CMPT 365 Multimedia Systems

Media Compression - Video

Spring 2017

Edited from slides by Dr. Jiangchuan Liu

Introduction

- What's video ?
  - a time-ordered sequence of frames, i.e., images.
- Why to compress?
  - A billionaire problem
- How to compress ?
  - Spatial redundancy – compression on each individual image (Motion JPEG)
  - Temporal redundancy – prediction based on previous images

Temporal Redundancy

- Characteristics of typical videos:
  - A lot of similarities between adjacent frames
  - Differences caused by object or camera motion

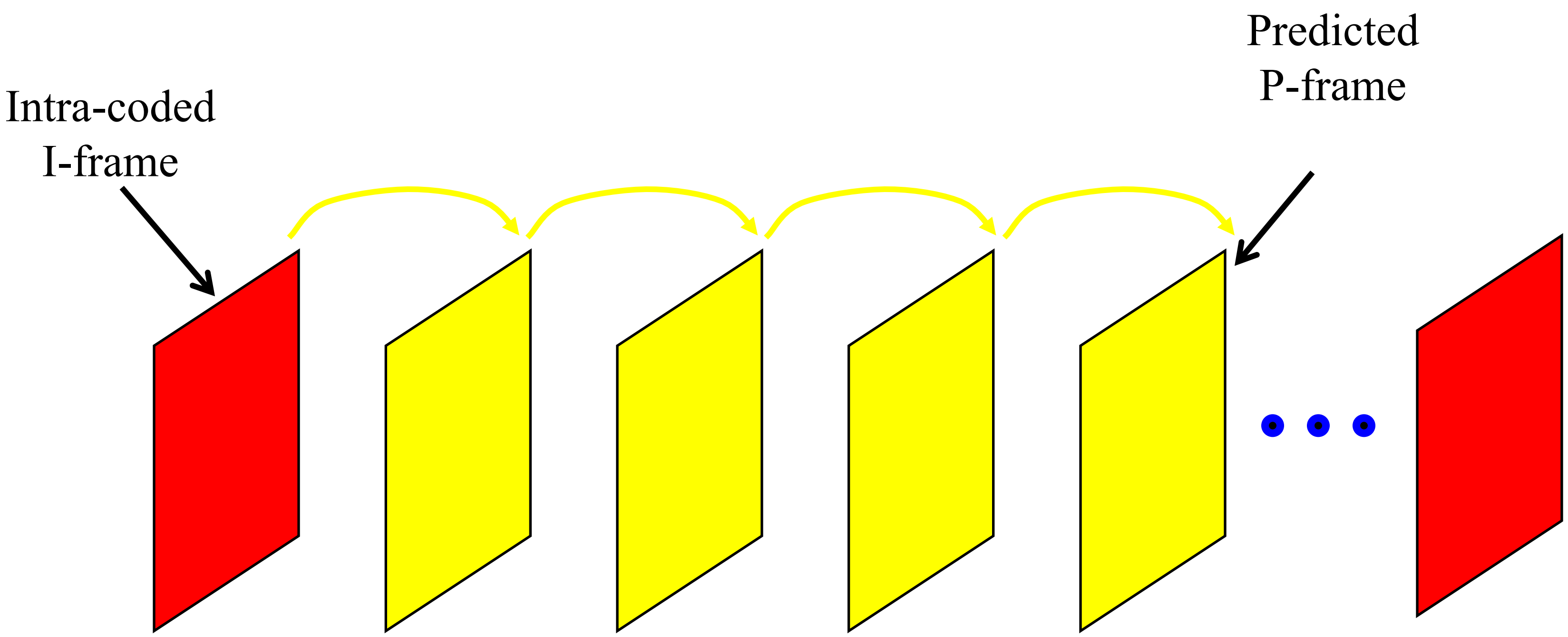
Frame 1

Frame 2

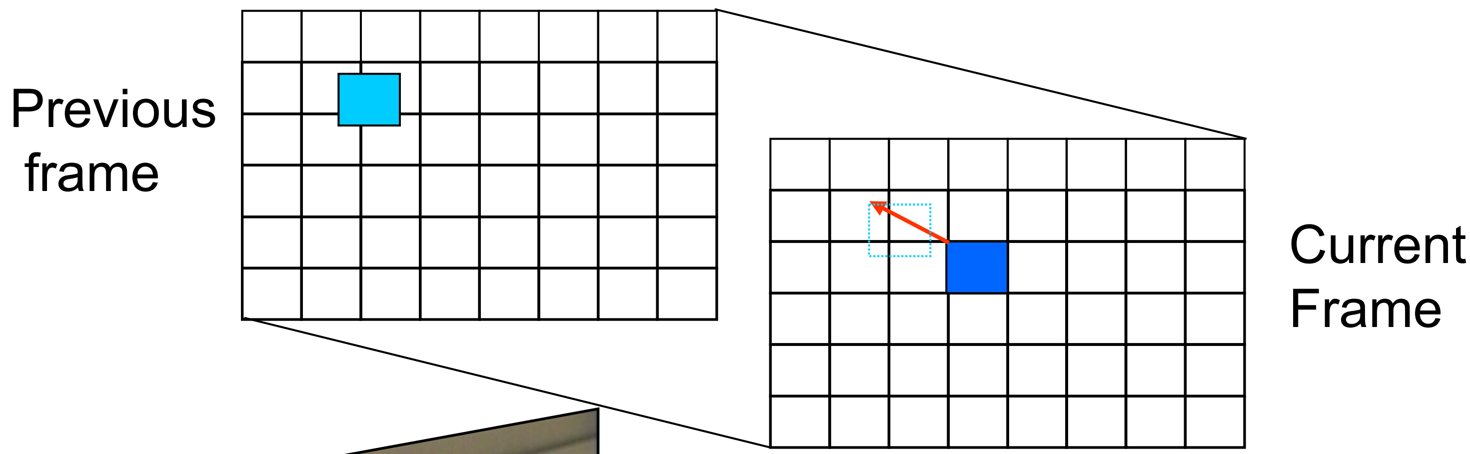
Direct Difference

Key Idea in Video Coding

- Predict each frame from the previous frame and only encode the prediction error:
  - Pred. error has smaller energy and is easier to compress

