ECE 634: Digital Video Systems
Video coding: 2/28/17

Professor Amy Reibman
MSEE 356
reibman@purdue.edu

http://engineering.purdue.edu/~reibman/ece634/index.html

Next few lectures

- “Generic” video coding
- Standardization, standards, and their evolution

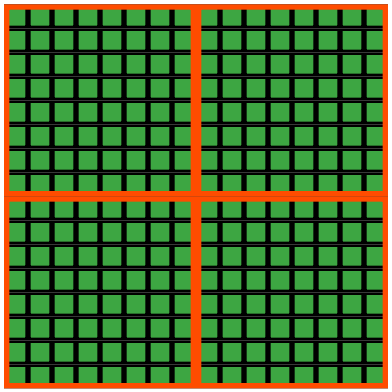
Outline

- Block-Based Hybrid Video Coding
  - Overview: putting previous lectures together
    - Representation; temporal prediction; spatial prediction; transform coding; quantization; variable bit-rate compression
  - Coding mode selection and rate control
  - Rate-distortion optimization
  - Loop filtering

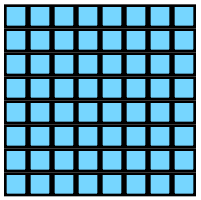
Key Ideas in Video Compression

- Prediction errors have smaller energy and can be coded with fewer bits
  - Predict new frame from "previous" frames --- Inter prediction
  - Predict current block from previous blocks in the same frame --- Intra prediction
- Prediction error is coded using Transform coding
- When prediction fails, don't use it!
  - Regions that cannot be predicted well are coded directly
- Work on each macroblock (MB) independently
  - Motion compensation done at the MB level
  - DCT coding of error at the block level (8x8 pixels)

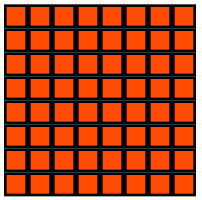
Representation: Macroblock (MB) Structure



4 8x8 Y blocks



1 8x8 Cb blocks



1 8x8 Cr blocks

4:2:0 Color Format

In HEVC: Coding Tree Unit is a generalization of a macroblock (square, up to 64*64 pixels) (more later)

Temporal compression

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

- In H.261 to H.264/AVC, motion compensation occurs at the Macroblock (16*16) level
- In H.265/HEVC, motion compensation occurs at the "Prediction Unit" (variable size)

Temporal compression: Theory vs. Practice (1)

- Theory: $\hat{f}_t = \alpha \hat{f}_{t-1}$ where α is chosen to obtain the best prediction that minimizes the expected error

- Problems with theory:
  - Finding the best α is difficult
  - α changes over time
  - Decoder and encoder need to use the same α
  - Implementation complexity: α should be limited to be some function of a power of two

Temporal compression: Theory vs. Practice (2)

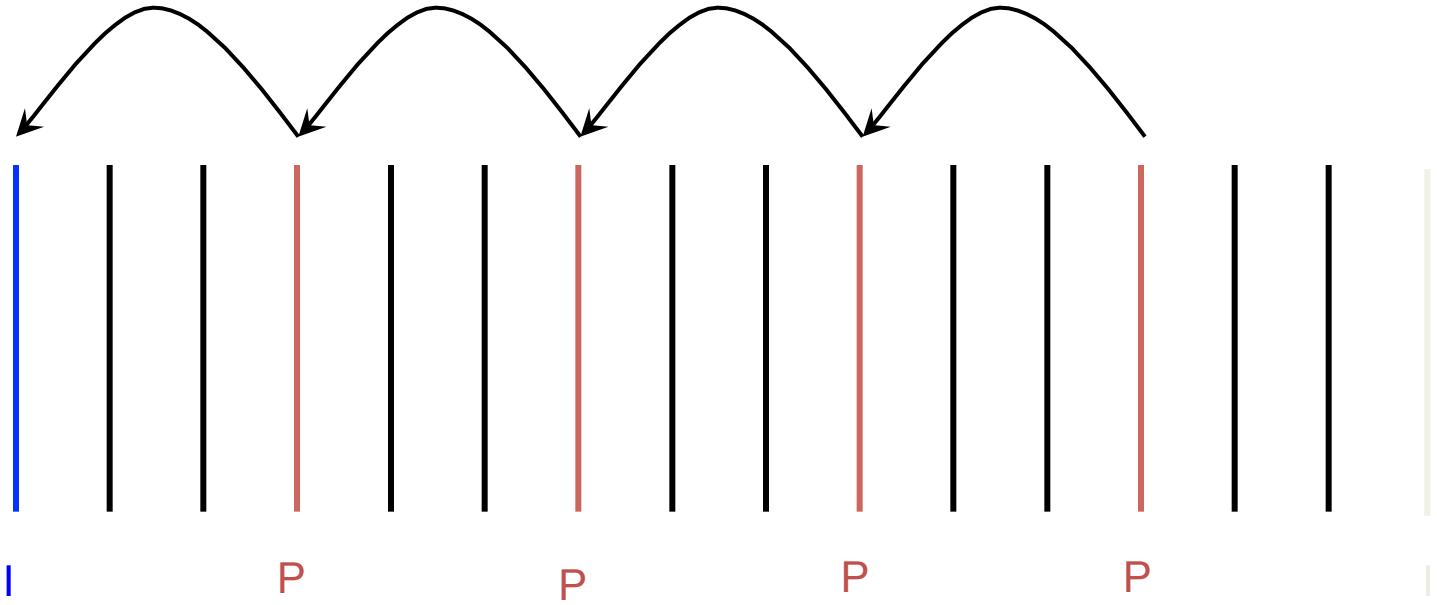
- More practical requirements
  - Every pixel must be predicted
  - Sometimes prediction works well; other times it does not
  - Some pixels are well predicted from a past frame
  - Some pixels are well predicted from a future frame

