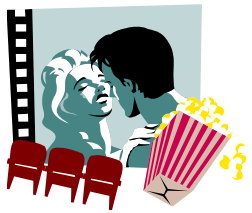
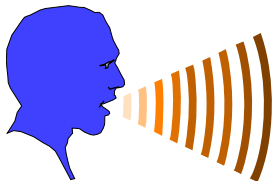





Video Processing & Communications

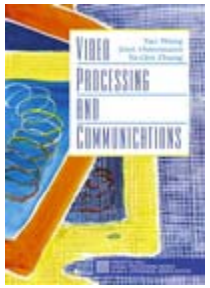
Video Coding Using Motion Compensation

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

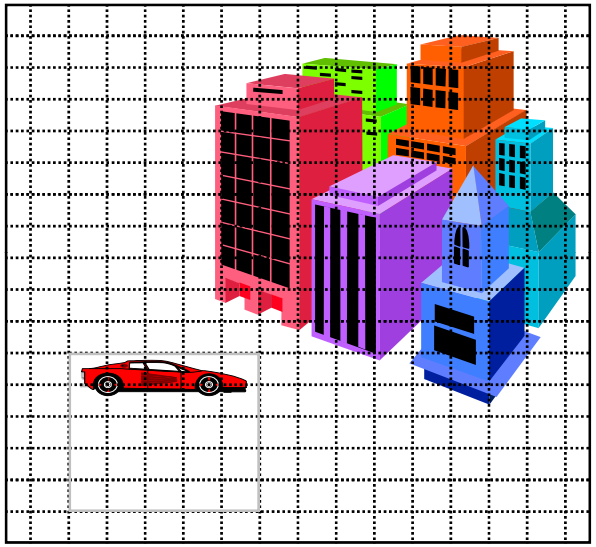
Based on: Y. Wang, J. Ostermann, and Y.-Q. Zhang, Video Processing and Communications, Prentice Hall, 2002.

Outline

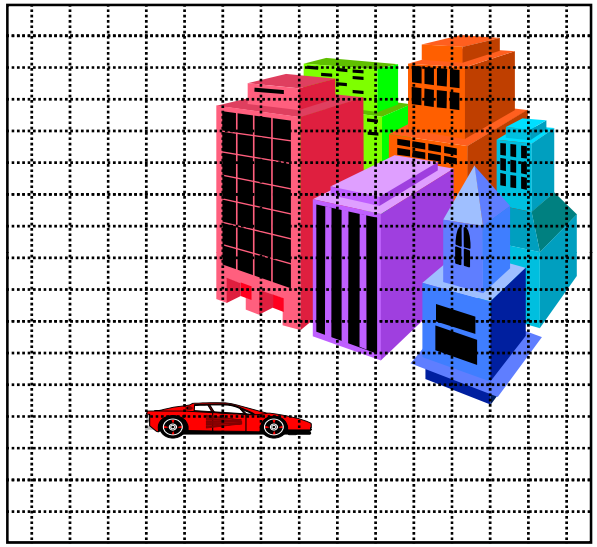
- Overview of Block-Based Hybrid Video Coding
- Coding mode selection and rate control
- Loop filtering



Characteristics of Typical Videos



Frame *t-1*



Frame *t*

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

Temporal Prediction

- No Motion Compensation (zero motion):
  - Work well in stationary regions

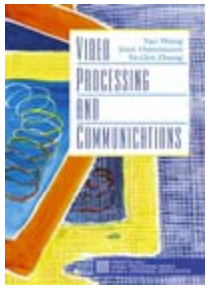
$$\hat{f}(t,m,n) = f(t-1,m,n)$$

- Uni-directional Motion Compensation:
  - Does not work well for uncovered regions by object motion

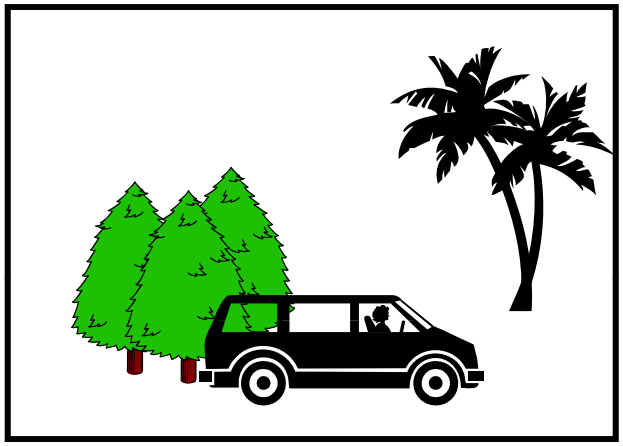
$$\hat{f}(t,m,n) = f(t-1,m-d_x,n-d_y)$$

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

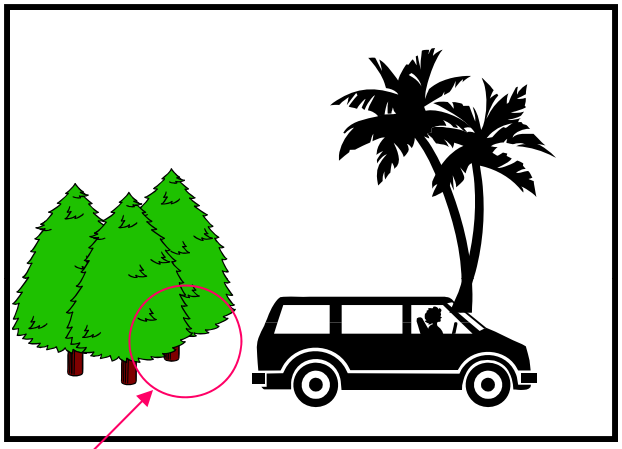
$$\hat{f}(t,m,n) = w_b f(t-1,m-d_{b,x},n-d_{b,y}) + w_f f(t+1,m-d_{f,x},n-d_{f,y})$$