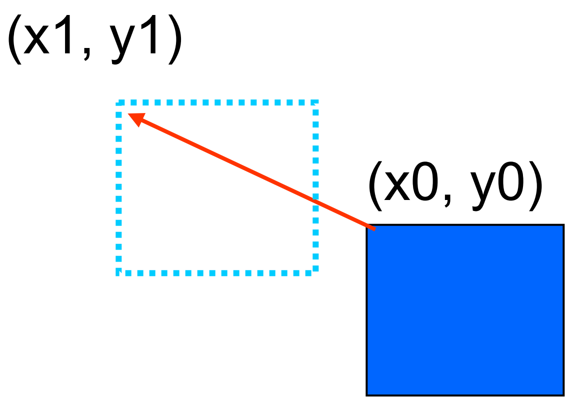


Motion ?



Motion Estimation (ME)

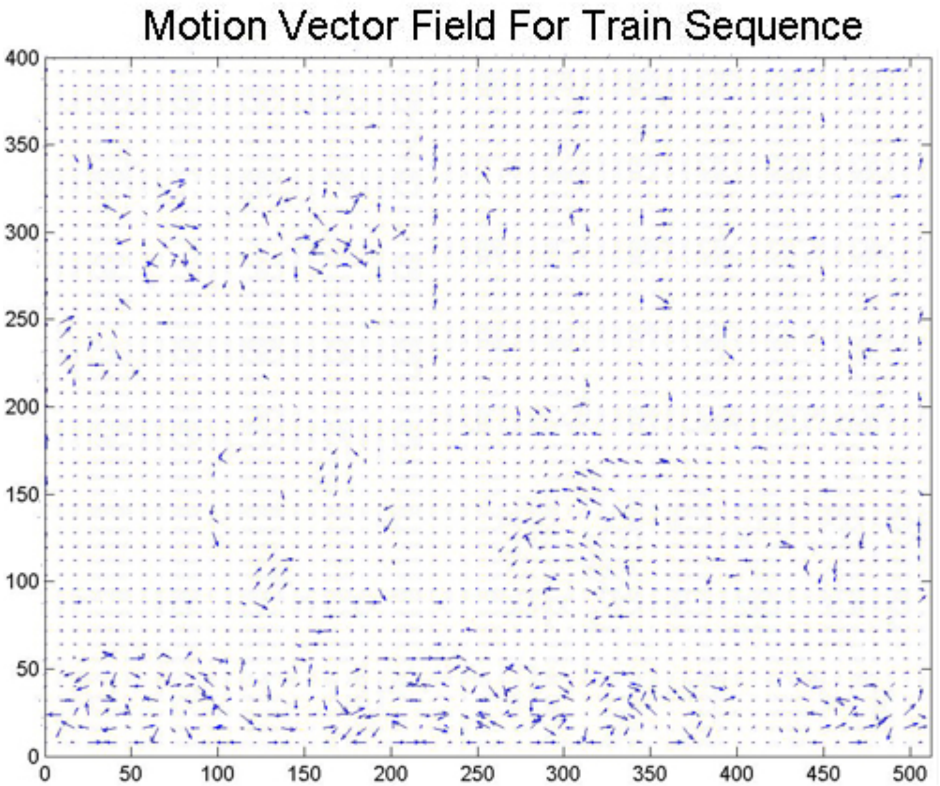
- For each block, find the best match in the previous frame (reference frame)
  - Upper-left corner of the block being encoded: (x0, y0)
  - Upper-left corner of the matched block in the reference frame: (x1, y1)
  - Motion vector (dx, dy): the offset of the two blocks:
    - (dx, dy) = (x1 – x0, y1 – y0)
    - (x0, y0) + (dx, dy) = (x1, y1)
  - Motion vector need to be sent to the decoder.