- Result: block-based motion compensation

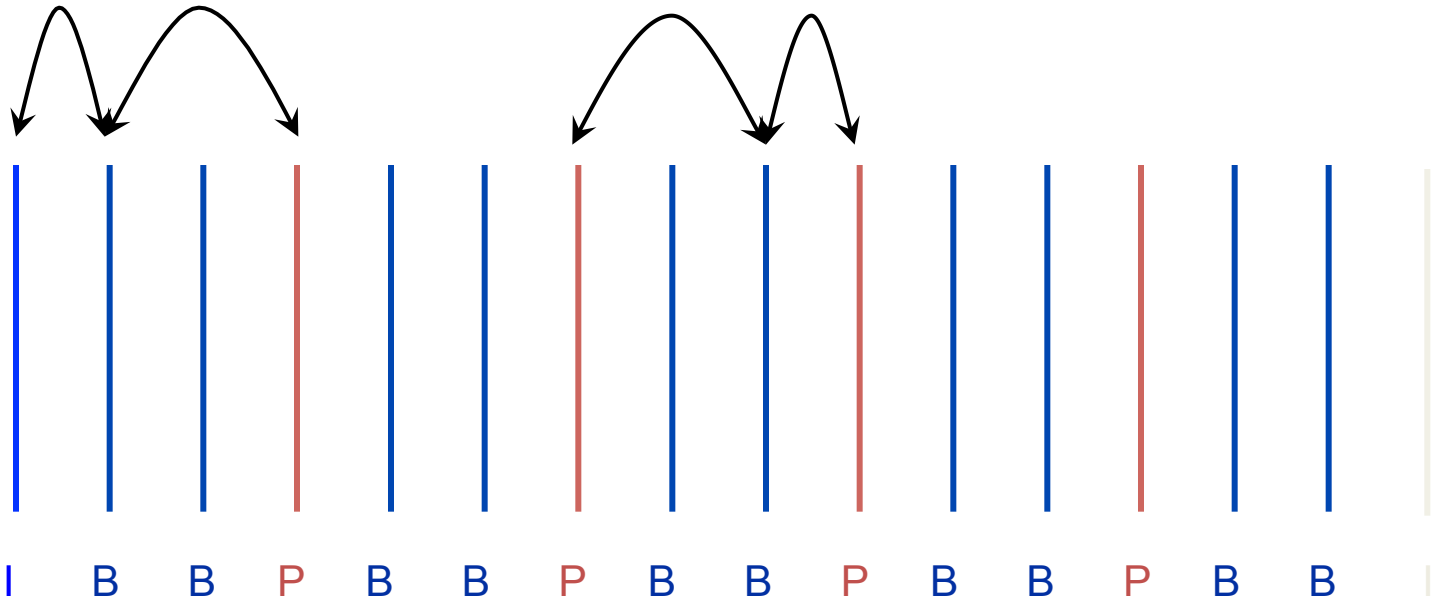
$$\hat{f}_t = (\hat{f}_{t-1} + \hat{f}_{t+1})/2$$

- Signal to decoder which option is used *for each block*
- Divide by 2 is a simple shift-right
- (Some options for more general weights in later standards)