Uni-Directional Temporal Prediction



Past frame



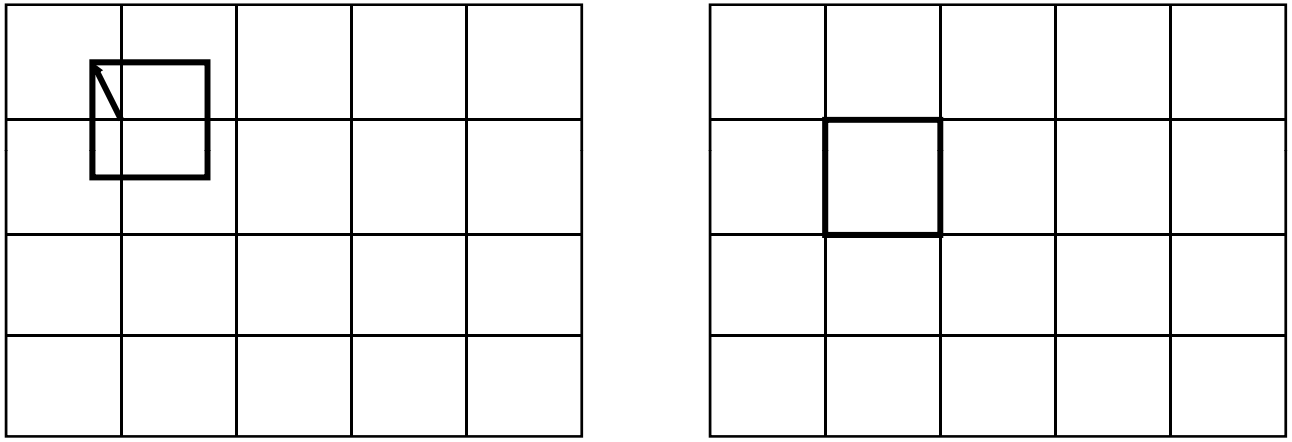
Current frame

All objects **except** this area have already been sent to decoder in “past frame”

Motion Compensated Prediction

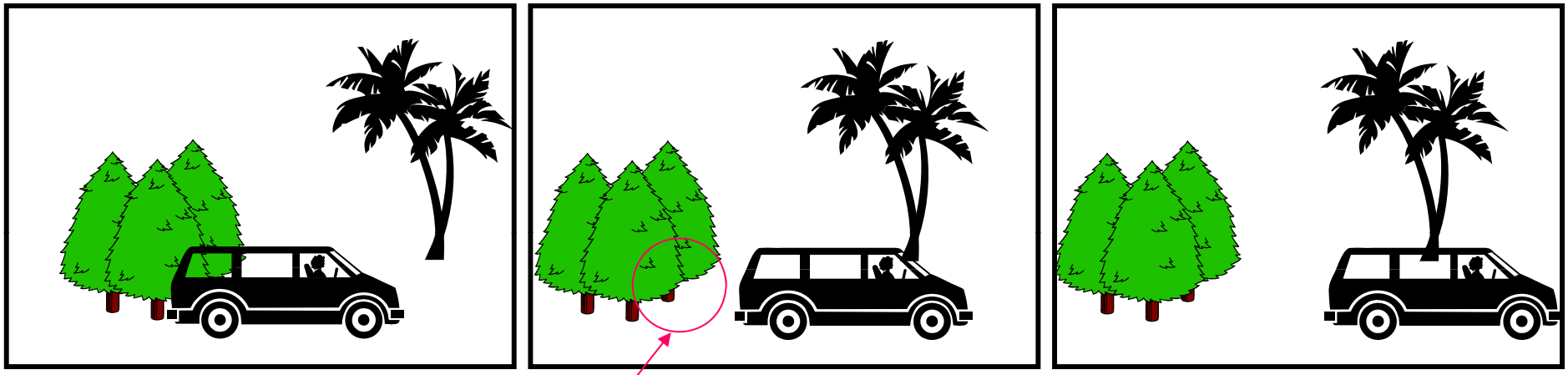
Past frame

Current frame



- Assumes rigid bodies move translationally; uniform illumination; no occlusion, no uncovered objects
- Big win: Improves compression by factor of 5-10