Motion Estimation Example

Reference

Current Frame

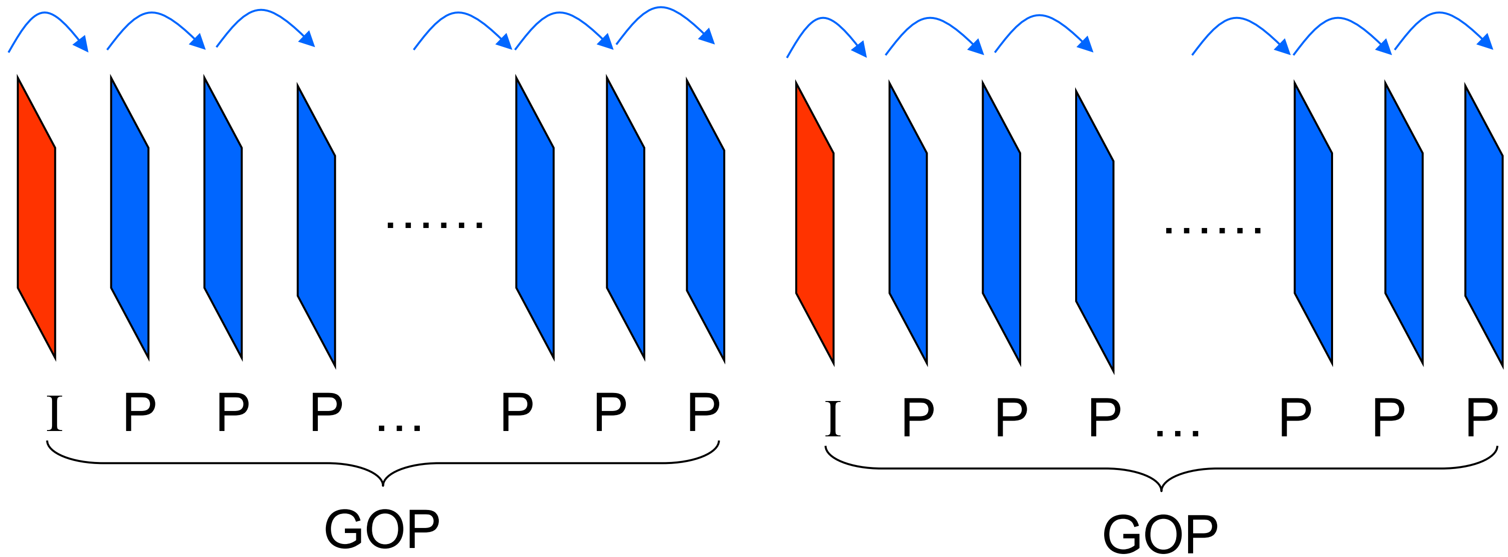


Plotted by quiver( ) in Matlab.

Motion Compensation (MC)

- Given reference frame and the motion vector, can obtain a prediction of the current frame
- Prediction error: Difference between the current frame and the prediction.
- The prediction error will be coded by DCT, quantization, and entropy coding.

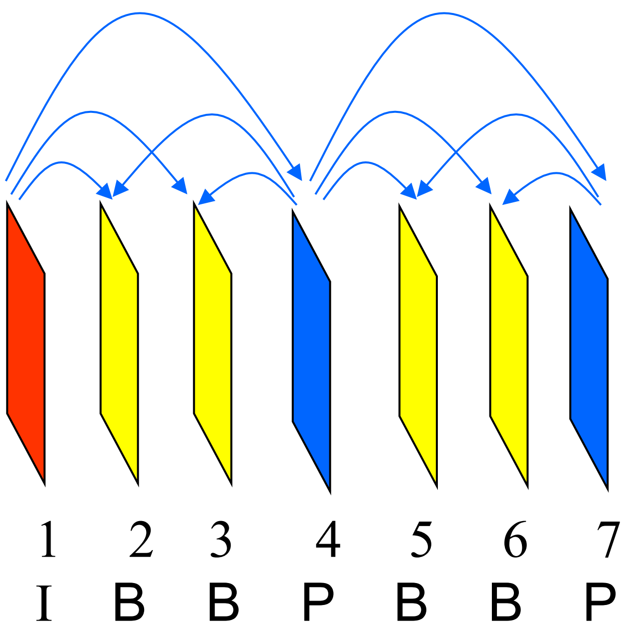
GOP, I, P, and B Frames



- GOP: Group of pictures (frames).
- I frames (Key frames):
  - Intra-coded frame, coded as a still image. Can be decoded directly.
  - Used for GOP head, or at scene changes.
  - I frames also improve the error resilience.
- P frames: (Inter-coded frames)
  - Predication-based coding, based on previous frames.