Group of Pictures



Group of Pictures



Display order:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bitstream order:	0	2	3	1	5	6	4	8	9	7	11	12	10	14	15	13

Bitstream order:	0	3	1	2	6	4	5	9	7	8	12	10	11

Group-of-picture structure

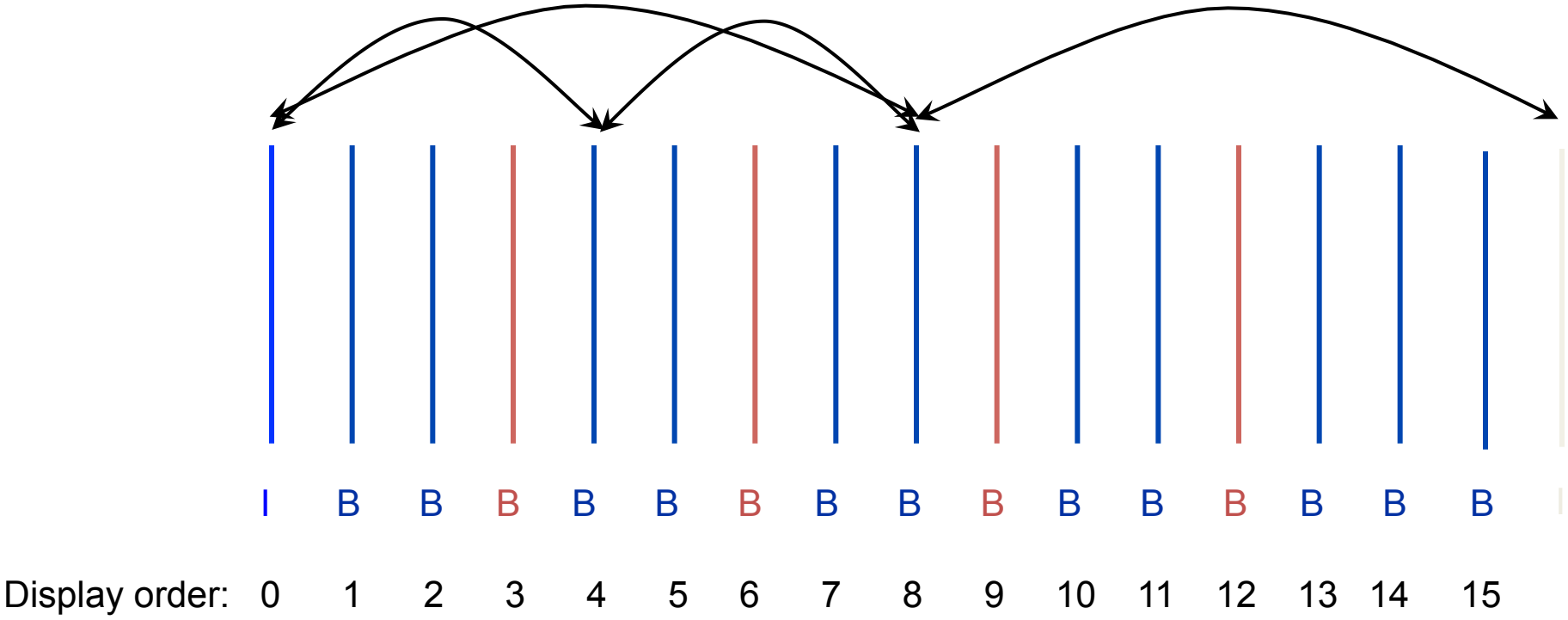
- I-frames coded without reference to other frames
- P-frames coded with reference to previous frames
- B-frames coded with reference to previous and future frames
  - Requires extra delay!

- *Typically*, an I-frame every 15 frames (0.5 seconds)
  - Fast random access (AKA channel change)

- *Typically*, two B frames between each P frame
  - Compromise between compression and delay