Bi-directional Prediction



Past frame

Current frame

Future frame

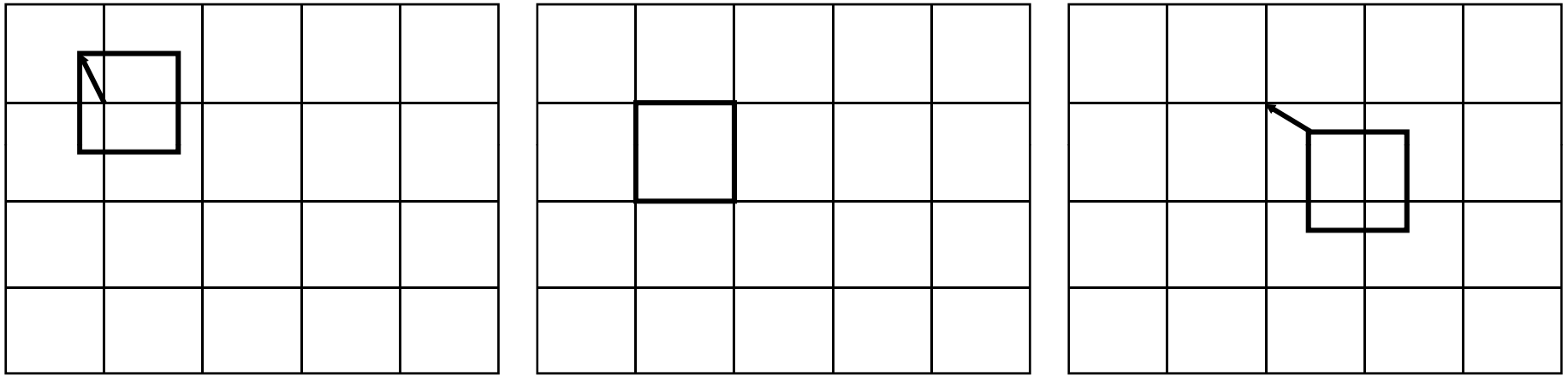
This area can now be predicted
using “future frame”

