Lecture 12
Basics Digital Video – Part 2

References:

1. Chapter 6: Two-Dimensional Motion Estimation, in Video Processing and Communications, by Wang, Ostermann and Zhang (specifically sections 6.1, 6.2, 6.4.1, and 6.4.4)

2. Ostermann et al. Video coding with H.264/AVC: Tools, performance, and complexity, IEEE Circuits and Systems Magazine, 1st Quarter, 2004.

Outline

- Three types of redundancies and where they are exploited in image and hybrid video compression
- Temporal redundancy via prediction
- Motion estimation and motion compensation
- Block Matching Algorithm (BMA)
- Motion compensated prediction
- Reducing artifacts (blocking and ringing)
- Video encoders

Redundancies in video compression

Perceptual redundancy

- The Human Visual System is less sensitive to color and high frequencies
- Down-sampled chrominance

Spatial redundancy

- Neighboring pixels have close luminance levels
- Intra-frame compression making use of
  - Correlation/compression within a frame

Temporal redundancy

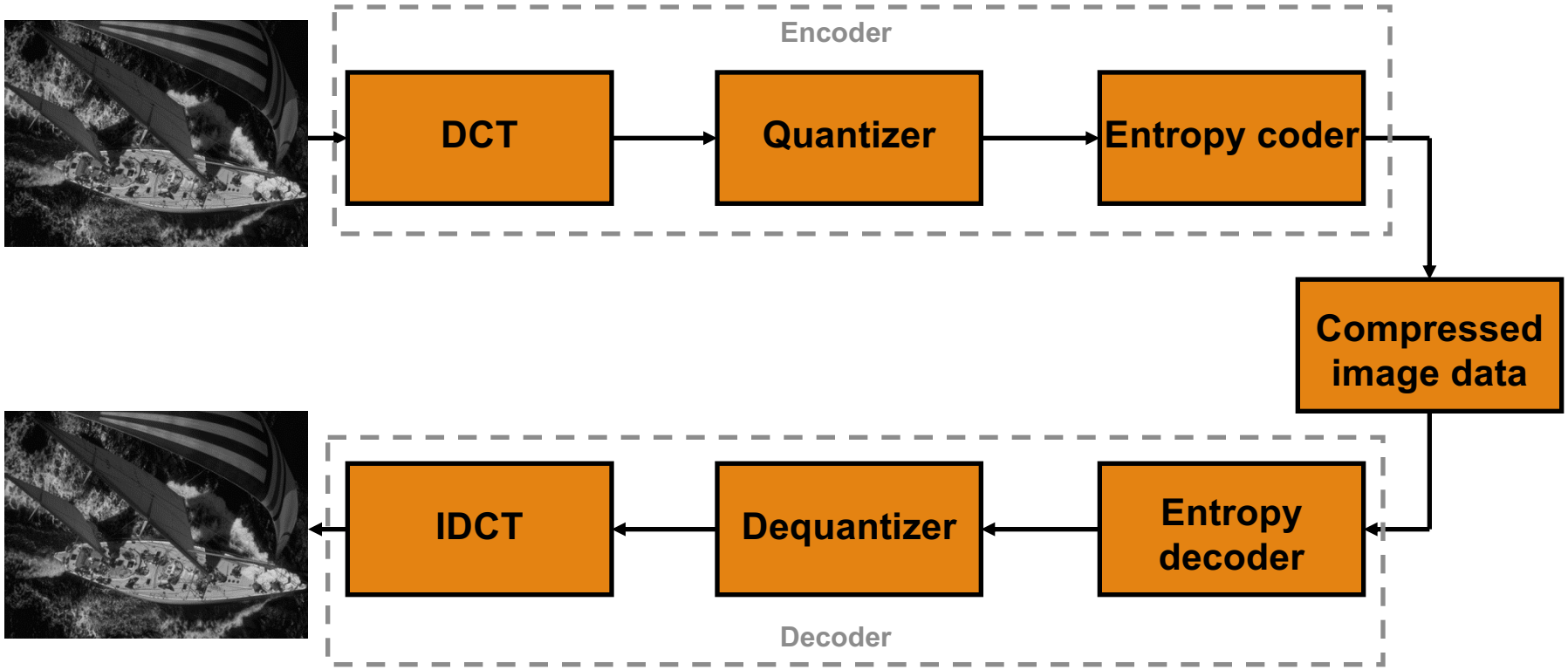
- Differences between adjacent frames can be small
- Inter-frame compression exploiting
  - Correlation/compression between frames
  - Motion estimation and motion compensation

Statistical redundancy

Context-adaptive binary arithmetic coding (**CABAC**) (entropy **encoding)** is applied to the output bitstream

Which of these redundancies can be exploited in Image Compression?

Fundamentals of JPEG



Fundamentals of JPEG

JPEG uses 8×8 blocks

Converts the blocks to DCT domain

Quantize each coefficient

- Different step-sizes for each coefficient
  - Based on sensitivity of human visual system

Order coefficients in zig-zag scan

- Similar frequencies are grouped together

Run-length encode the quantized values and then use variable length (Huffman) coding on what is left

Conclusion:

Perceptual, spatial and statistical redundancies are exploited.

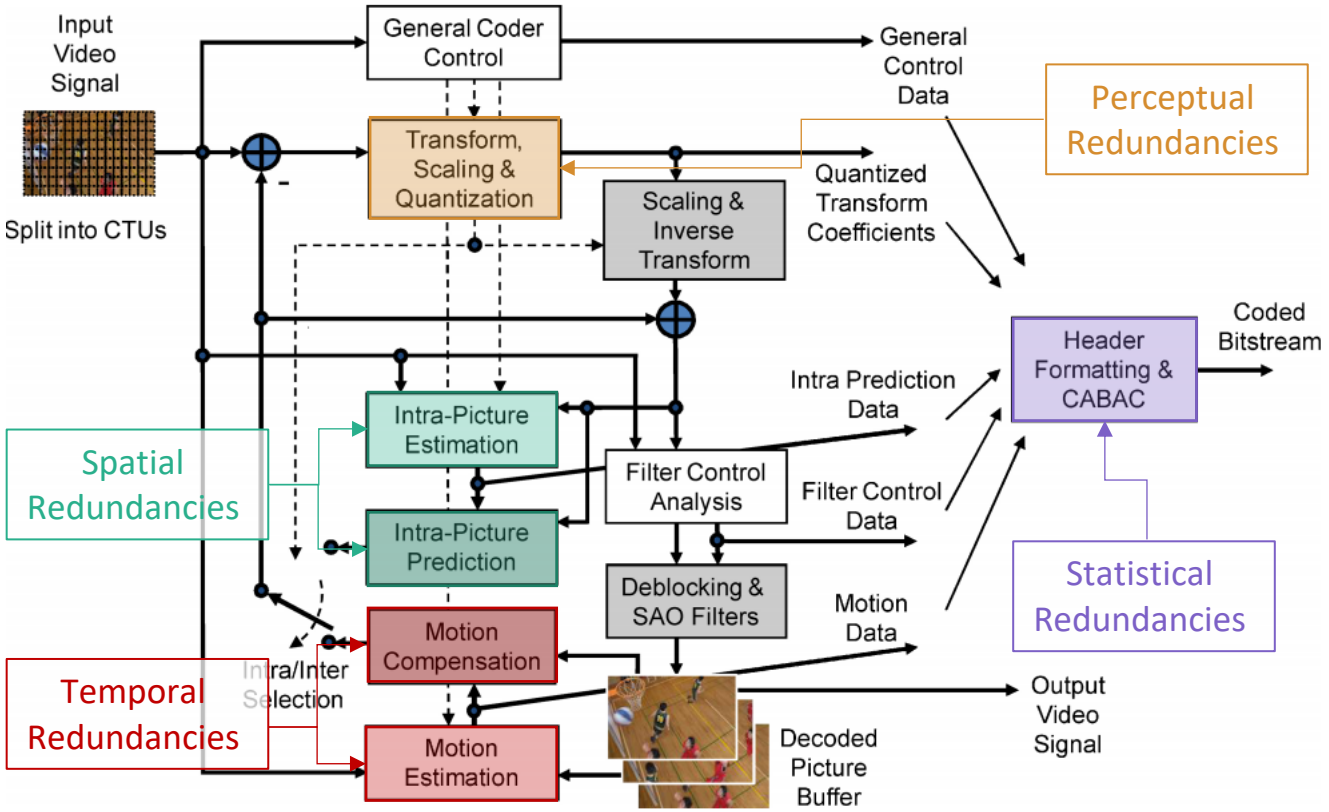
Still image compression versus hybrid video compression