Hierarchical temporal prediction

In H.264 and beyond, B frames can be used for prediction



Block-based Temporal Prediction

- No Motion Compensation
  - Works well in stationary regions

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

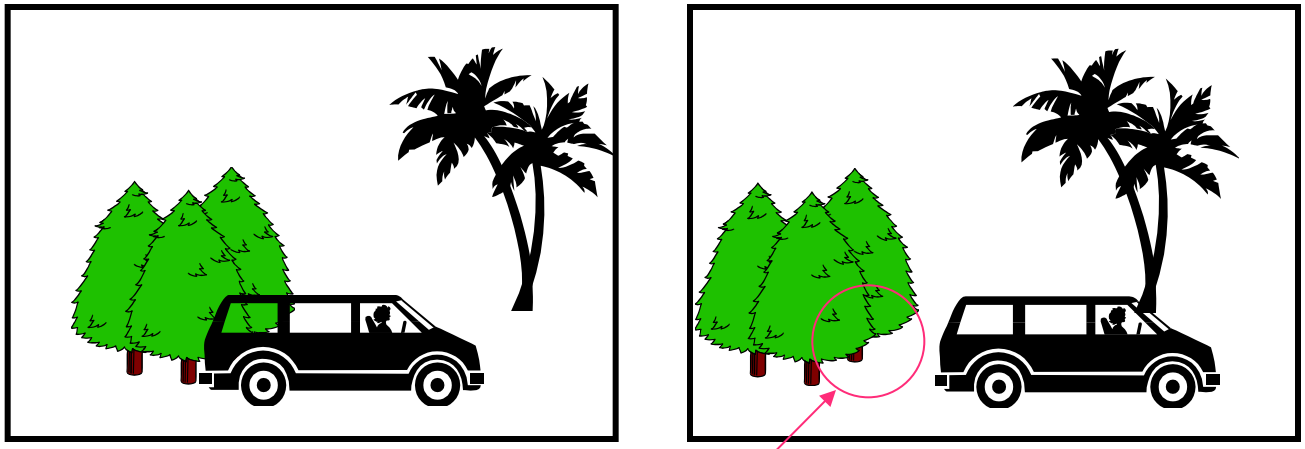
- Uni-directional Motion Compensation
  - Does not work well for uncovered regions by object motion