Motion Compensated Bidirectional Prediction

Past frame

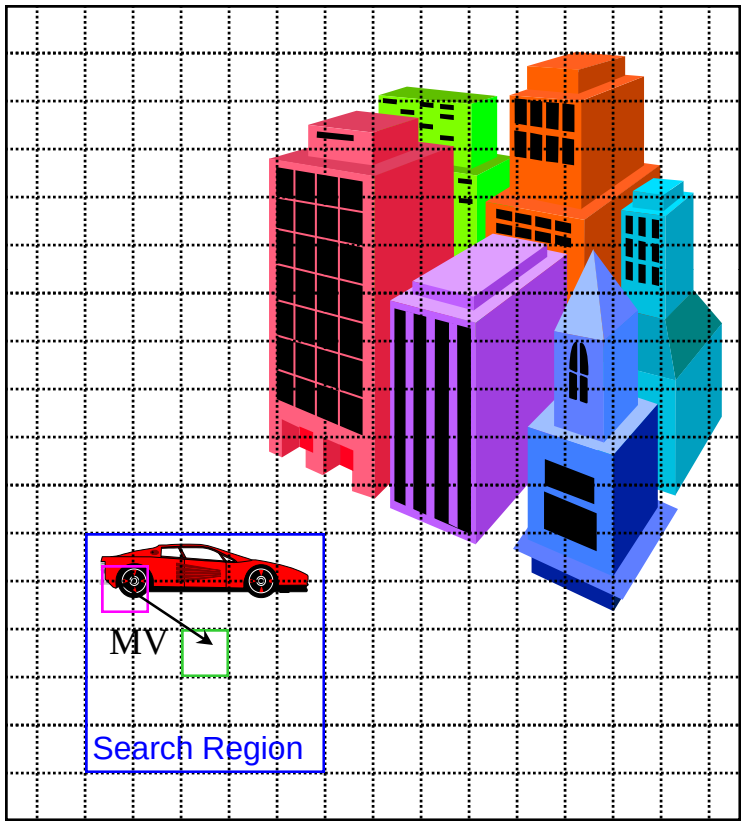
Current frame

Future frame

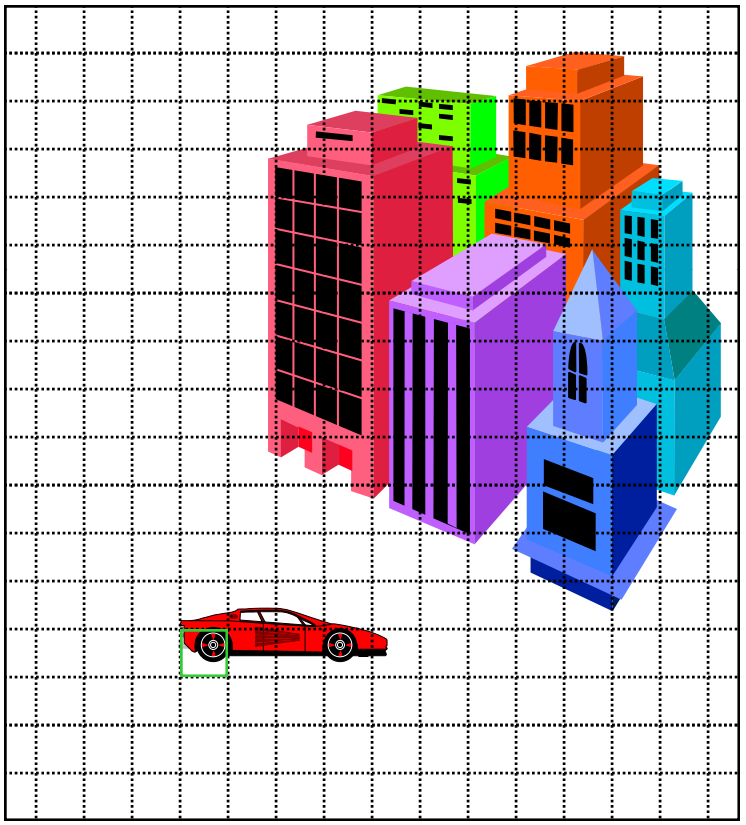


- Helps when there is occlusion or uncovered objects
- Vector into the future need not be the same as vector into the past

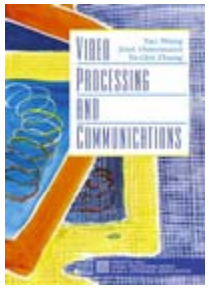
Block Matching Algorithm for Motion Estimation



Frame *t-1*
(Reference Frame)



Frame *t*
(Predicted frame)



Multiple Reference Frame Temporal Prediction

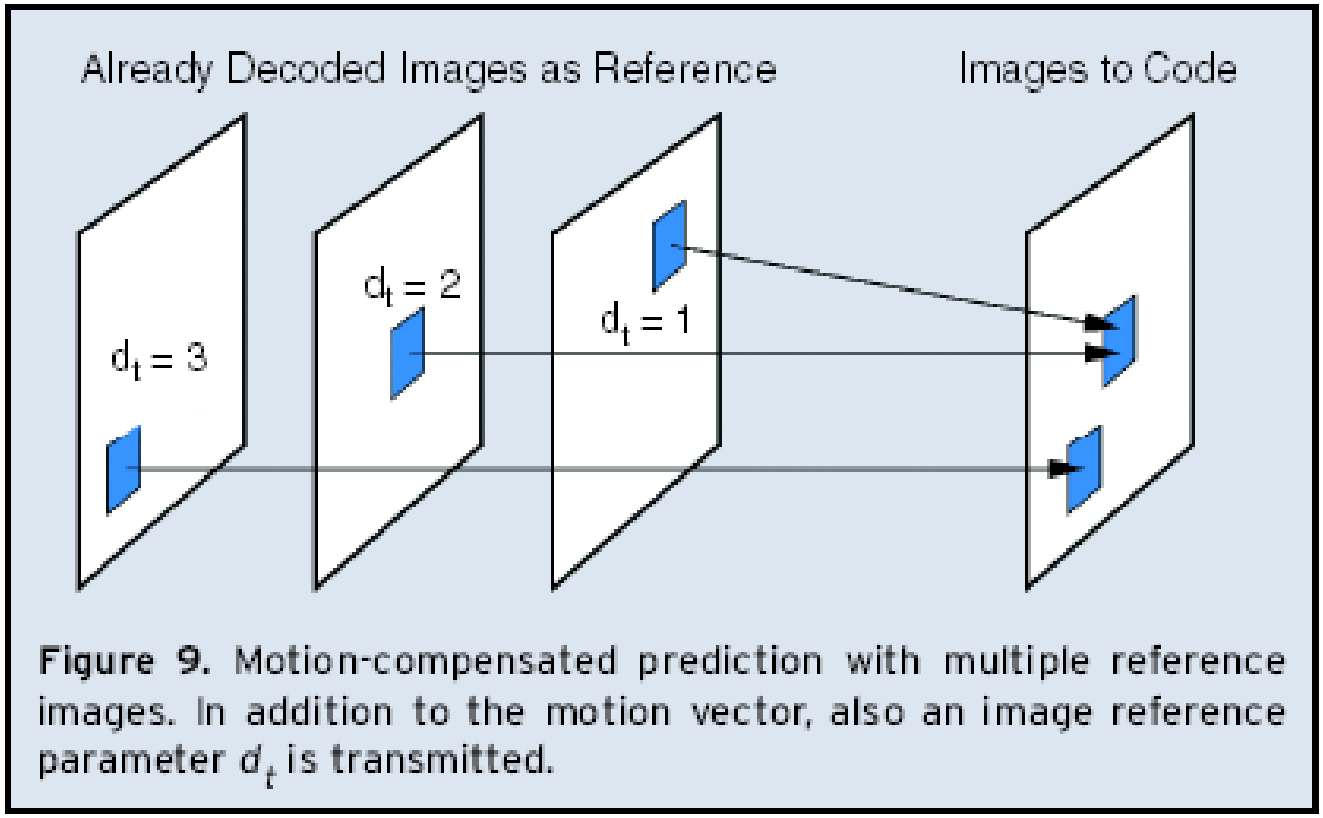


Figure 9. Motion-compensated prediction with multiple reference images. In addition to the motion vector, also an image reference parameter d_t is transmitted.