- Still image compression ignores the correlation between frames of video
  - JPEG achieves ~10:1 compression ratio
  - Wavelet transform-based image coding reaches compression ratios up to ~30:1
- Adding motion estimation and compensation results in much higher compression ratios
  - Good video quality at compression ratios as high as ~200:1

Key ideas in video compression

- Predict a new frame from a previous frame and only code the prediction error
- 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
- Use motion-compensated prediction to account for object motion
- 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)

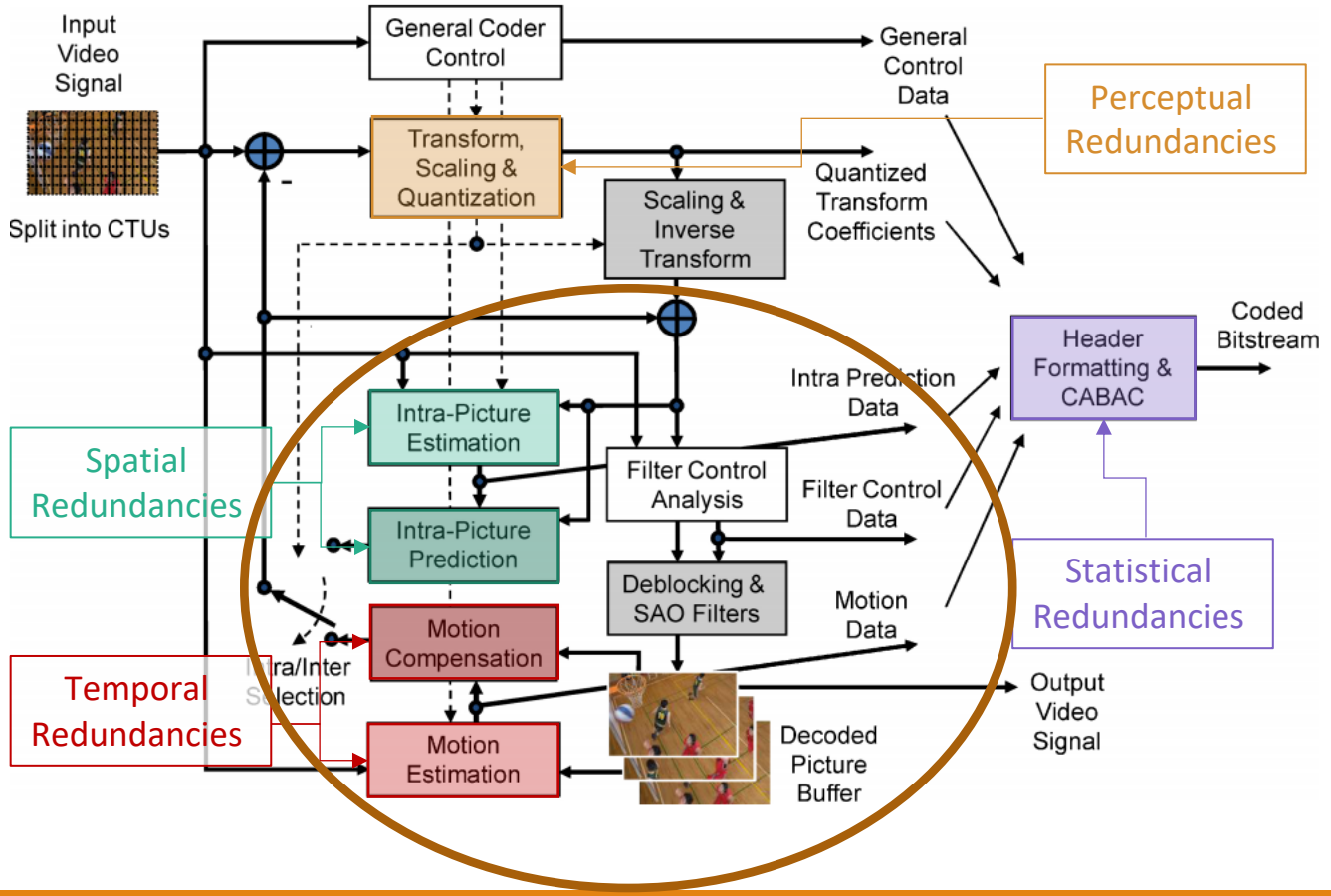
How are redundancies exploited in Hybrid Video Coding?



CTU: coding tree unit

SAO: sample adaptive offset

How to compute the prediction signal?



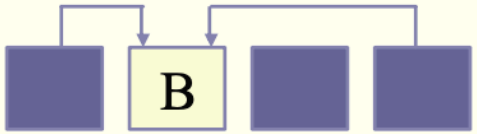
Prediction = Estimation + Compensation

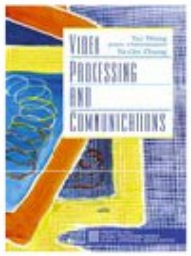
- Requires at least one "reference frame"
  - Reference frame must be encoded before the current frame
  - But, reference frame can be a future frame in the display sequence
- Three kinds of frames: I, P, and B

- I frame is encoded as a still image and doesn't depend on any reference frame

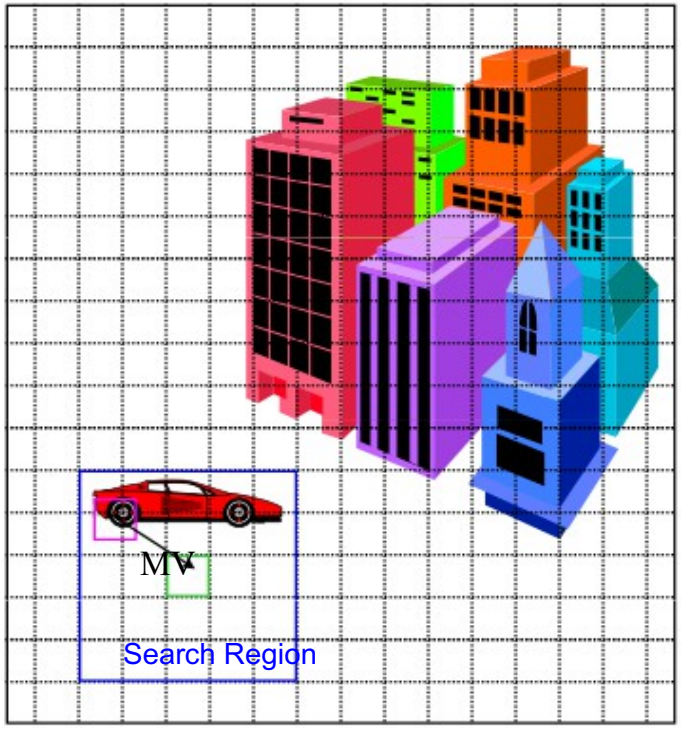
- P frame depends on previously displayed reference frame