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

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

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

MPEG-2: Motion Compensation

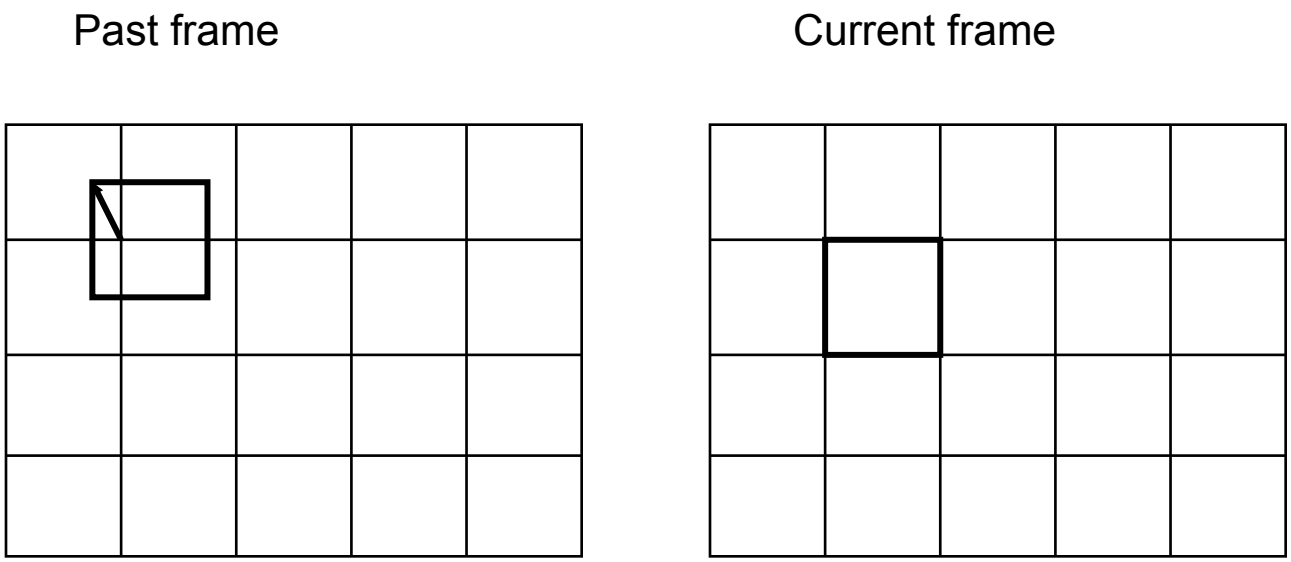


Past frame

Current frame

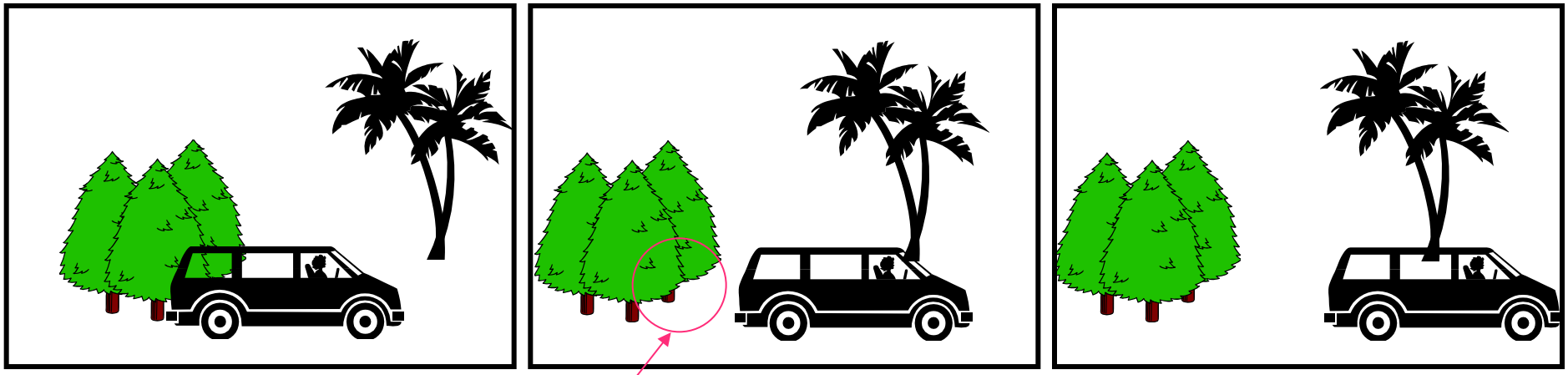
All objects **except** this area have already been sent to decoder in “past frame”

Motion Compensated Prediction (P-frame)



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

MPEG-2: Motion Compensation



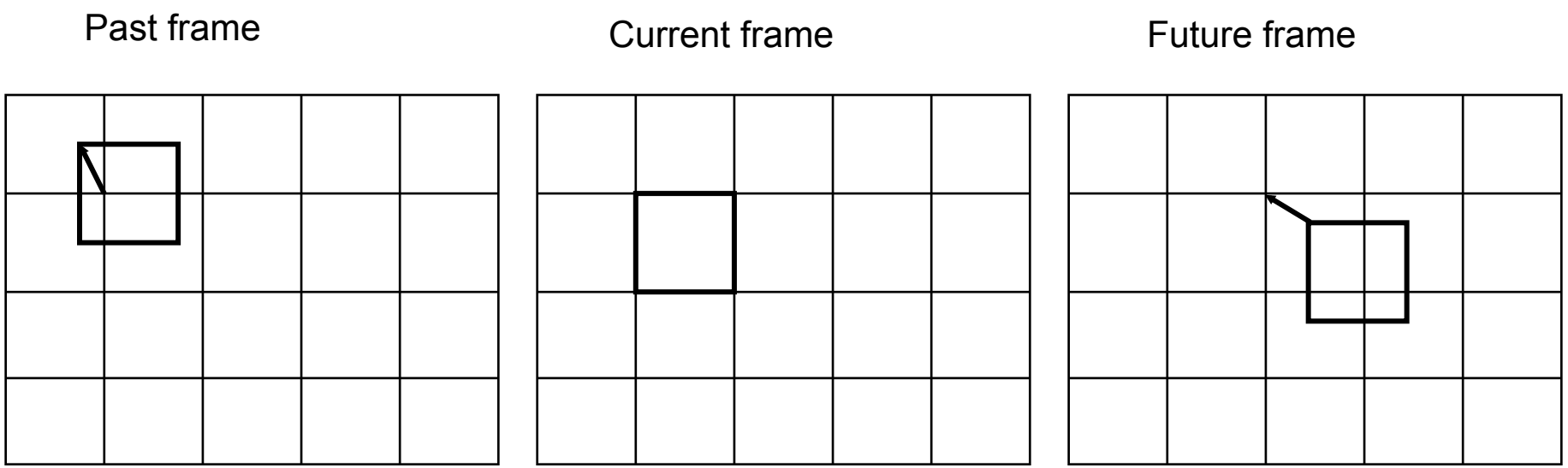
Past frame

Current frame

Future frame

This area can now be predicted using “future frame”

Motion Compensated Prediction (B-frame)



