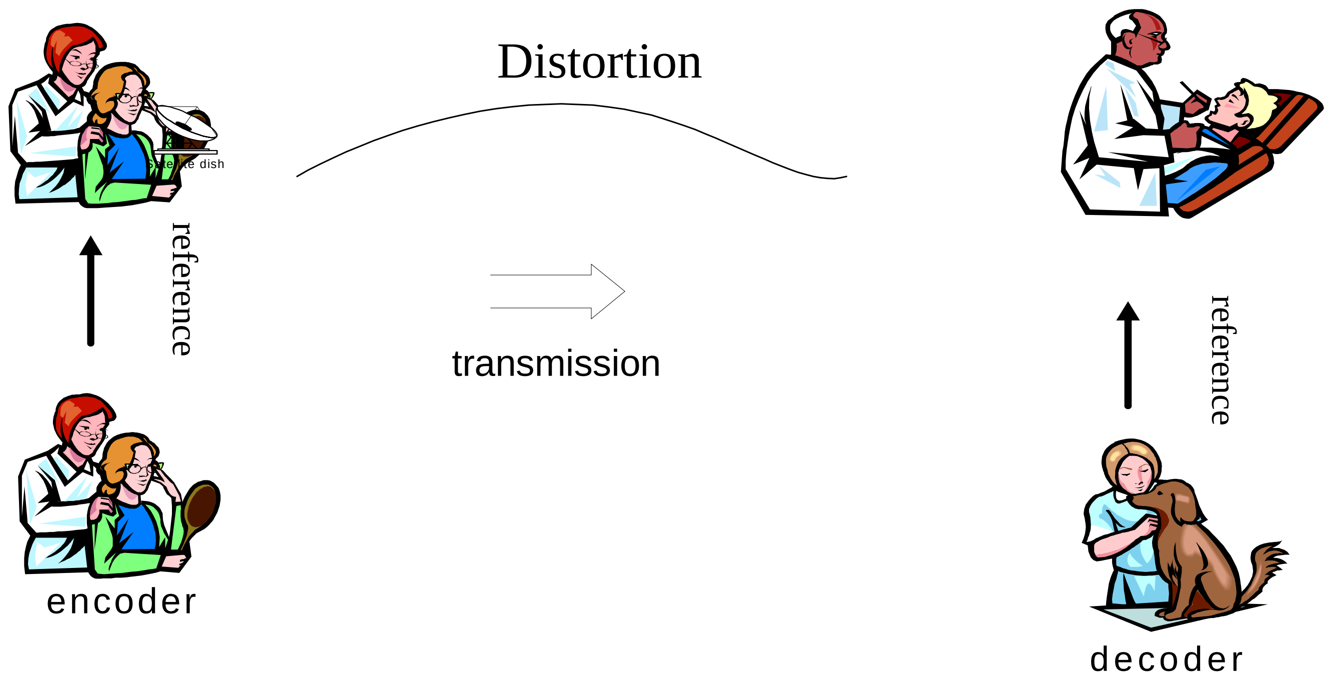
- Which mode should we use for a given MB?
- Frame-level control
  - I frame use only I-mode
  - P-frame use P-mode, except when prediction does not work (back to I-mode)
  - B-frame use B-mode (but can switch to P-mode and I-mode)
- Block-level control
  - A MB is coded using the mode that leads to the lowest bitrate for the same distortion -> rate-distortion optimized mode selection
  - 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

Rate Control

- For a fixed QP, the bit rate varies from block to block
  - I mode needs more bits than P and B modes
  - Even when the mode is the same, blocks with complex motion and texture require more bits
- To reach a desired bit rate (averaged over a frame or a group of frames), one can adjust
  - QP
  - Encoding frame rate (frame skip)
  - Controlled by the status of a buffer that stores the bits produced by the encoder

Sensitivity to Transmission Errors

- Prediction and variable length coding makes the video stream very sensitive to transmission errors on the bitstream or packet stream
  - Error in one frame will propagate to subsequent frames
  - Bit errors or packet losses in one part of the bit stream make the following bits undecodable



Effect of Transmission Errors

Coded, No loss

3%

5%

10%

Example reconstructed video frames from a H.263 coded sequence, subject to packet losses

Error Resilient Encoding

- To help the decoder to resume normal decoding after errors occur, the encoder can
  - Periodically insert INTRA mode (INTRA refresh)
  - Insert resynchronization codewords at the beginning of a group of blocks (GOB)
- More sophisticated error-resilience tools
  - Multiple description coding
- Trade-off between efficiency and error-resilience
- Can also use channel coding / retransmission to correct errors

Error Concealment

- With proper error-resilience tools, packet loss typically lead to the loss of an isolated segment of a frame
- The lost region can be "recovered" based on the received regions by spatial/temporal interpolation → Error concealment
- Decoders on the market differ in their error concealment capabilities

Without concealment With concealment

Scalable Coding

- Motivation
  - Real networks are heterogeneous in rate
    - streaming video from home (56 kbps) using modem vs. corporate LAN (10-100 mbps)
- Scalable video coding
  - Ideal goal (embedded stream): Creating a bitstream that can be accessed at any rate
  - Practical video coder:
    - layered coder: base layer provides basic quality, successive layers refine the quality incrementally
    - Coarse granularity (typically known as layered coder)
    - Fine granularity (FGS):
      - bit plane coding or wavelet-based coding