- B frame depends on previous and future reference frames

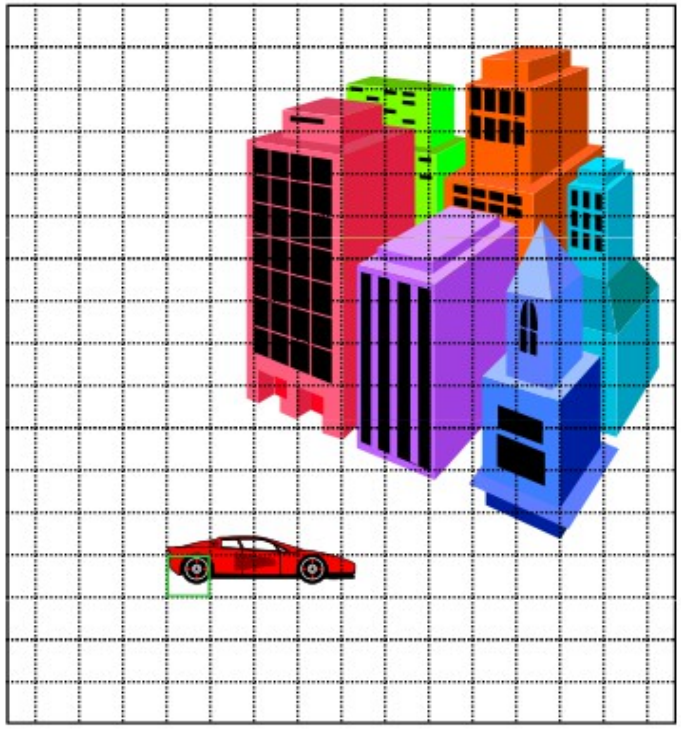




Motion Estimation – Block Matching Algorithm (BMA)



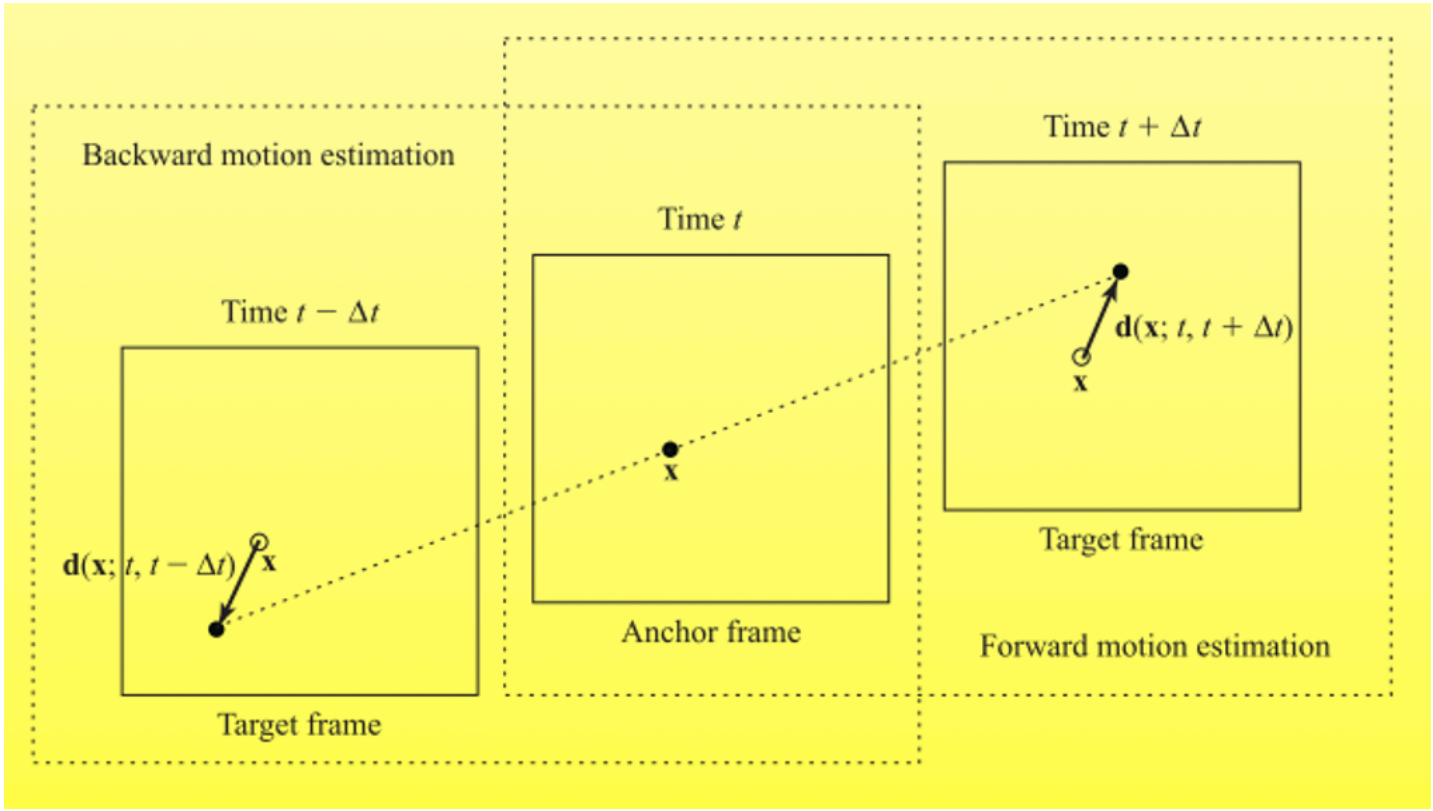
Frame *t-1*
(Reference Frame)



Frame *t*
(Predicted frame)

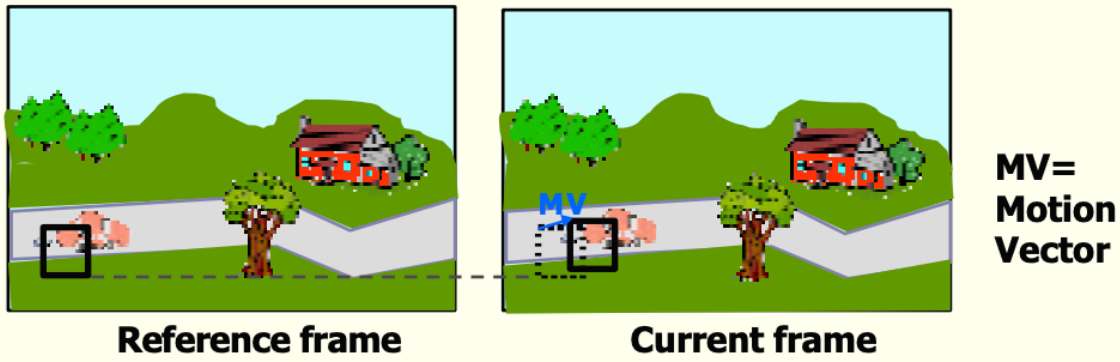
Adjacent frames are similar and changes are due to object or camera motion --- Temporal correlation