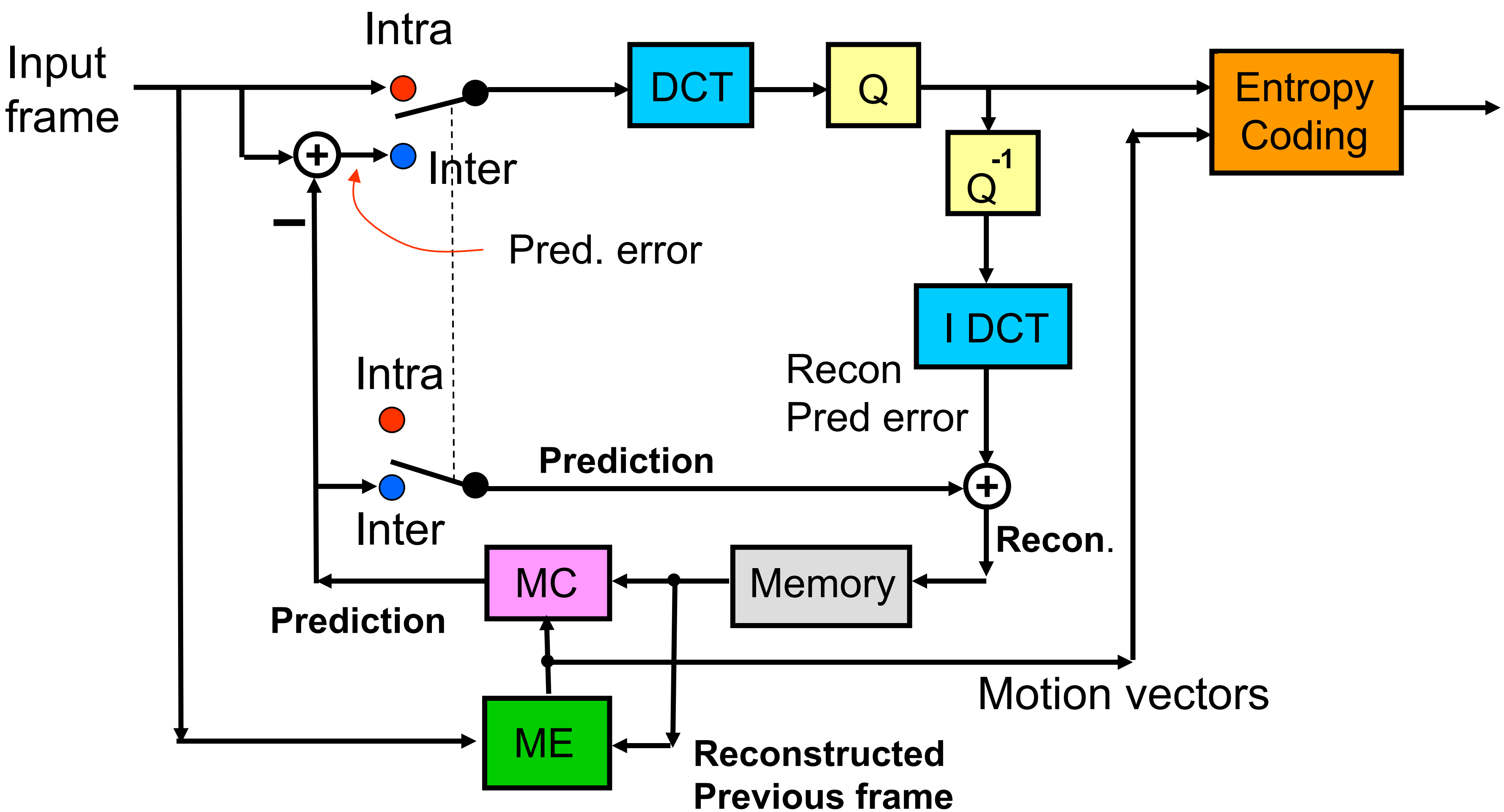
GOP, I, P, and B Frames

- B frames: Bi-directional interpolated prediction frames
  - Predicted from both the previous frame and the next frame: more flexibilities ➔ better prediction.
- B frames are not used as reference for future frames:
  - B frames can be coded with lower quality or can be discarded without affecting future frames.



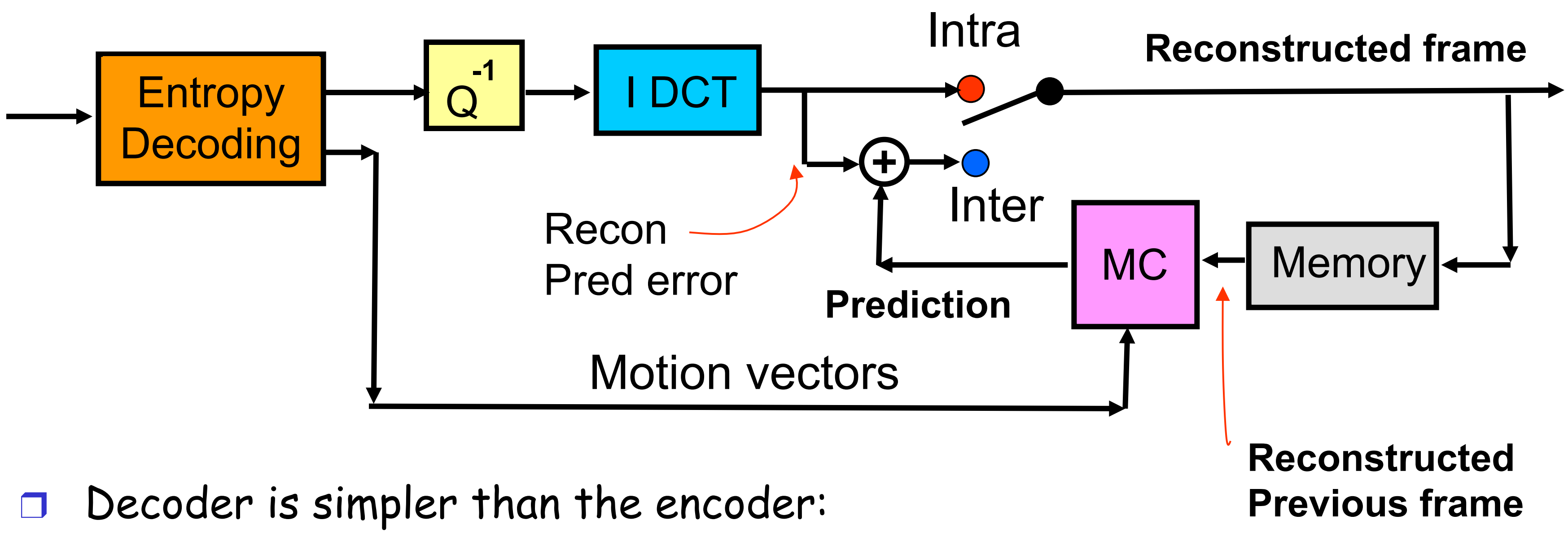
- Encoding order: 1 4 2 3 7 5 6
- Decoding order: 1 4 2 3 7 5 6
- Display order: 1 2 3 4 5 6 7
- Need more buffers
- Need buffer manipulations to display the correct order.

Basic Encoder Block Diagram



Use reconstructed error in the loop to prevent drifting.
Original input is not available to the decoder.
Need a buffer to keep the reference frame.