When multiple references are combined, the best weighting coefficients can be determined using ideas similar to minimal mean square error predictor

Spatial Prediction

- General idea:
  - A pixel in the new block is predicted from previously coded pixels in the same frame
  - What neighbors to use?
  - What weighting coefficients to use?
- Content-adaptive prediction
  - No edges: use all neighbors
  - With edges: use neighbors along the same direction
  - The best possible prediction pattern can be chosen from a set of candidates, similar to search for best matching block for inter-prediction
    - H.264 has many possible intra-prediction pattern

H.264 Intra-Prediction

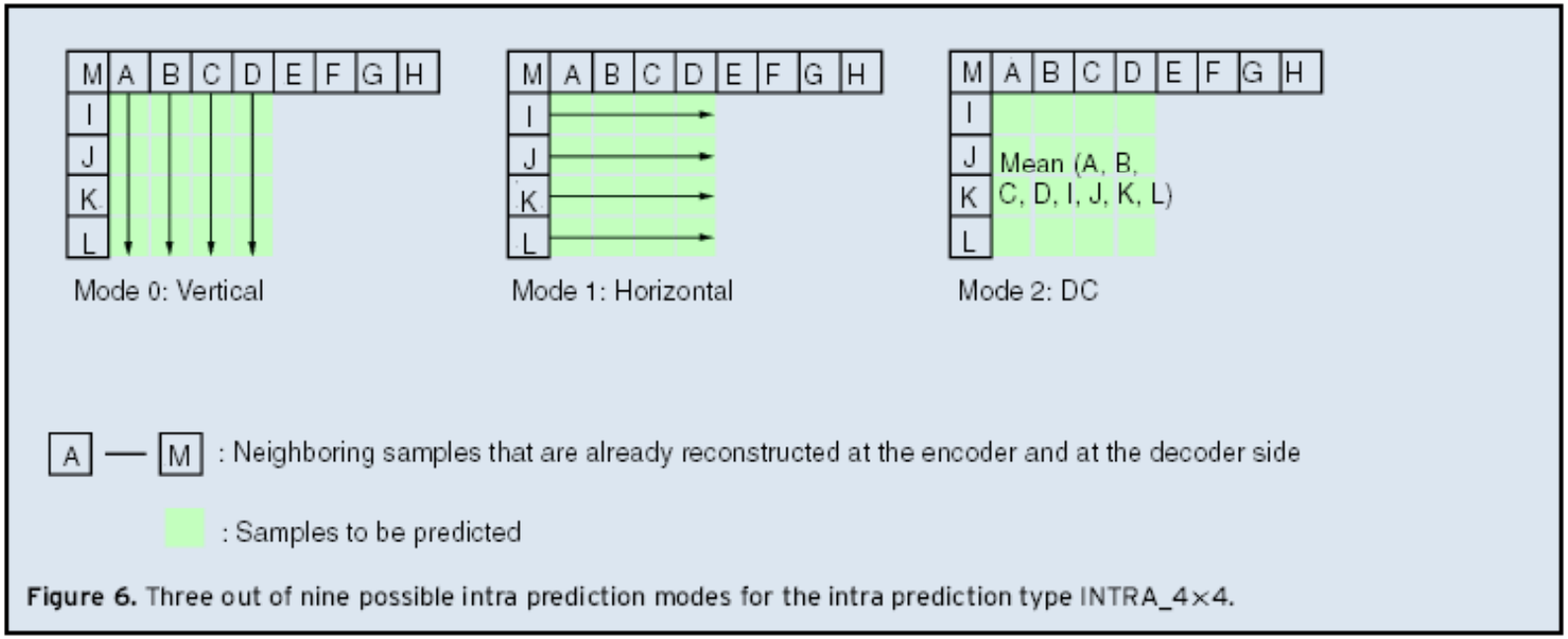


Figure 6. Three out of nine possible intra prediction modes for the intra prediction type INTRA_4×4.

From: Ostermann et al., Video coding with H.264/AVC: Tools, performance, and complexity, IEEE Circuits and Systems Magazine, First Quarter, 2004

Coding of Prediction Error Blocks

- Error blocks typically still have spatial correlation
- To exploit this correlation:
  - Vector quantization
  - Transform coding
- Vector quantization
  - Can effectively exploit the typically error patterns due to motion estimation error
  - Computationally expensive, requires training
- Transform coding
  - Can work with a larger block under the same complexity constraint
  - Which transform to use?