Reference (target) frames before and after the anchor frame



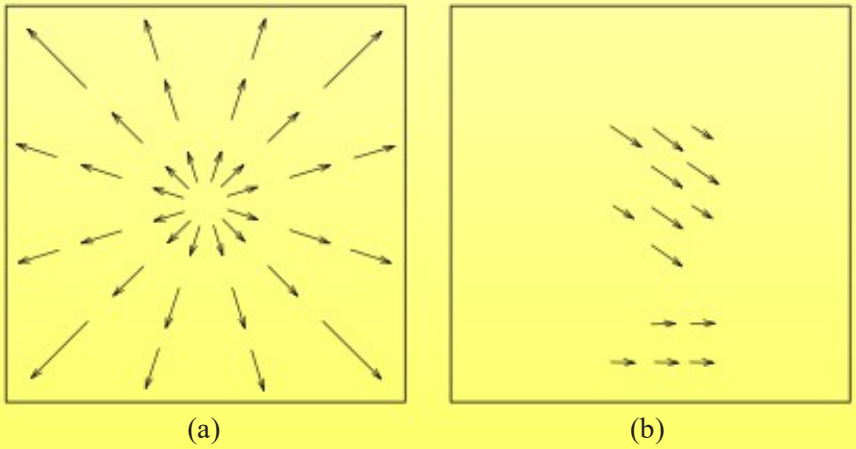
Motion estimation

- Predict the contents of each macroblock based on motion relative to reference frame
  - Search reference frame for a 16x16 region that matches the macroblock
  - Encode motion vectors
  - Encode difference between predicted and actual macroblock pixels



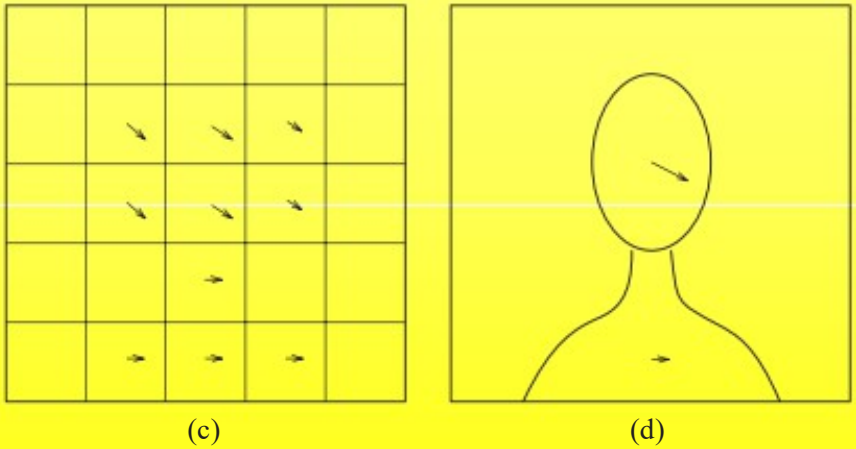
Motion Representations

Global Motion:
Entire motion field is represented by a few global parameters



Pixel-based motion:
One motion vector at each pixel, with some smoothness constraint between adjacent motion vectors.

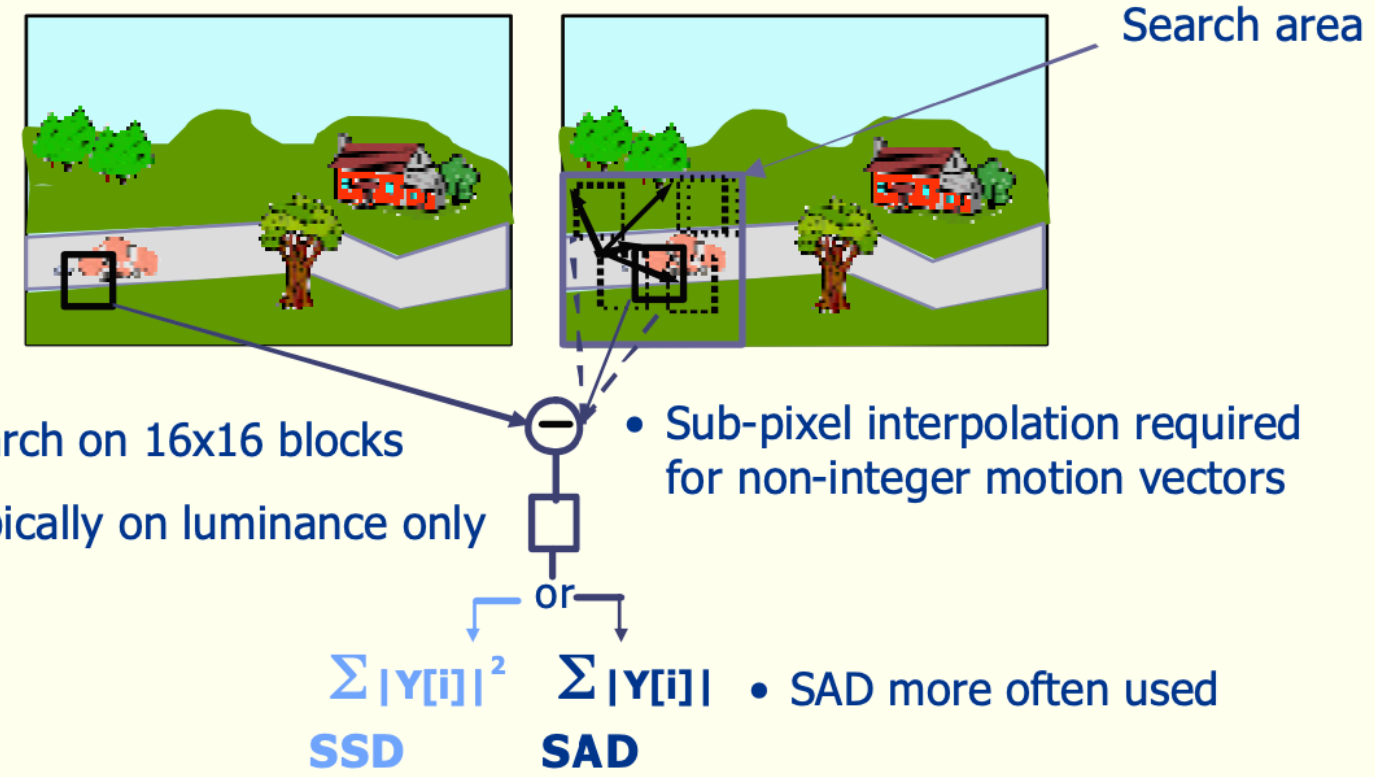
Block-based motion:
Entire frame is divided into blocks, and motion in each block is characterized by a few parameters.



Region-based motion:
Entire frame is divided into regions, each region corresponding to an object or sub-object with consistent motion, represented by a few parameters.