Bit Stream Scalability

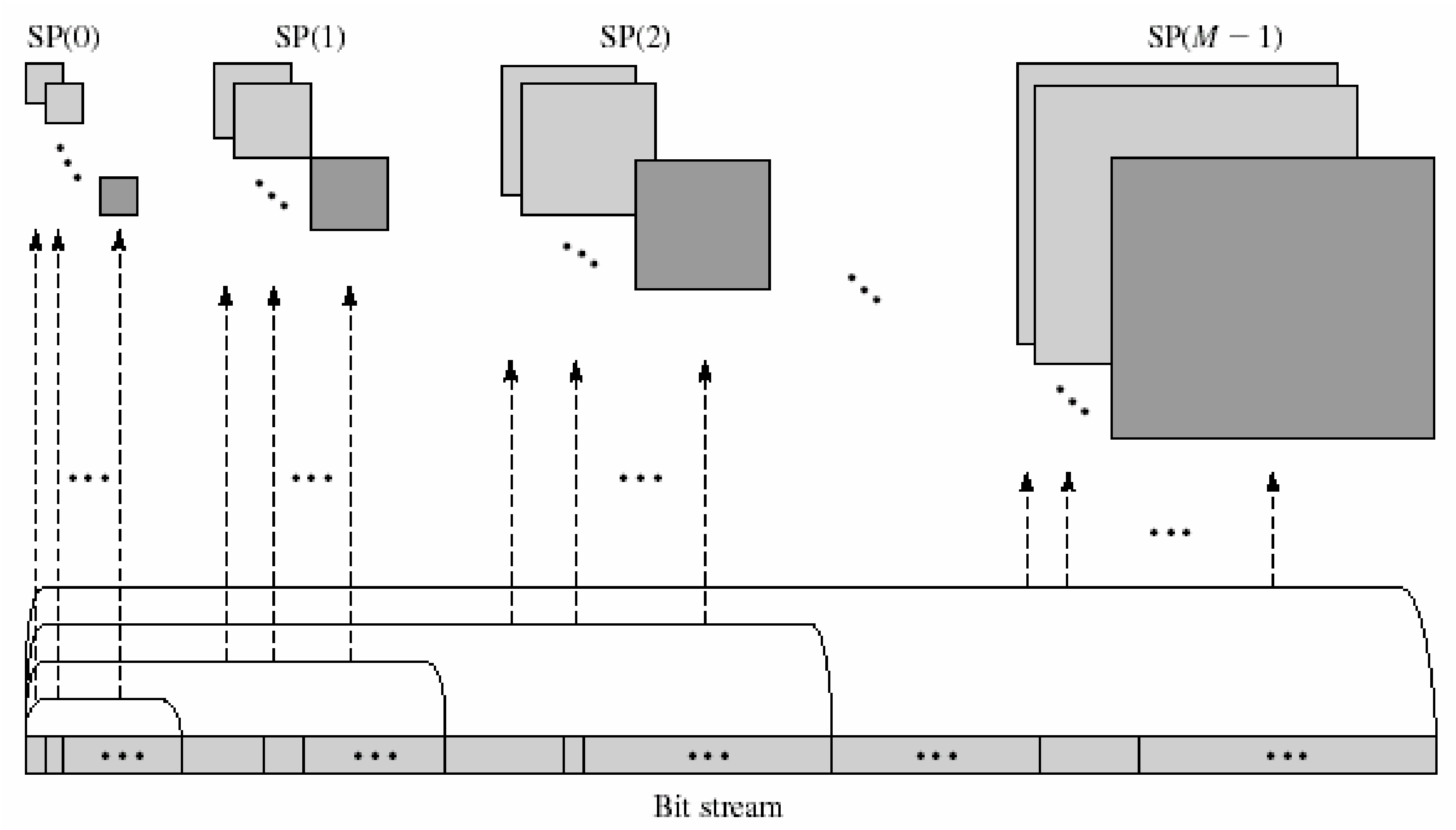
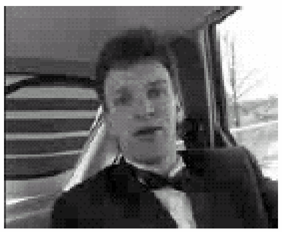
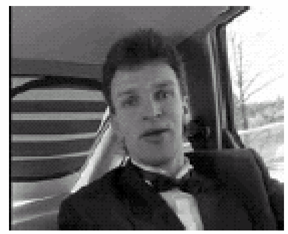


Illustration of Scalable Coding

Spatial scalability



6.5 kbps



133.9 kbps

21.6 kbps

436.3 kbps

Quality (SNR) scalability

What you should know

- What are the principle steps in a video coder? What are the three types of information coded? You should be able to draw the block diagram of a typical block-based video codec (encoder and decoder) using motion-compensation and know the function of each step
- Why do we use motion-compensated prediction?
- What are the difference between I, B, and P modes? Why do we use different modes? What may be the problem if we use P-modes only (except the first frame)?
- What are the basic steps in DCT-based coding? How to apply it to I and P/B blocks ?
- Why is error-resilience and error-concealment important in video encoder and decoder design?
- What is scalable coding? What are the benefits and trade-offs?

References

- Y. Wang, J. Ostermann, Y. Q. Zhang, *Video Processing and Communications,* Prentice Hall, 2002. Chapters 9,11,13
- Y. Wang and Q. Zhu, "Error control and concealment for video communication: a review," *Proceedings of the IEEE*, vol. 86, pp. 974-997. May 1998.