Basic Decoder Block Diagram



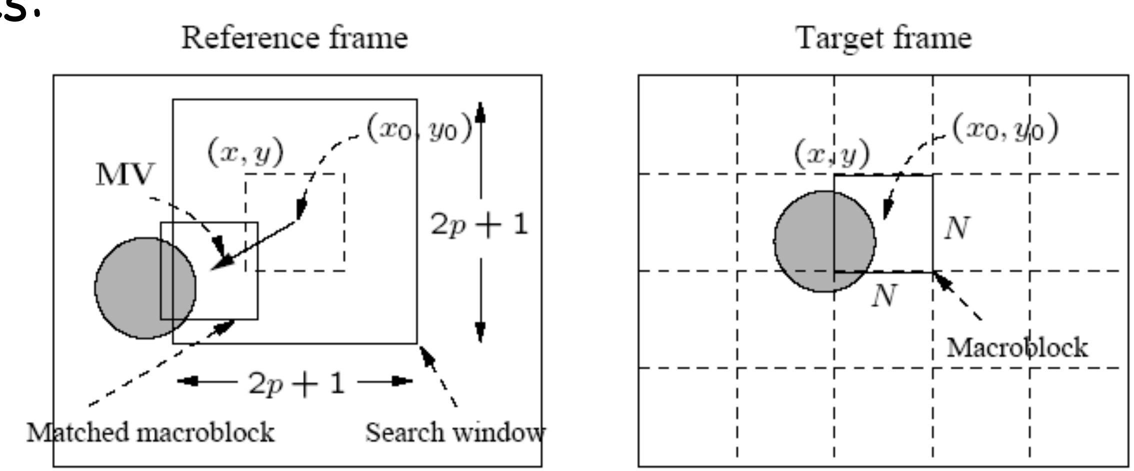
- Decoder is simpler than the encoder:
  - No need to do motion estimation.

Motion Estimation - Revisit

- Formulation:
- Find (i, j) in a search window (-p, p) that minimizes

$$e(i,j) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \left| C(x+k,\ y+l) - R(x+i+k,\ y+j+l) \right|^z$$

- Mean square error (MSE)
  - If z=2
- Mean absolute distance (MAD):
  - If z = 1.
- # of search candidates: (2p+1) x (2p + 1)



MAD-based Motion Estimation

$$MAD(i,j) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} |C(x+k, y+l) - R(x+i+k, y+j+l)|$$

N – size of the macroblock,

k and l – indices for pixels in the macroblock,

i and j – horizontal and vertical displacements,

$C(x+k, y+l)$ – pixels in macroblock in Target frame,

$R(x+i+k, y+j+l)$ – pixels in macroblock in Reference frame.

- Objective
  - Find vector (*i, j*) as the motion vector **MV** = (**u**,**v**), such that *MAD*(*i*,*j*) is minimum

$$(u, v) = [\ (i,j)\ |\ MAD(i,j) \ \text{is minimum}, \ i \in [-p,p], \ j \in [-p,p]\ ]$$

Naive Method

- **Sequential search (Full search):**
  - sequentially search the whole $(2p+1)$ $(2p+1)$ window in the Reference frame
    - a macroblock centered at each of the positions within the window is compared to the macroblock in the Target frame, pixel by pixel
    - respective *MAD* is derived
    - vector (i, j) that offers the least *MAD* is designated as the **MV** (u, v) for the macroblock in the target frame