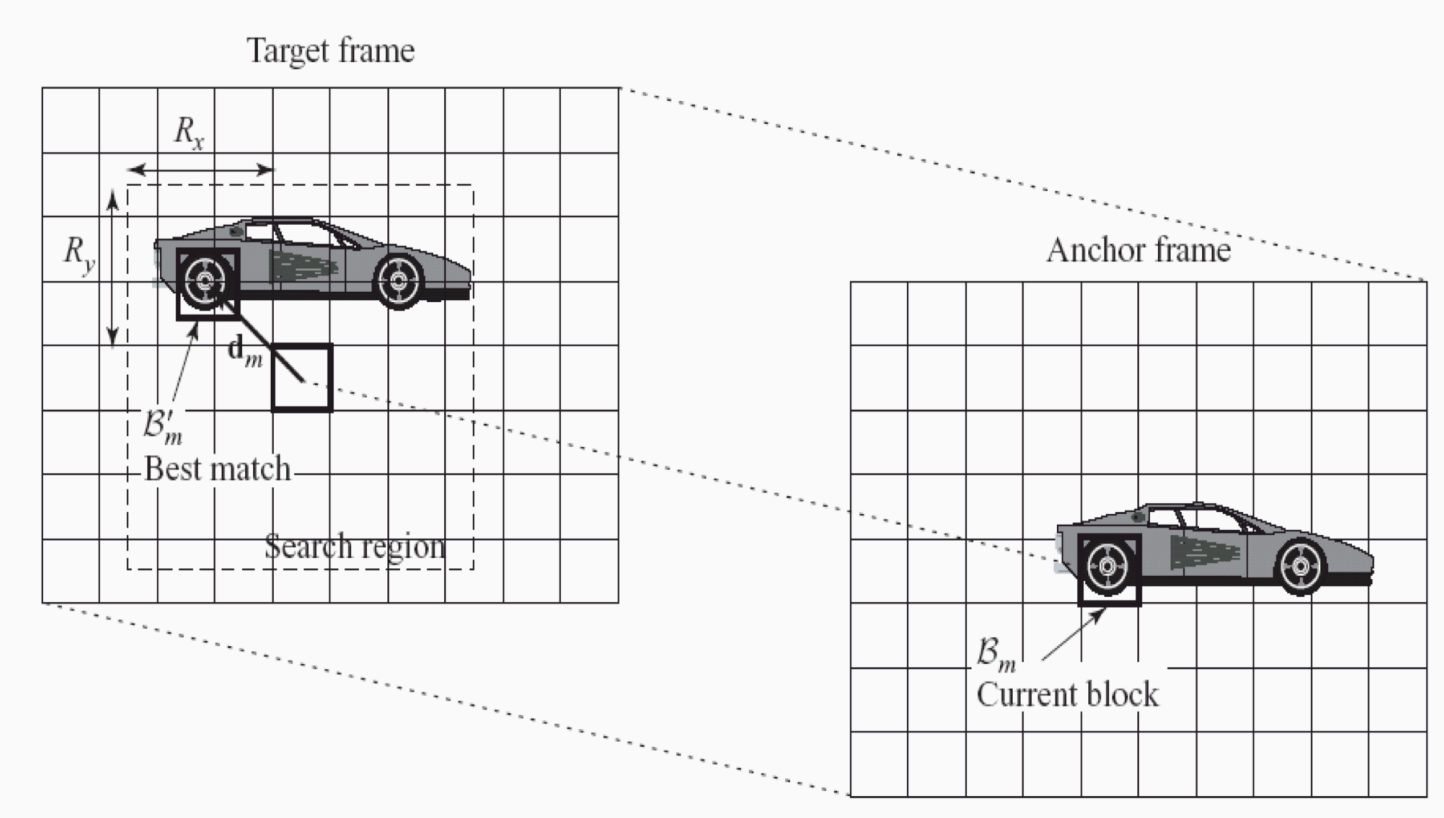
Other representation: mesh-based (control grid)

Motion estimation: The problem



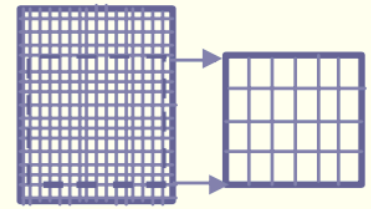
SSD = Sum of Squared Differences SAD = Sum of Absolute Differences

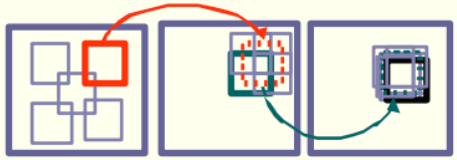
Exhaustive Block Matching Algorithm (EBMA)

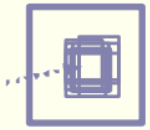


EBMA searches the full search region to find the best match

Motion estimation: efficient motion vector search







- Exhaustive search is impractical
- Evaluate only promising candidate motion vectors
  - Don't have to find absolute best match
    - Trade video quality for computational load
  - Many methods in use
    - Often proprietary
    - Refine candidate vector selection in stages
    - Predict candidate vectors from surrounding macroblocks and/or previous frames

Motion vector search approach is a key differentiator between video encoder implementations

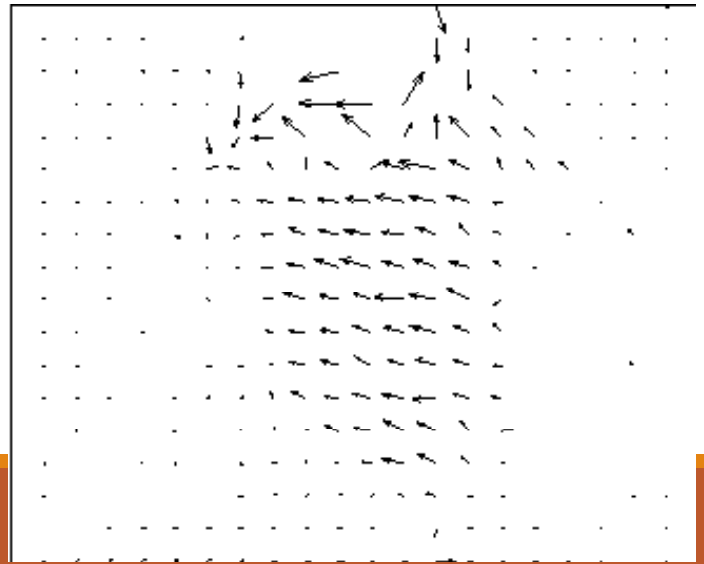
Motion estimation: complexity

- Compute load
  - Most demanding task in video compression
    - Up to 80% of total encoder processor cycles
    - Many search methods exist; requirements vary by method
    - May vary with video program content
    - Makes encoder computational demand several times greater than that of the decoder
    - Dominated by SAD computation
- Memory usage
  - Motion estimation requires reference frame buffers
    - Frame buffers dominate the memory requirements of the encoder
    - E.g., 152,064 bytes per frame @ CIF (352x288) resolution
  - High memory bandwidth required

Sample (2D) Motion Field

Anchor Frame

Target Frame



Motion Field

Advantages and disadvantages of Block Matching Algorithms

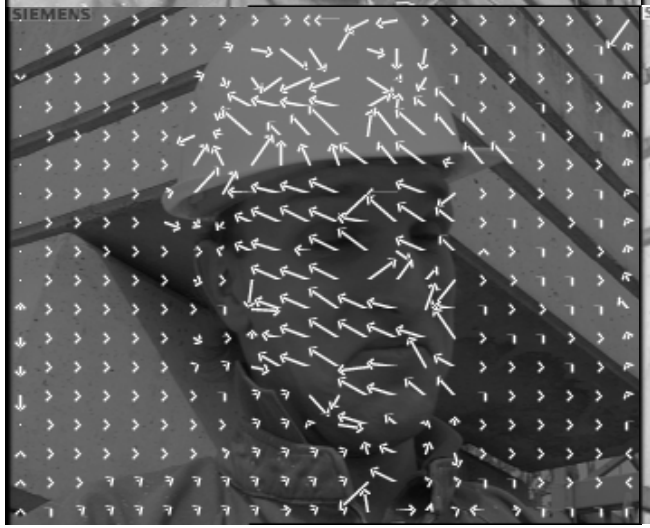
- Blocking effect (discontinuity across block boundary) in the predicted image
  - Because the block-wise translation model is not accurate
  - Fix: Deformable BMA
- Motion field somewhat chaotic
  - because MVs are estimated independently from one block to another
  - Fix 1: Mesh-based motion estimation
  - Fix 2: Imposing smoothness constraint explicitly
- Wrong MV in flat regions
  - because motion is indeterminate when spatial gradient is near zero
- Nonetheless, it is widely used for motion compensated prediction in video coding
  - Because of its simplicity and optimality in minimizing the prediction error