Transform Coding of Error Blocks

- Theory: Karhunen Loeve Transform is best possible block-based transform
- Problems with theory:
  - Finding an accurate model (covariance matrix) of the source is difficult
  - Model and KLT change over time and in different regions
  - Decoder and encoder need to use same KLT
  - Implementation complexity: a full matrix multiply is necessary to implement KLT
- Practice: Discrete Cosine Transform
  - When the the inter-pixel correlation approaches one, the KLT approaches the DCT

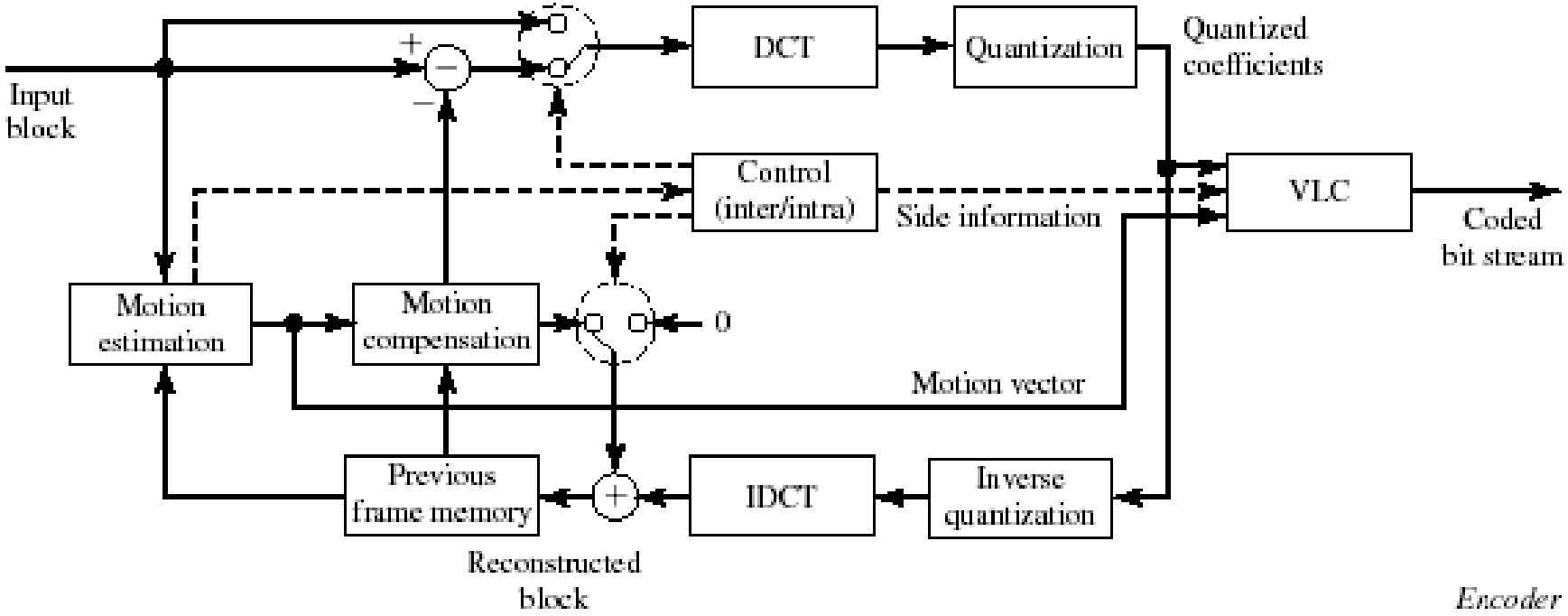
Transform Coding: What block size?

- Theory: Larger transform blocks (using more pixels) are more efficient
- Problem with theory:
  - Hard to get an accurate model of the correlation of distant pixels
  - In the limit as the inter-pixel correlation approaches one, the KLT approaches the DCT; however, the inter-pixel correlation of distant pixels is not close to one
- Practice:
  - Small block transforms – usually 8x8 pixels, although in more recent systems we can use 4x4 blocks or 16x16 blocks

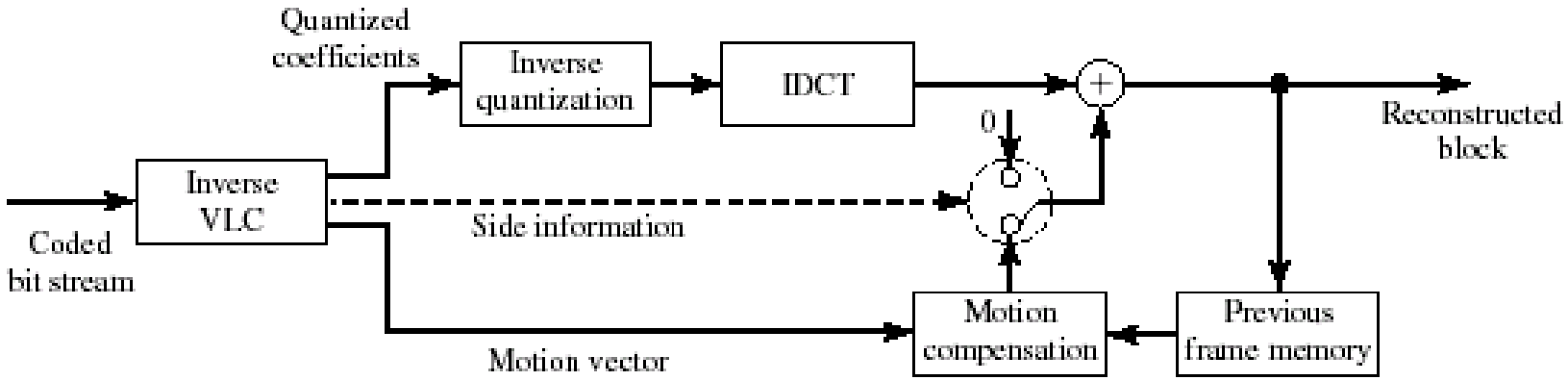
Key Idea in Video Compression

- Predict a new frame from a previous frame and only code the prediction error --- Inter prediction
- Predict a current block from previously coded blocks in the same frame --- Intra prediction (introduced in the latest standard H.264)
- Prediction error will be coded using the DCT method
- 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 --- Intra coding without intra-prediction
- 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)