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

Multiple Reference Frame Temporal Prediction

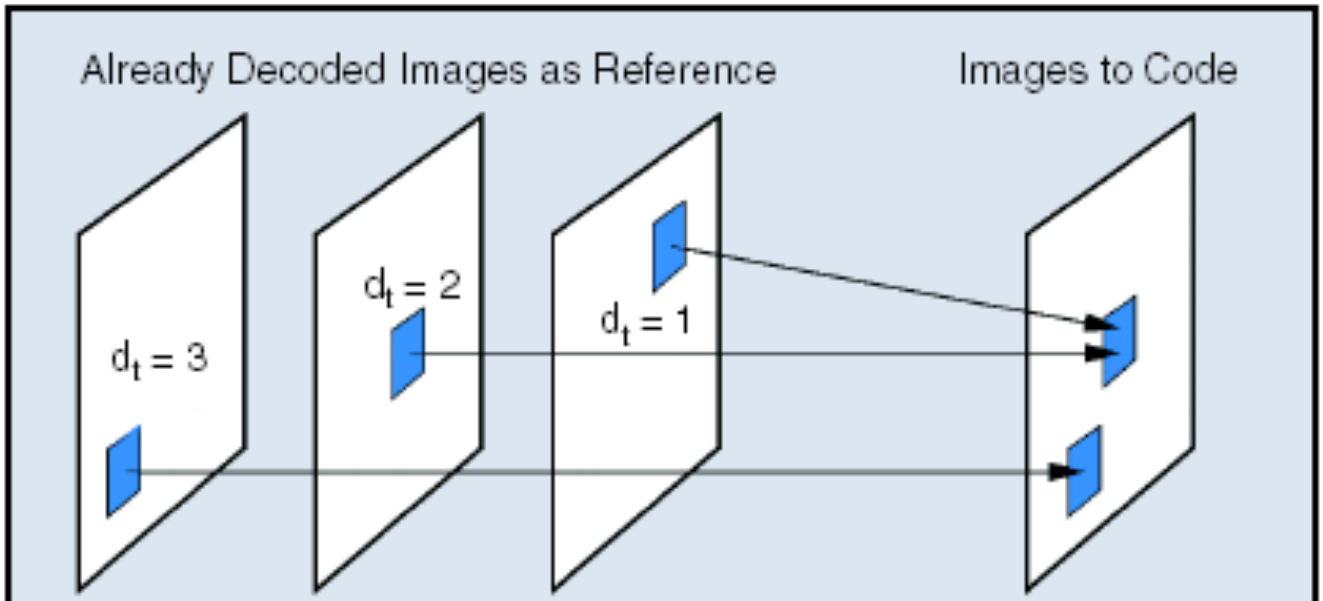


Figure 9. Motion-compensated prediction with multiple reference images. In addition to the motion vector, also an image reference parameter d_t is transmitted.

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

Temporal prediction options

- Predict using one frame or two
- Save this frame for subsequent predictions (or not)
- Some limited ability to use prediction coefficients other than 1 or ½

- Lots of flexibility for frame types to be chosen for best compression, or low delay, or error resilience

Spatial Compression: Theory vs. Practice (1)

- Theory: Karhunen Loeve Transform is best possible block-based transform
- Problems with theory:
  - Finding an accurate model 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
  - Also, approximations to DCT and also DST option

Spatial Compression: Theory vs. Practice (2)

- 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
  - *There is still correlation between adjacent blocks*

Spatial Prediction

- General idea:
  - A pixel in the new block is predicted from previously coded pixels in the same frame
  - What neighbors? What weighting coefficients?
- 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 (and HEVC) have many possible intra-prediction patterns