Examples of motion estimation results (1/3)

target frame

anchor frame

Motion field



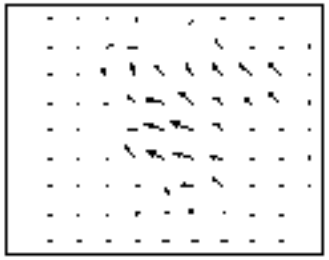
Predicted target frame obtained by half-pixel accuracy

Half-pel accuracy interpolates motion vectors in subpixels between existing rows and columns

Examples of motion estimation results (2/3)

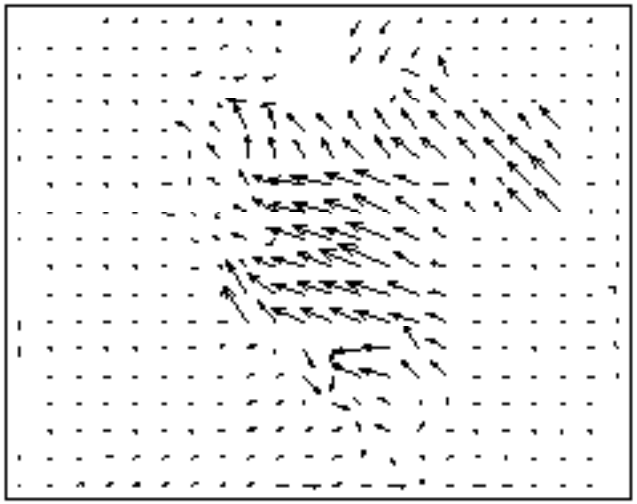
(a)

(b)



(c)

(d)

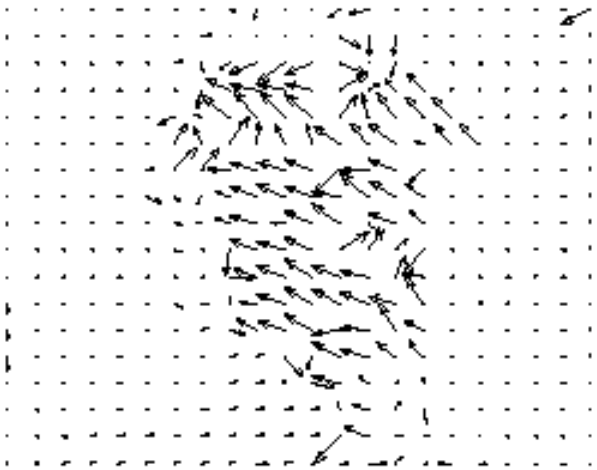


(e)

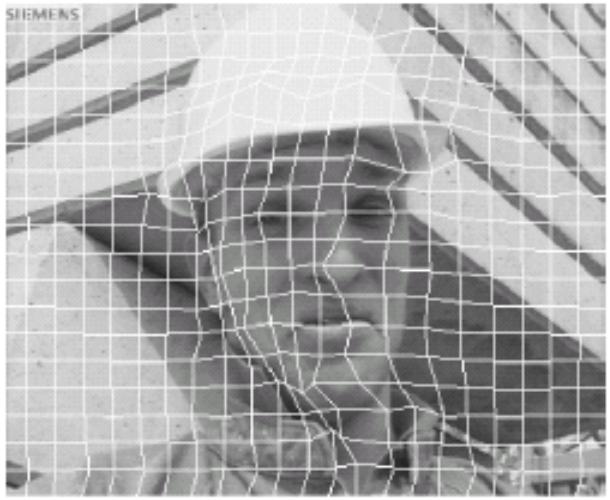
(f)

Predicted target frame

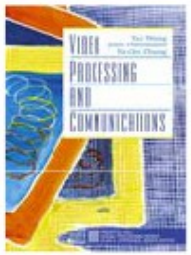
Examples of motion estimation results (3/3)



EBMA



mesh-based method



Multiple Reference Frame Temporal Prediction

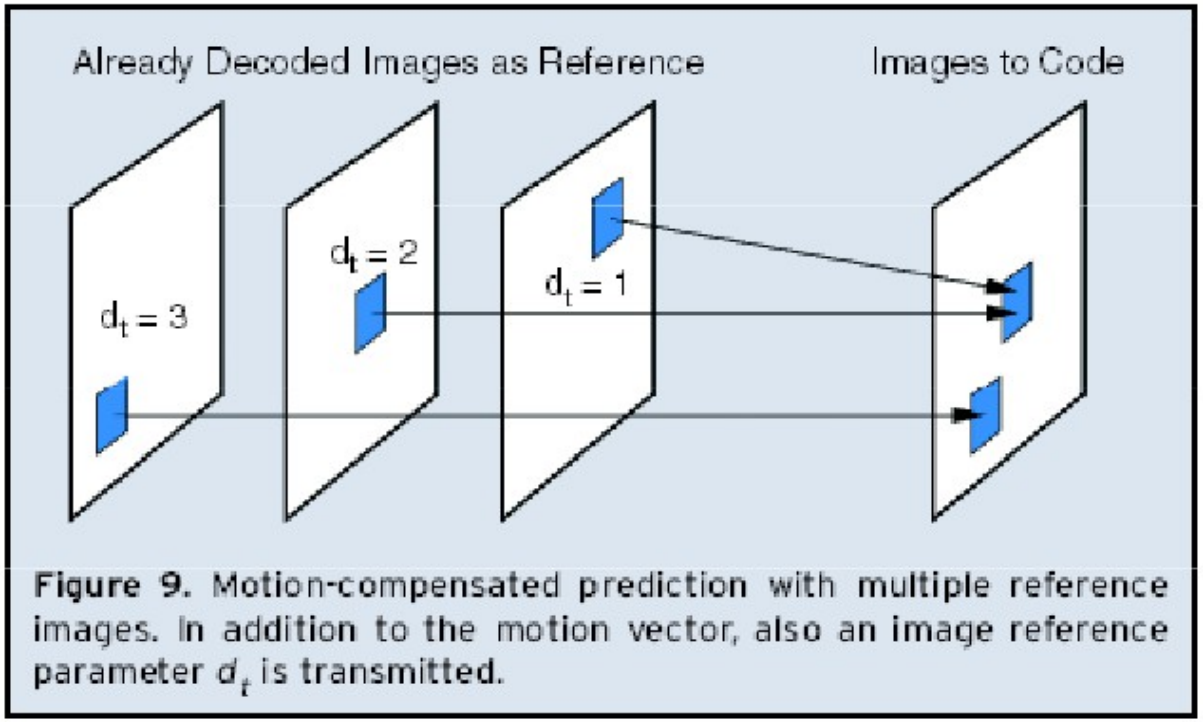
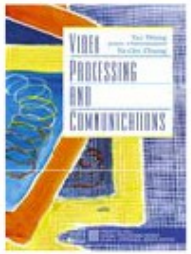


Figure 9. Motion-compensated prediction with multiple reference images. In addition to the motion vector, also an image reference parameter d_t is transmitted.