Encoder Block Diagram of a Typical Block-Based Video Coder (Assuming No Intra Prediction)

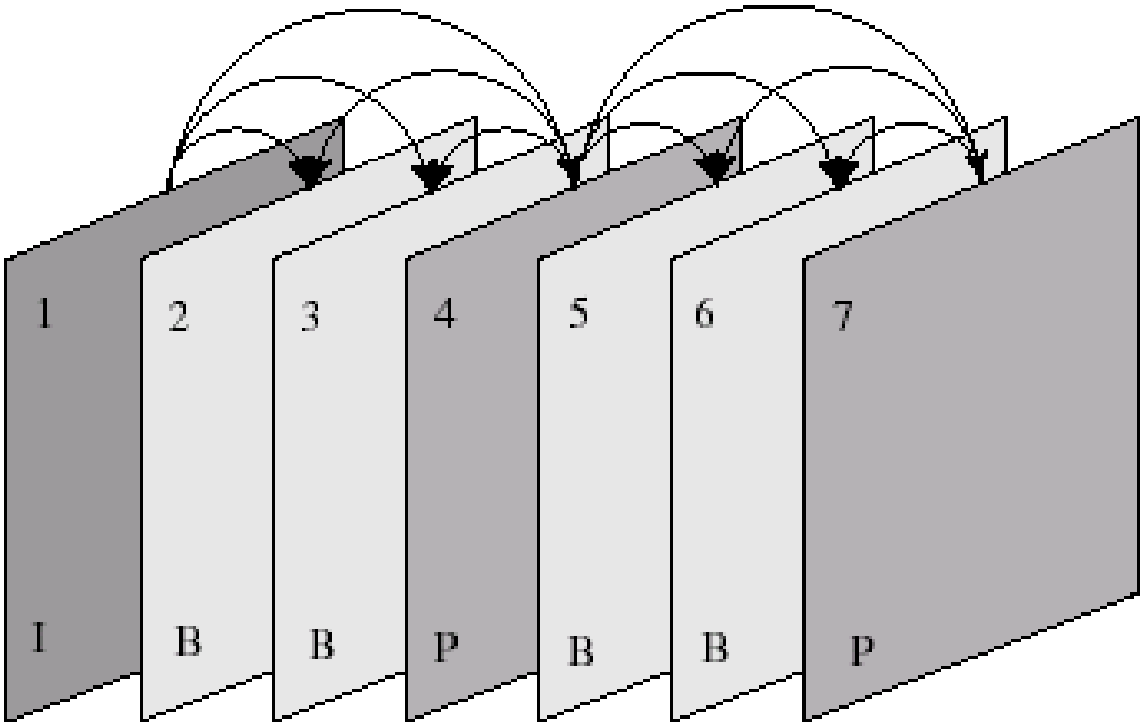


Decoder Block Diagram



Decoder

Group of Picture Structure

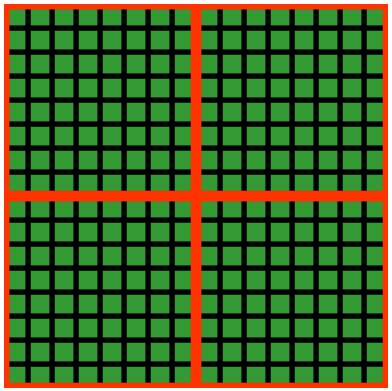


Encoding order: 1 4 2 3 7 5 6

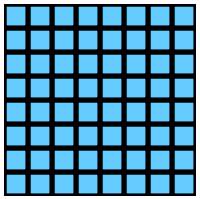
Group-of-picture structure

- I-frames coded without reference to other frames
  - To enable random access (AKA channel change), fast forward, stopping error propagation
- P-frames coded with reference to previous frames
- B-frames coded with reference to previous and future frames
  - Requires extra delay!
  - Enable frame skip at receiver (temporal scalability)
- *Typically*, an I-frame every 15 frames (0.5 seconds)
- *Typically*, two B frames between each P frame
  - Compromise between compression and delay

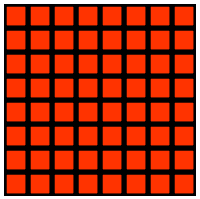
MB Structure in 4:2:0 Color Format



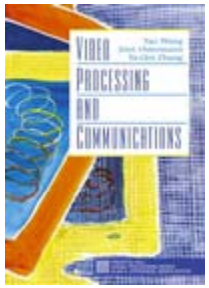
4 8x8 Y blocks



1 8x8 Cb blocks



1 8x8 Cr blocks



Macroblock Coding in I-Mode (assuming no intra-prediction)

DCT transform each 8x8 DCT block

↓

Quantize the DCT coefficients with properly chosen quantization matrices

↓

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

With intra-prediction, after the best intra-prediction pattern is found, the prediction error block is coded using DCT as above.