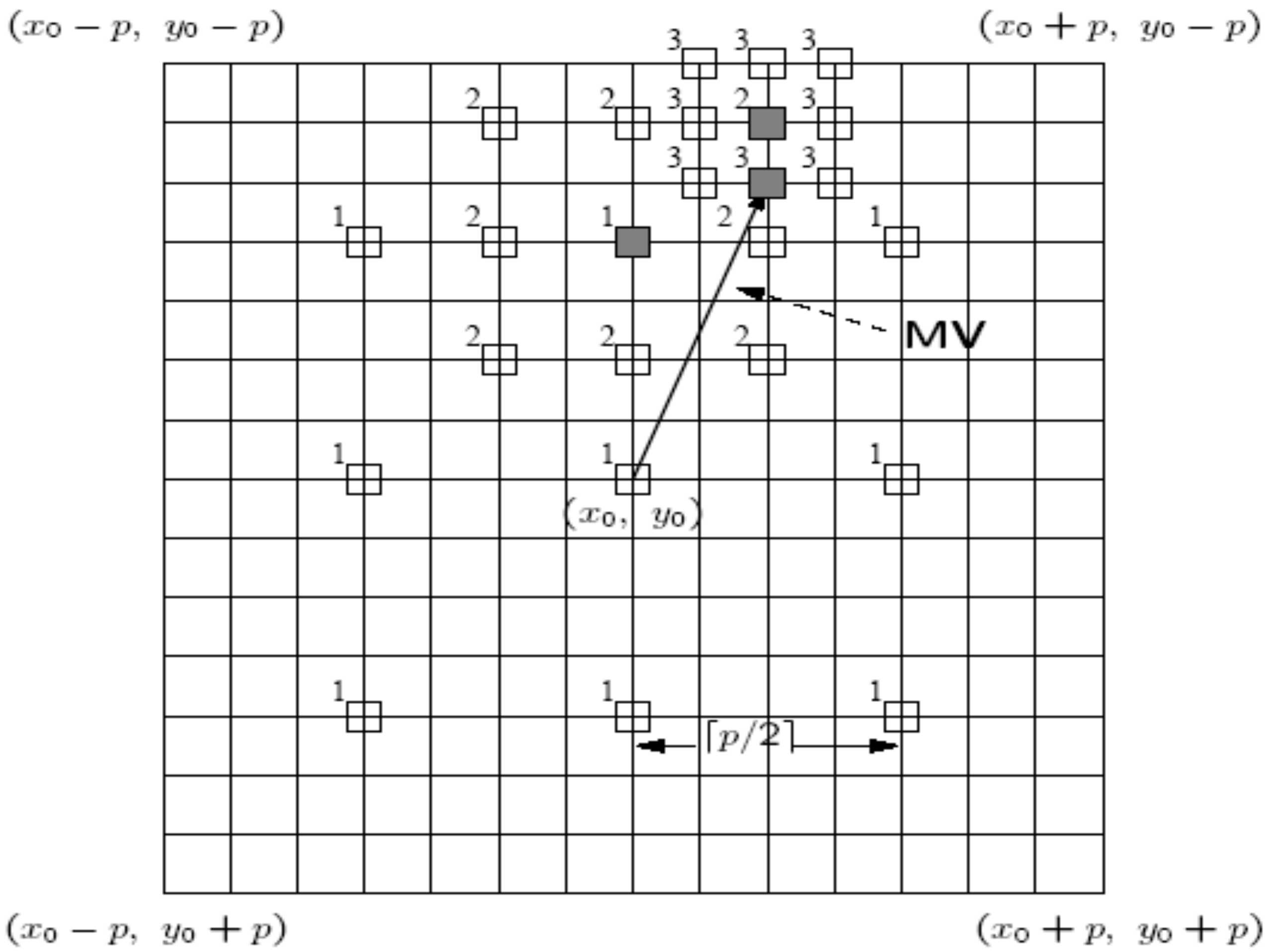
Fast Motion Estimation

- Full-search motion estimation is time consuming:
  - Each (i, j) candidate: N^2 summations
  - If search window size is W^2, need $W^2 \times N^2$ comparisions / MB
    - W=2p+1=31, N=16: ➔ 246016 comparisons / MB !
    - Each comparison three operations (subtraction, absolute value, addition)
- Fast motion estimation is desired:
  - Lower the number of search candidates
  - Many methods

2-D Log Search

- **Logarithmic search**:
  - a cheaper version
  - *suboptimal* but still usually effective.
- Procedure – similar to a binary search
  - Initially, only nine locations in the search window are used as seeds for a MAD-based search; marked as `1'.
  - After the one that yields the minimum *MAD* is located, the center of the new search region is moved to it and the step-size ("offset") is reduced to half.
  - In the next iteration, the nine new locations are marked as `2', and so on.

Log Search



$(x_0 - p,\ y_0 - p)$
$(x_0 + p,\ y_0 - p)$
3 3 3
2 2 3 2 3
3 3 3
1 2 1 2 1
MV
2 2 2
1 1 1
$(x_0,\ y_0)$
1 1 1
$\lceil p/2 \rceil$
$(x_0 - p,\ y_0 + p)$
$(x_0 + p,\ y_0 + p)$

Computations

- W=2p+1=31, N=16 (p=15)

$$\left(8 \cdot (\lceil \log_2 p \rceil + 1) + 1\right) \cdot N^2$$

- 10496 Comparison per Macroblock

Hierarchical Search

- **Hierarchical search**:
  - $W^2 \times N^2$: Comparison Per macroblock for sequential search
  - The search can benefit from a hierarchical (multiresolution) approach in which initial estimation of the motion vector can be obtained from images with a significantly reduced resolution.
  - Since the size of the macroblock is smaller and p can also be proportionally reduced, the number of operations required is greatly reduced.