When multiple references are combined, the best weighting coefficients can be determined using SSD and SAD predictor



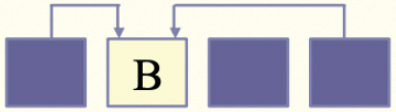
Group-of-Picture Structure (GOP)

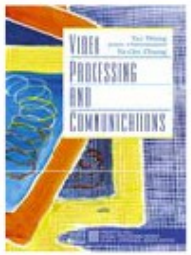
- I-frames coded without reference to other frames
  - To enable random access (channel change), fast forward, stopping error propagation
- P-frames coded with reference to previous frames
- B-frames coded with reference to previous and futu frames (bi-directional prediction)
  - Require extra delay!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

- I frame is encoded as a still image and doesn't depend on any reference frame

- P frame depends on previously displayed reference frame

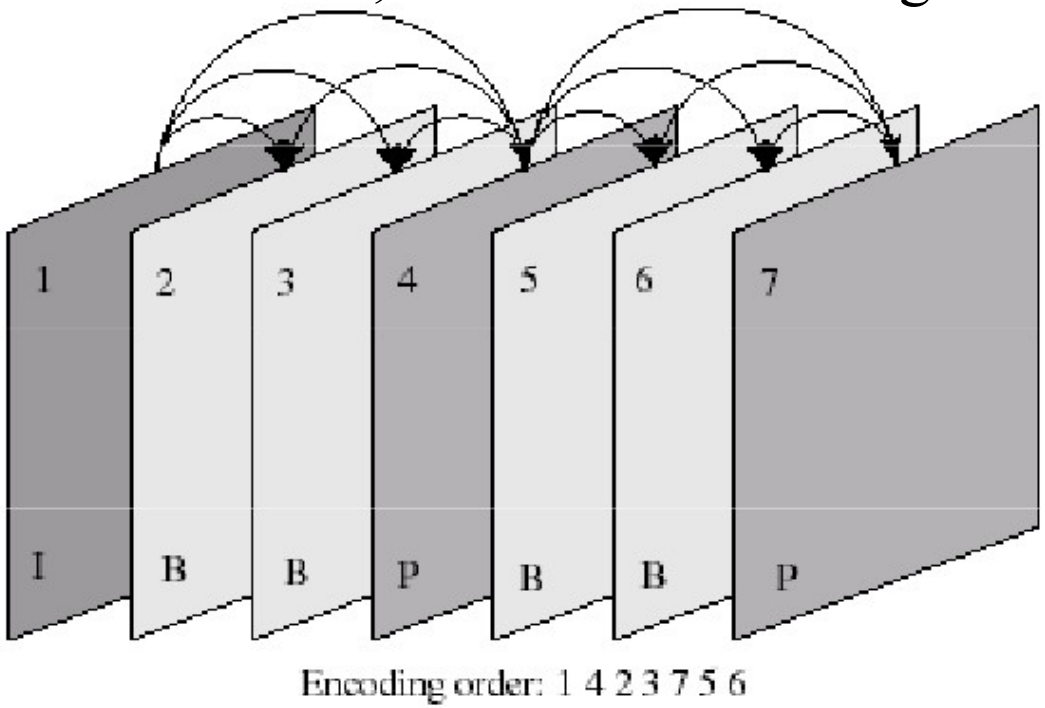
- B frame depends on previous and future reference frames





Group of Picture Structure

Given a GOP of IBBPBBP, what is the encoding order?



Encoding order:

1st, 4th (using 1st), 2nd (using 1st, 4th), 3rd (using 1st, 4th), 7th (using 4th), 5th (using 4th , 7th), 6th (using 4th , 7th)

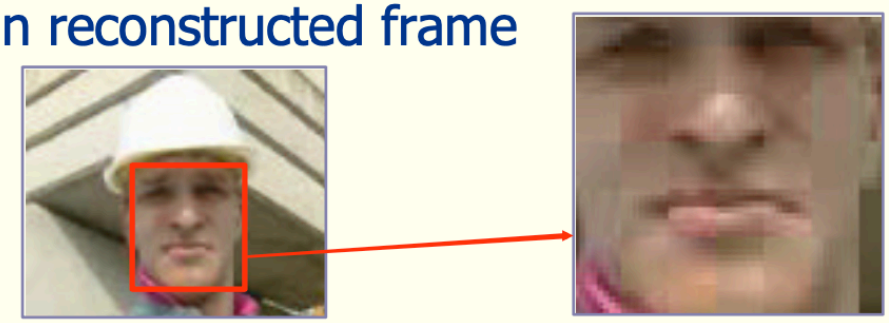
Motion Compensated Prediction

- Divide current frame into disjoint 16×16 macroblocks (MB)
- For each MB, search a window in the reference frame(s) for closest match (e.g. min Sum of Absolute Difference, SAD)
- Motion compensation copies pixels from the reference frame to predict the current macroblock (calculate the prediction error from min SAD)
- For each of the four 8×8 blocks in the macroblock, perform DCT-based coding
- Transmit motion vector (displacement between the current MB and the best matching MB) + entropy coded prediction error (lossy coding)
- Computational load
  - Varies with video content
  - can require 5-40% of the total decoder processor cycles
- Memory usage
  - Require reference frame buffers

Reducing artifacts

Artifacts: Blocking and Ringing

- **Blocking:** Borders of 8x8 blocks become visible in reconstructed frame



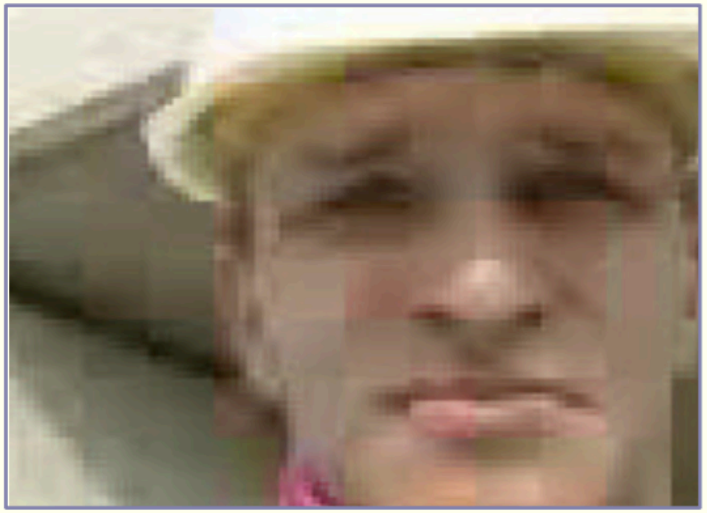
- **Ringing:** Distortions near edges of image features

Deblocking and deringing filters

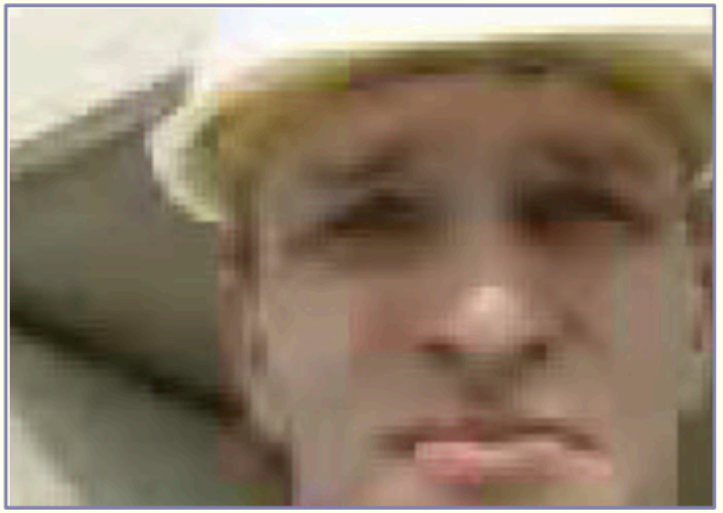
Low-pass filters are used to smooth the image where artifacts occur