H.264 Intra-Prediction

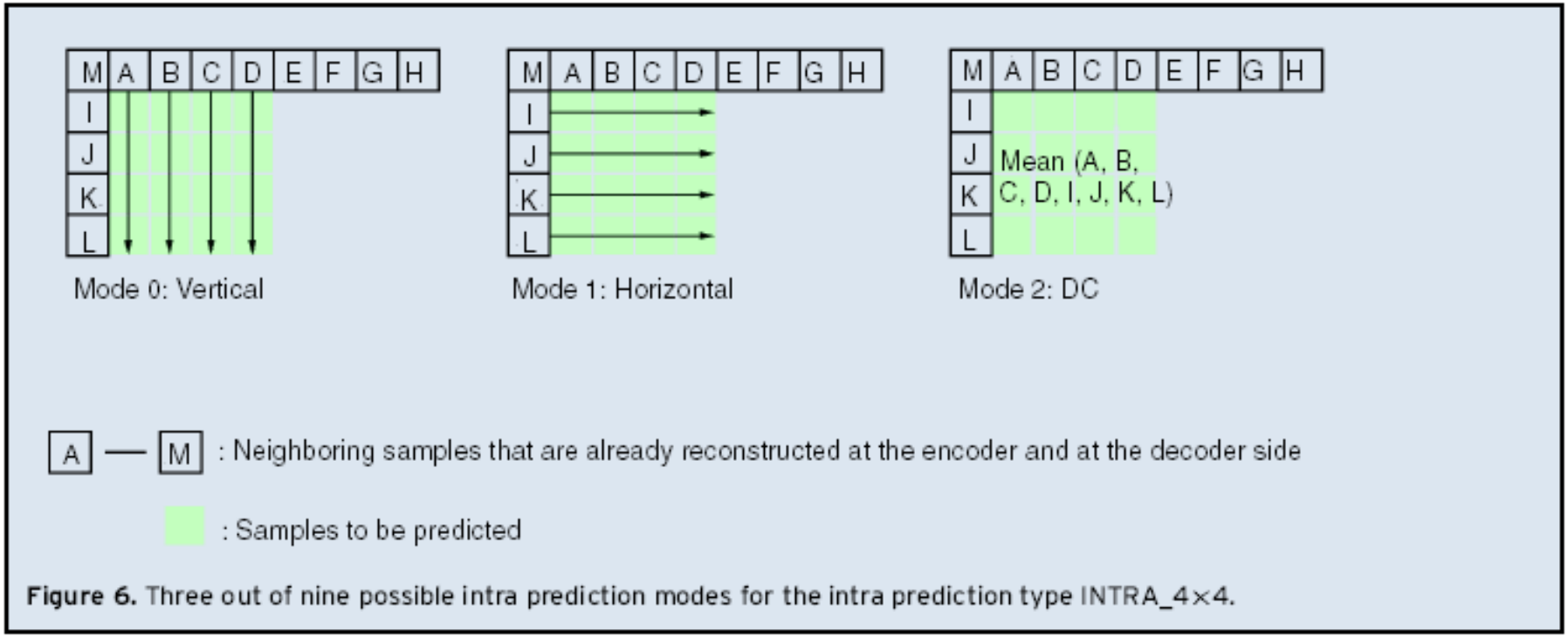
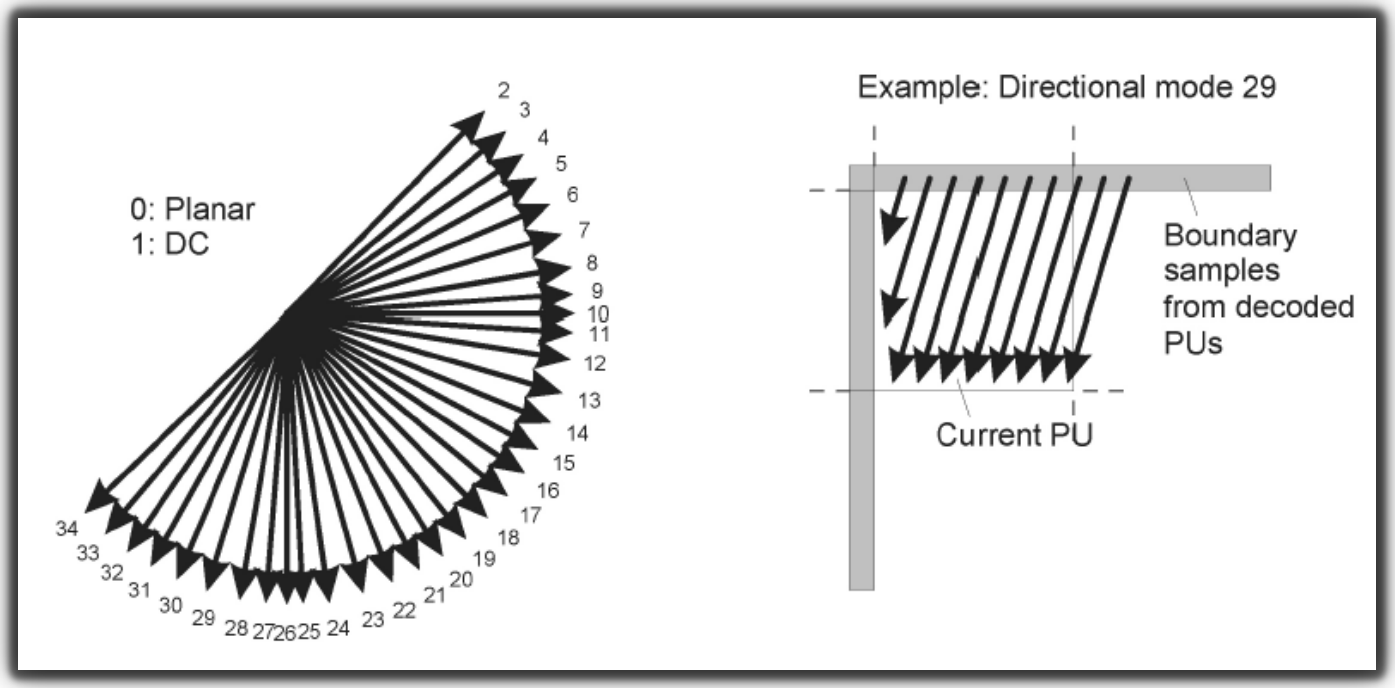