Macroblock Coding in P-Mode

Estimate one MV for each macroblock (16x16)

↓

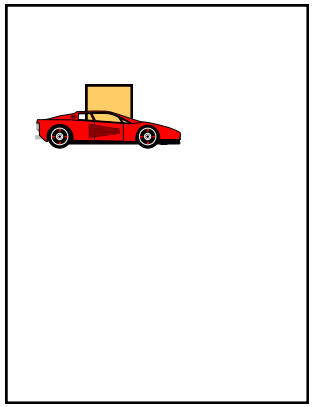
Depending on the motion compensation error, determine the coding mode (intra, inter-with-no-MC, inter-with-MC, etc.)

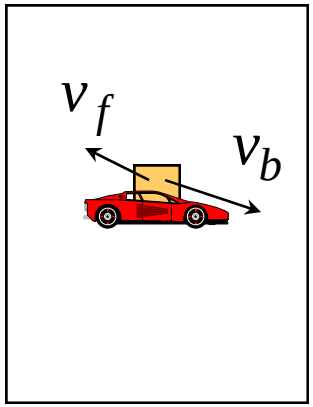
↓

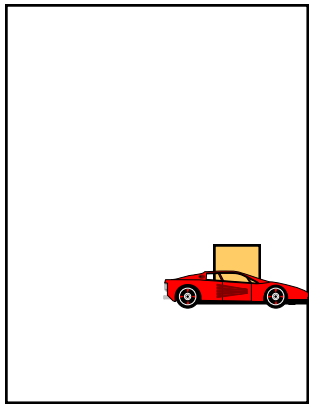
The original values (for intra mode) or motion compensation errors (for inter mode) in each of the DCT blocks (8x8) are DCT transformed, quantized, zig-zag/alternate scanned, and run-length coded

Macroblock Coding in B-Mode

- Same as for the P-mode, except a macroblock can be predicted from a previous picture, a following one, or both.



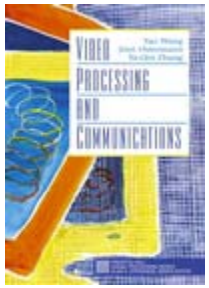




Operational control of a video coder

- Typical video sequences contain varying content and motion
- Different content is compressed well with different techniques
- Encoders should match coding techniques to content

- Coding parameters
  - Macroblock type (coding mode)
  - Motion vectors
  - Quantizer step size
- Each leads to different rate and distortion



Coding Mode Selection

- Coding modes:
  - Intra vs. inter, QP to use, for each MB, each leading to different rate
- Rate-distortion optimized selection, given target rate:
  - Minimize the distortion, subject to the target rate constraint

$$\text{minimize} \quad \sum_n D_n(m_n),$$

$$\text{subject to} \quad \sum_n R_n(m_k, \forall k) \le R_d.$$

$$\text{minimize} \quad J(m_n, \forall n) = \sum_n D_n(m_n) + \lambda \sum_n R_n(m_k, \forall k)$$

Simplified version $$J_n(m_n) = D_n(m_n) + \lambda R_n(m_n).$$

The optimal mode is chosen by coding the block with all candidates modes and taking the mode that yields the least cost.

Note that one can think of each candidate MV (and reference frame) as a possible mode, and determine the optimal MV (and reference frame) using this frame work --- Rate-distortion optimized motion estimation.

Rate Control: Why

- The coding method necessarily yields variable bit rate
- Active areas (with complex motion and/or complex texture) are hard to predict and requires more bits under the same QP
- Rate control is necessary when the video is to be sent over a constant bit rate (CBR) channel, where the rate when averaged over a short period should be constant
- The fluctuation within the period can be smoothed by a buffer at the encoder output
  - Encoded bits (variable rates) are put into a buffer, and then drained at a constant rate
  - The encoder parameter (QP, frame rate) need to be adjusted so that the buffer does not overflow or underflow

Rate Control: How

- General ideas:
  - Step 1) Determine the target rate at the frame or GOB level, based on the current buffer fullness
  - Step 2) Satisfy the target rate by varying frame rate (skip frames when necessary) and QP
    - Determination of QP requires an accurate model relating rate with Q (quantization stepsize)
    - Model used in MPEG2: R ~ A/Q + B/Q^2
- A very complex problem in practice

Loop Filtering

- Errors in previously reconstructed frames (mainly blocking artifacts) accumulate over time with motion compensated temporal prediction
  - Reduce prediction accuracy
  - Increase bit rate for coding new frames
- Loop filtering:
  - Filter the reference frame before using it for prediction
  - Can be embedded in the motion compensation loop
    - Half-pel motion compensation
    - OBMC
  - Explicit deblocking filtering: removing blocking artifacts after decoding each frame
- Loop filtering can significantly improve coding efficiency
- Theoretically optimal design of loop filters:
  - See text

Homework

- Reading assignment: Sec. 9.3
- (sec. 9.3.2 on OBMC optional)
- Computer assignment
  - Prob. 9.11, 9.12
  - Optional: 9.15