- **Deblocking:**
  - Low-pass filter the pixels at borders of 8x8 blocks
  - One-dimensional filter applied perpendicular to 8x8 block borders
- **Deringing:**
  - Detect edges of image features
  - Adaptively apply 2D filter to smooth out areas near edges
  - Little or no filtering applied to edge pixels in order to avoid blurring

Example of deblocking with a lowpass filter



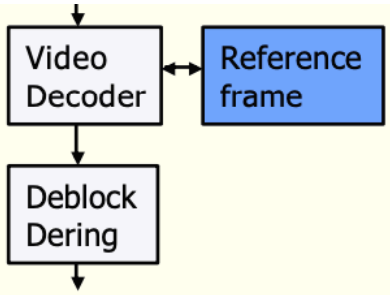
Original MPEG Still Frame



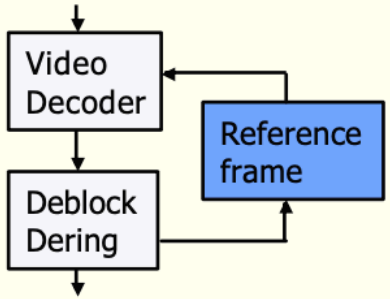
Horizontally & Vertically Deblocked Still Frame

Artifact reduction: Post-processing versus In-loop filtering

- Deblocking/deringing often applied after the decoder (post-processing)
  - Reference frames are not filtered
  - Developers free to select best filters for the application or not filter at all



- Deblocking/deringing can be incorporated in the compression algorithm (in-loop filtering)
  - Reference frames are filtered
  - Same filters must be applied in encoder and decoder
  - Better image quality at very low bit-rates



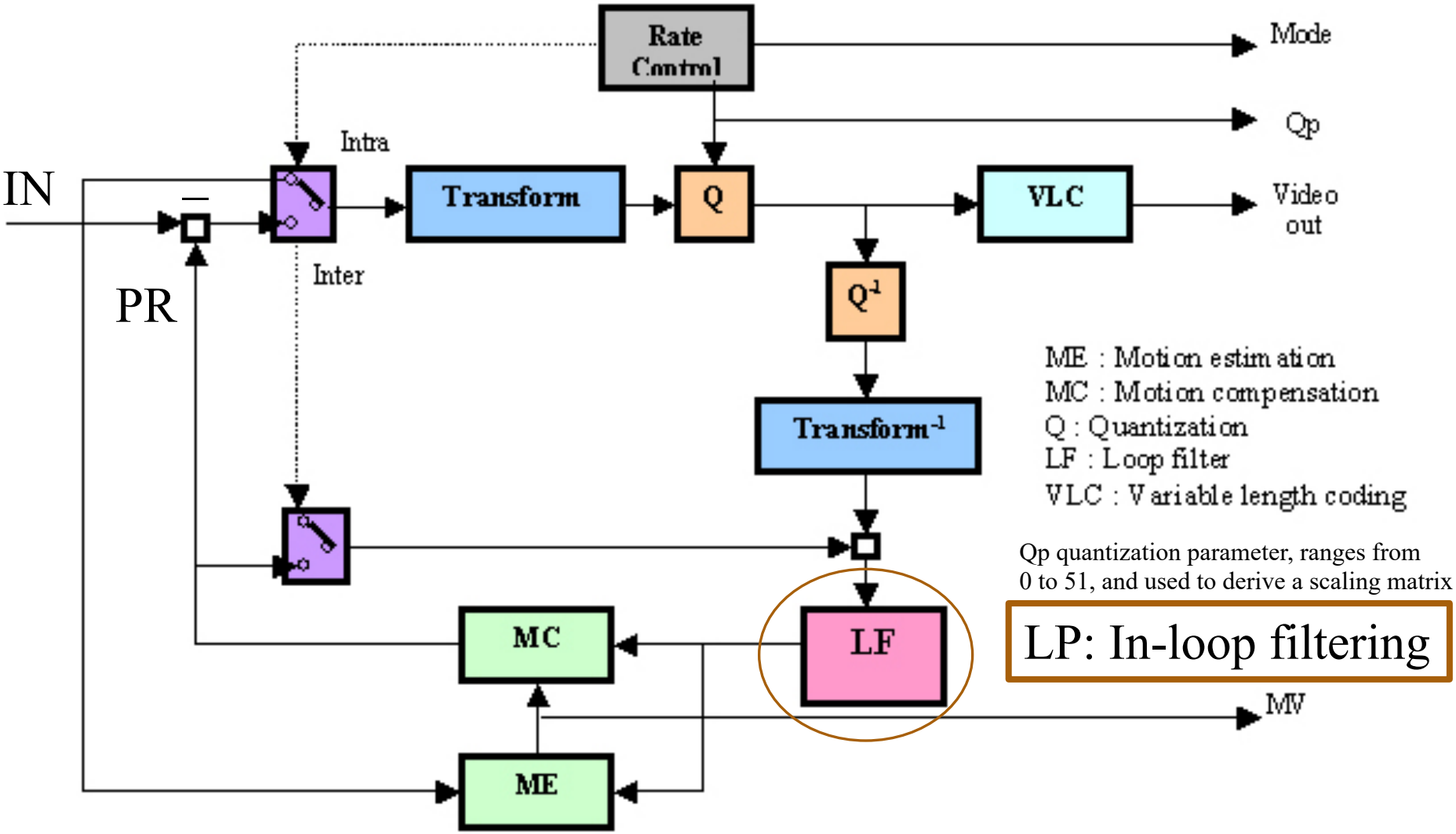


Figure 1: Generic Video encoder's block diagram

Artifact reduction: Complexity

- Deblocking and deringing filters can require more processor cycles than the video decoder
  - Example: MPEG-4 Simple Profile, Level 1 (176x144, 15 fps) decoding requires 14 MIPS on ARM's ARM9E for a relatively complex video sequence
  - With deblocking and deringing added, load increases to 39 MIPS
    - Nearly 3x increase compared to MPEG-4 decoding alone!
- Post-processing may require an additional frame buffer

MIPS = Million of Instructions Per Second

Current Image and Video Compression Standards

Standard	Application	Bit Rate
JPEG	Continuous-tone still-image compression	Variable
H.261	Video telephony and teleconferencing over ISDN	p x 64 kb/s
MPEG-1	Video on digital storage media (CD-ROM)	1.5 Mb/s
MPEG-2	Digital Television	2-20 Mb/s
H.263	Video telephony over PSTN	33.6-? kb/s
MPEG-4	Object-based coding, synthetic content, interactivity	Variable
JPEG-2000	Improved still image compression	Variable
H.264 / MPEG-4 AVC	Improved video compression	10's kb/s to Mb/s

MPEG and JPEG: International Standards Organization (ISO)
H.26x family: International Telecommunications Union (ITU)

Final exam

Moodle exam on

Wednesday 16 Dec 2020, 17:00 – 20:00 (sharp)

Make-up exam in March 2021.

Check Moodle, it should have the latest updates!