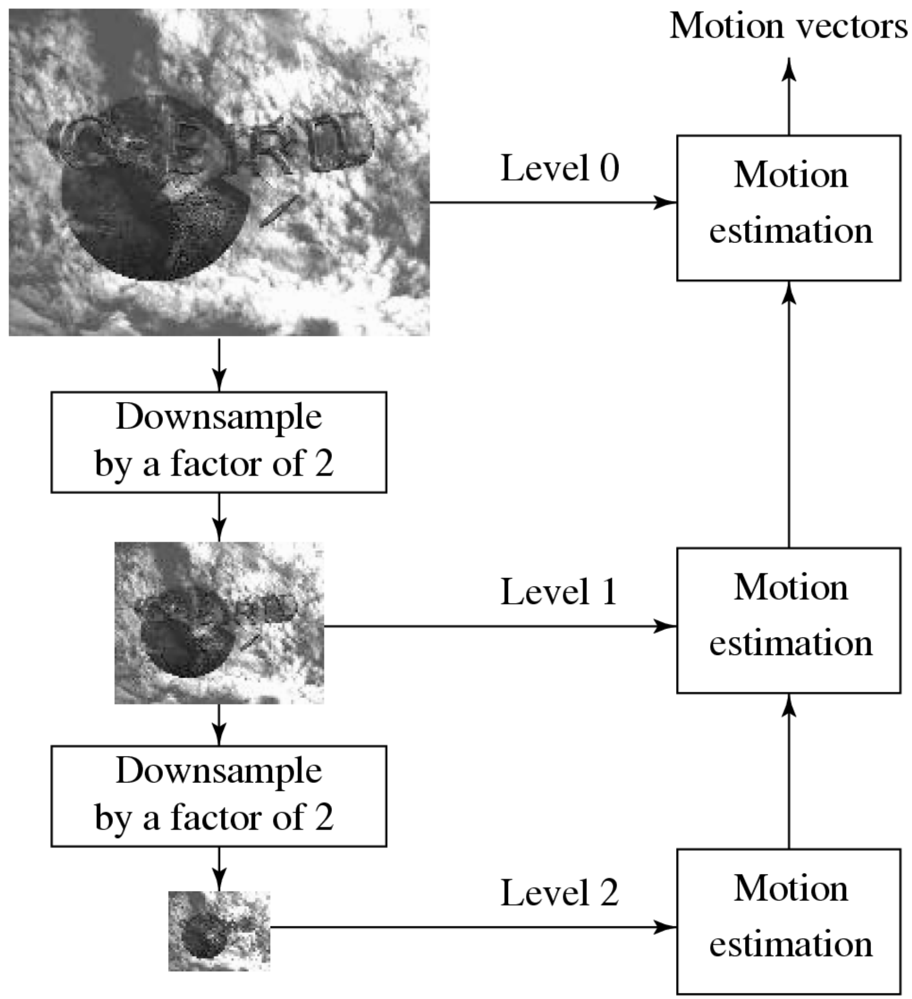


Fig. 10.3: A Three-level Hierarchical Search for Motion Vectors.

Hierarchical Search (Cont'd)

- Given the estimated motion vector (u^k, v^k) at Level k, a 3 x 3 neighborhood centered at $(2 \cdot u^k, 2 \cdot v^k)$ at Level $k - 1$ is searched for the refined motion vector.
- The refinement is such that at Level $k - 1$ the motion vector (u^{k-1}, v^{k-1}) satisfies:
  - $(2u^k - 1 \leq u^{k-1} \leq 2u^k + 1, 2v^k - 1 \leq v^{k-1} \leq 2v^k + 1)$
- Let $(x^k{}_0, y^k{}_0)$ denote the center of the macroblock at Level k in the target frame. The procedure for hierarchical motion vector search for the macroblock centered at $(x^0{}_0, y^0{}_0)$ in the Target frame can be outlined as follows:

PROCEDURE 10.3 Motion-vector:hierarchical-search

```
BEGIN
    // Get macroblock center position at the lowest resolution Level k
    x^k_0 = x^0_0 /2^k ;      y^k_0 = y^0_0 / 2^k;
    Use Sequential (or 2D Logarithmic) search method to get initial estimated
    MV(u^k, v^k) at Level k;
    WHILE last ≠ TRUE
    {
          Find  one of the nine macroblocks that yields minimum MAD at Level
          k − 1 centered at
          (2(x^k_0+u^k)− 1 ≤ x ≤  2(x^k_0+u^k) + 1; 2(y^k_0+v^k)−1 ≤ y ≤ 2(y^k_0 +v^k) + 1 );
          IF k = 1 THEN last = TRUE;
          k = k − 1;
          Assign (x^k_0; y^k_0 ) and (u^k, v^k) with the new center location and MV;
    }
END
```

Computations

- W=2p+1=31, N=16 (p=15)
- Reduced size

$$\left[\left(2\left\lceil\frac{p}{4}\right\rceil+1\right)^2\left(\frac{N}{4}\right)^2+9\left(\frac{N}{2}\right)^2+9N^2\right]$$

- 4176 Comparison per Macroblock

Intra-frame (I-frame) Coding

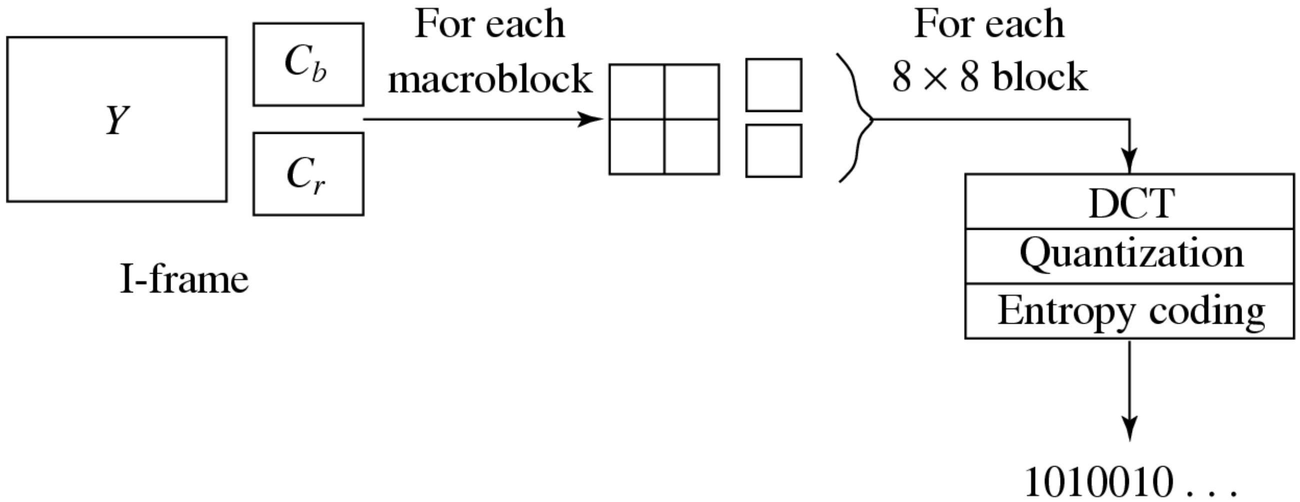


Fig. 10.5: I-frame Coding.

- **Macroblocks** are of size 16 x 16 pixels for the Y frame, and 8 x 8 for Cb and Cr frames, since 4:2:0 chroma subsampling is employed. A macroblock consists of four Y, one Cb, and one Cr 8 x 8 blocks.
- For each 8 x 8 block a DCT transform is applied, the DCT coefficients then go through quantization zigzag scan and entropy coding.