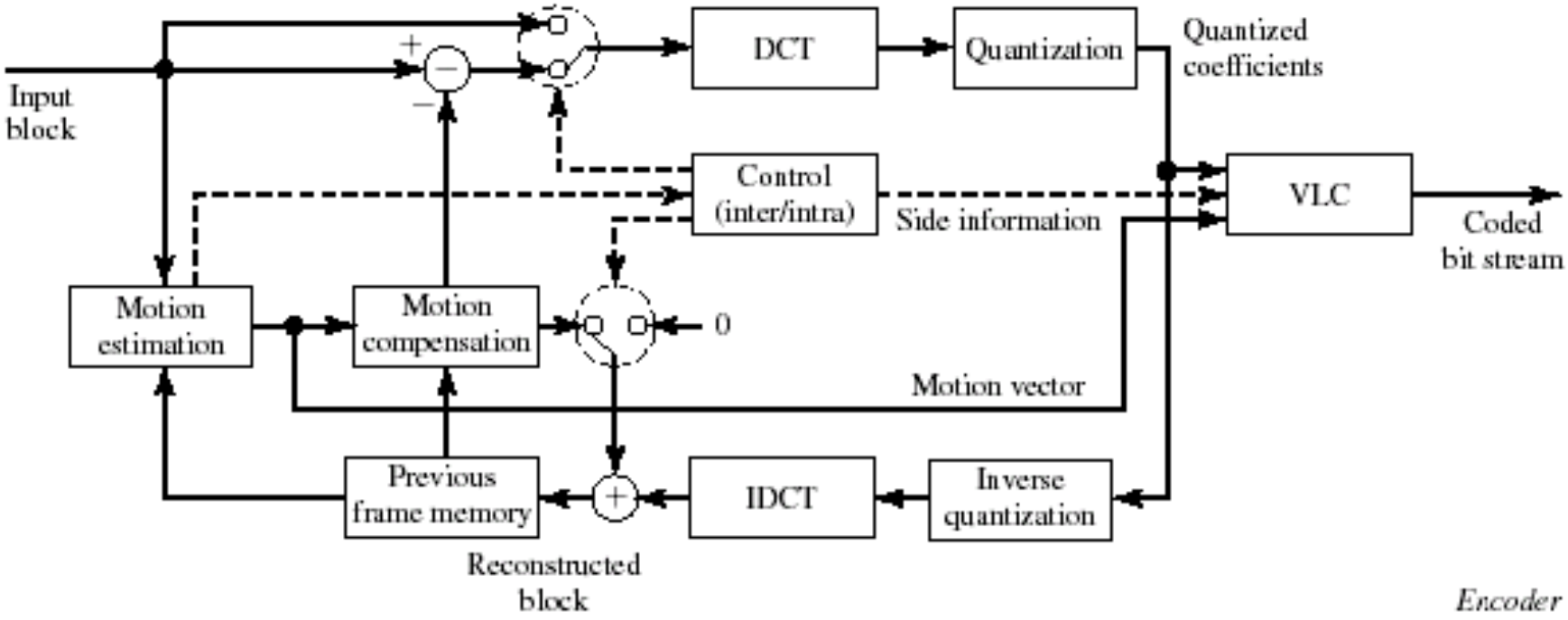
Figure 6. Three out of nine possible intra prediction modes for the intra prediction type INTRA_4×4.

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

HEVC intra-prediction



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



Hybrid: both prediction (temporal) and transform (spatial)