Inter-frame (P-frame) Predictive Coding

- Figure 10.6 shows the H.261 P-frame coding scheme based on motion compensation:
  - For each macroblock in the Target frame, a motion vector is allocated by one of the search methods discussed earlier.
  - After the prediction, a *difference macroblock* is derived to measure the *prediction error*.
  - Each of these 8 x 8 blocks go through DCT, quantization, zigzag scan and entropy coding procedures.

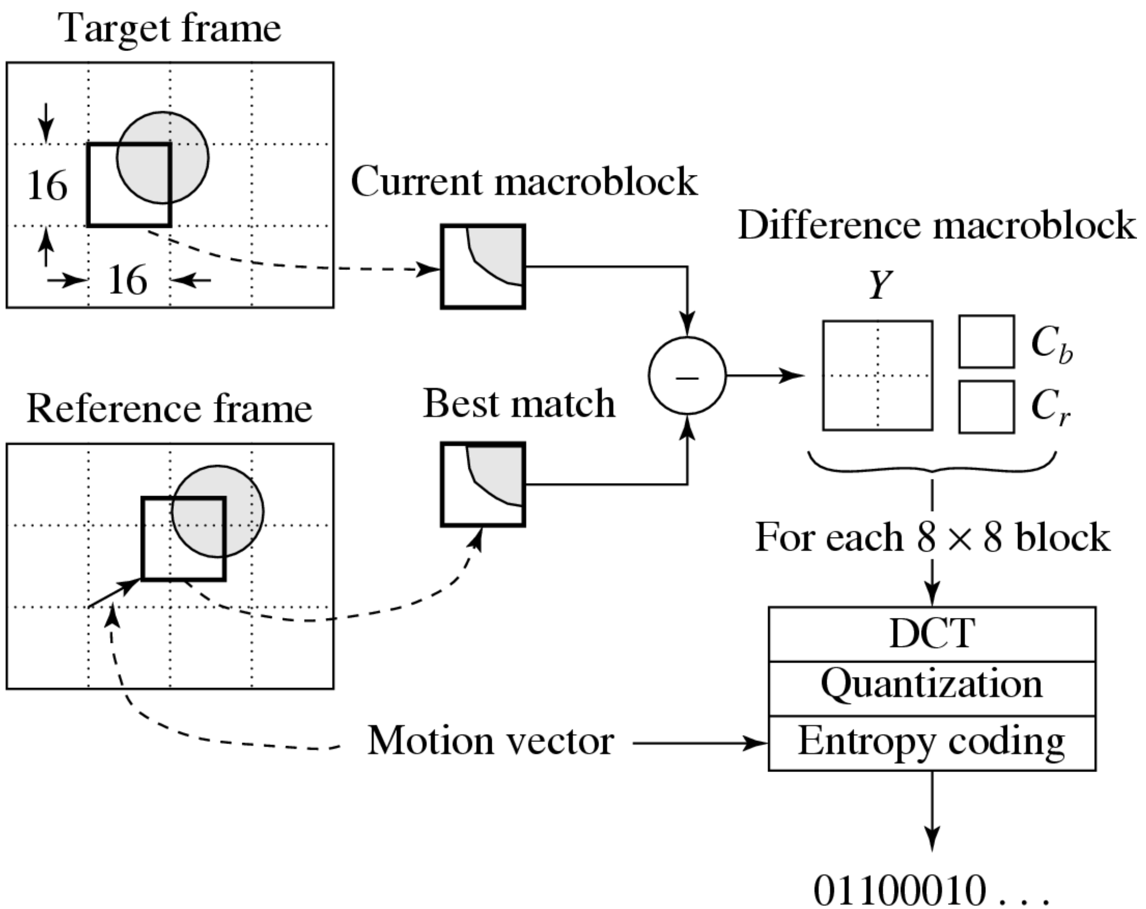


Fig. 10.6: H.261 P-frame Coding Based on Motion Compensation

- The P-frame coding encodes the difference macroblock (not the Target macroblock itself).
- Sometimes, a good match cannot be found, i.e., the prediction error exceeds a certain acceptable level.
  - The MB itself is then encoded (treated as an Intra MB) and in this case it is termed a *non-motion compensated MB*.
- For a motion vector, the difference $\mathbf{MVD}$ is sent for entropy coding:

$$\mathbf{MVD} = \mathbf{MV}_{\mathbf{Preceding}} - \mathbf{MV}_{\mathbf{Current}} \qquad (10.3)$$

Quantization in H.261

- The quantization in H.261 uses a constant *step_size*, for all DCT coefficients within a macroblock.
- If we use *DCT* and *QDCT* to denote the DCT coefficients before and after the quantization, then for DC coefficients in Intra mode:

$$QDCT = round\left(\frac{DCT}{step\_size}\right) = round\left(\frac{DCT}{8}\right) \qquad (10.4)$$

for all other coefficients:

$$QDCT = \left\lfloor \frac{DCT}{step\_size} \right\rfloor = \left\lfloor \frac{DCT}{2 * scale} \right\rfloor \qquad (10.5)$$

scale — an integer in the range of [1, 31].

Further Exploration

- **Textbook Chapter 10**
- **Other sources**
  - *A Java H.263 decoder* by A.M. Tekalp
  - *Digital Video and HDTV Algorithms and Interfaces* by C.A. Poynton
  - *Image and Video Compression Standards* by V. Bhaskaran and K. Konstantinides
  - *Video Coding: An introduction to standard codecs* by M. Ghanbari
  - *Video processing and communications* by Y. Wang et al.