ECE462 – Lecture 21

VIDEO COMPRESSION TECHNIQUES
H.261 / H.263 Standards

10.4 H.261

- **H.261**: An earlier digital video compression standard, its principle of MC-based compression is retained in all later video compression standards.
  - The standard was designed for videophone, video conferencing and other audiovisual services over ISDN.
  - The video codec supports bit-rates of $p \times 64$ kbps, where p ranges from 1 to 30 (Hence also known as $p * 64$).
  - Require that the delay of the video encoder be less than 150 msec so that the video can be used for real-time bi-directional video conferencing.

ITU Recommendations & H.261 Video Formats

- H.261 belongs to the following set of ITU recommendations for visual telephony systems:

  1. H.221 — Frame structure for an audiovisual channel supporting 64 to 1,920 kbps.
  2. H.230 — Frame control signals for audiovisual systems.
  3. H.242 — Audiovisual communication protocols.
  4. H.261 — Video encoder/decoder for audiovisual services at $p \times 64$ kbps.
  5. H.320 — Narrow-band audiovisual terminal equipment for $p \times 64$ kbps transmission.

Table 10.2 Video Formats Supported by H.261

Video format	Luminance image resolution	Chrominance image resolution	Bit-rate (Mbps) (if 30 fps and uncompressed)	H.261 support
QCIF	176 × 144	88 × 72	9.1	required
CIF	352 × 288	176 × 144	36.5	optional

QCIF: Quarter Common Intermidiate Format

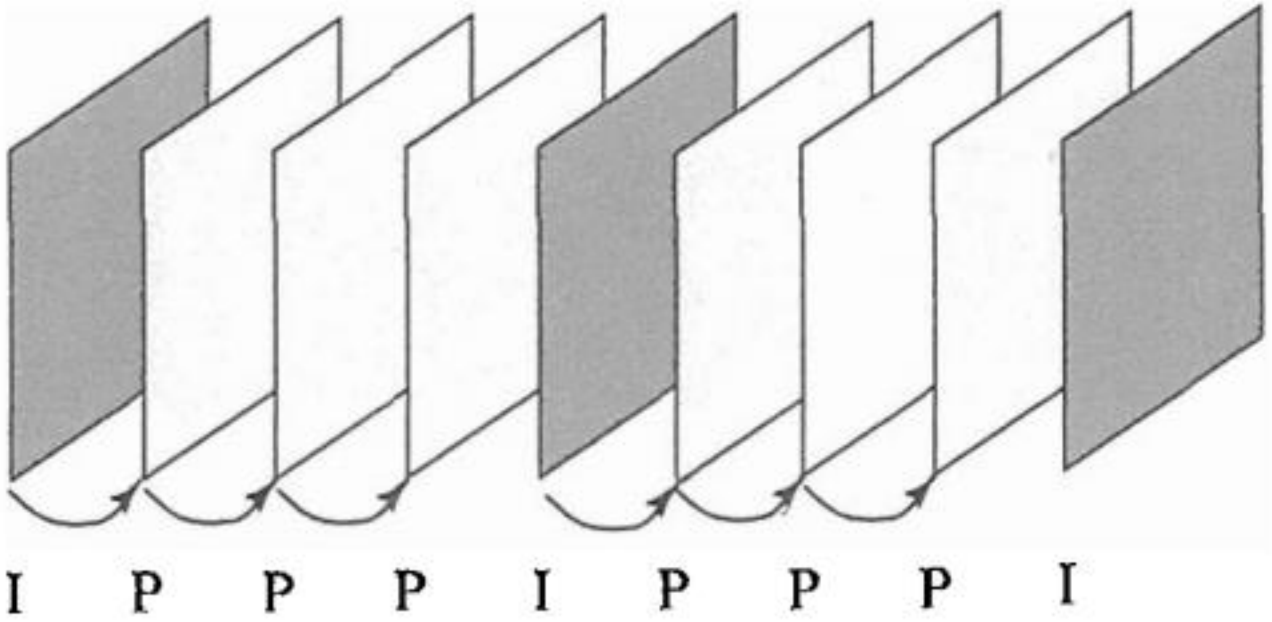


Fig. 10.4: H.261 Frame Sequence.

H.261 Frame Sequence

- Two types of image frames are defined: Intra-frames (**I-frames**) and Inter-frames (**P-frames**):
  - I-frames are treated as independent images. Transform coding method similar to JPEG is applied within each I-frame, hence "Intra".
  - P-frames are not independent: coded by a forward predictive coding method (prediction from a previous P-frame is allowed — not just from a previous I-frame).
  - **Temporal redundancy removal** is included in P-frame coding, whereas I-frame coding performs only **spatial redundancy removal**.
  - To avoid propagation of coding errors, an I-frame is usually sent a couple of times in each second of the video.
- Motion vectors in H.261 are always measured in units of full pixel and they have a limited range of ± 15 pixels, i.e., $p = 15$.

Intra-frame (I-frame) Coding

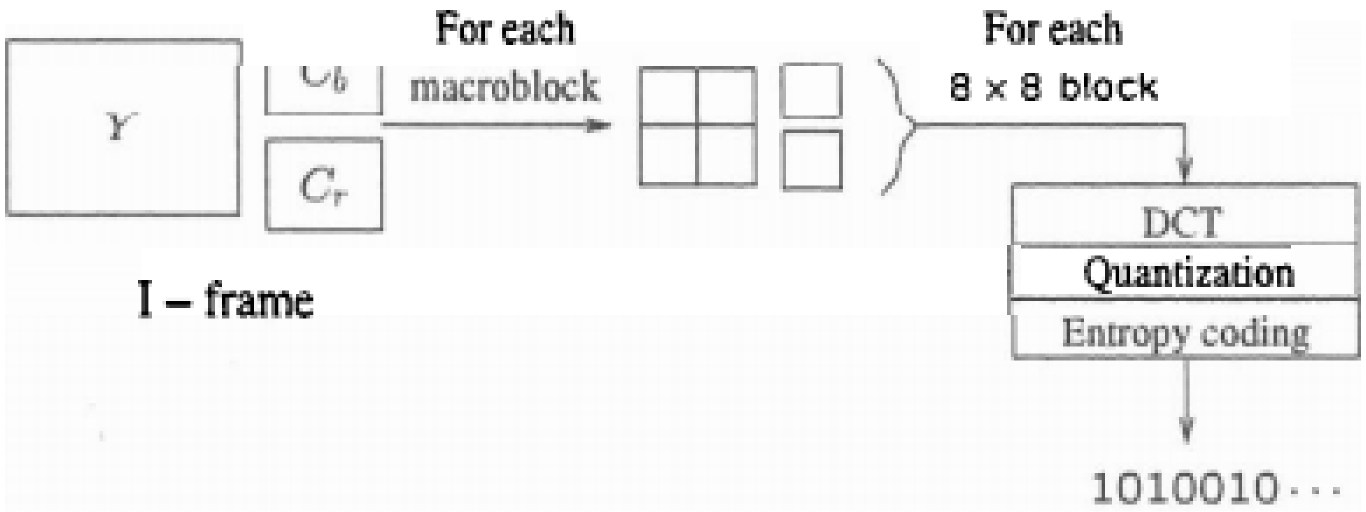


Fig. 10.5: I-frame Coding.

- **Macroblocks** are of size 16 × 16 pixels for the Y frame, and 8 × 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 × 8 blocks.
- For each 8 × 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 8×8 blocks go through DCT, quantization, zigzag scan and entropy coding procedures.

- 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 motion vector, the difference $\mathbf{MVD}$ is sent for entropy coding:

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

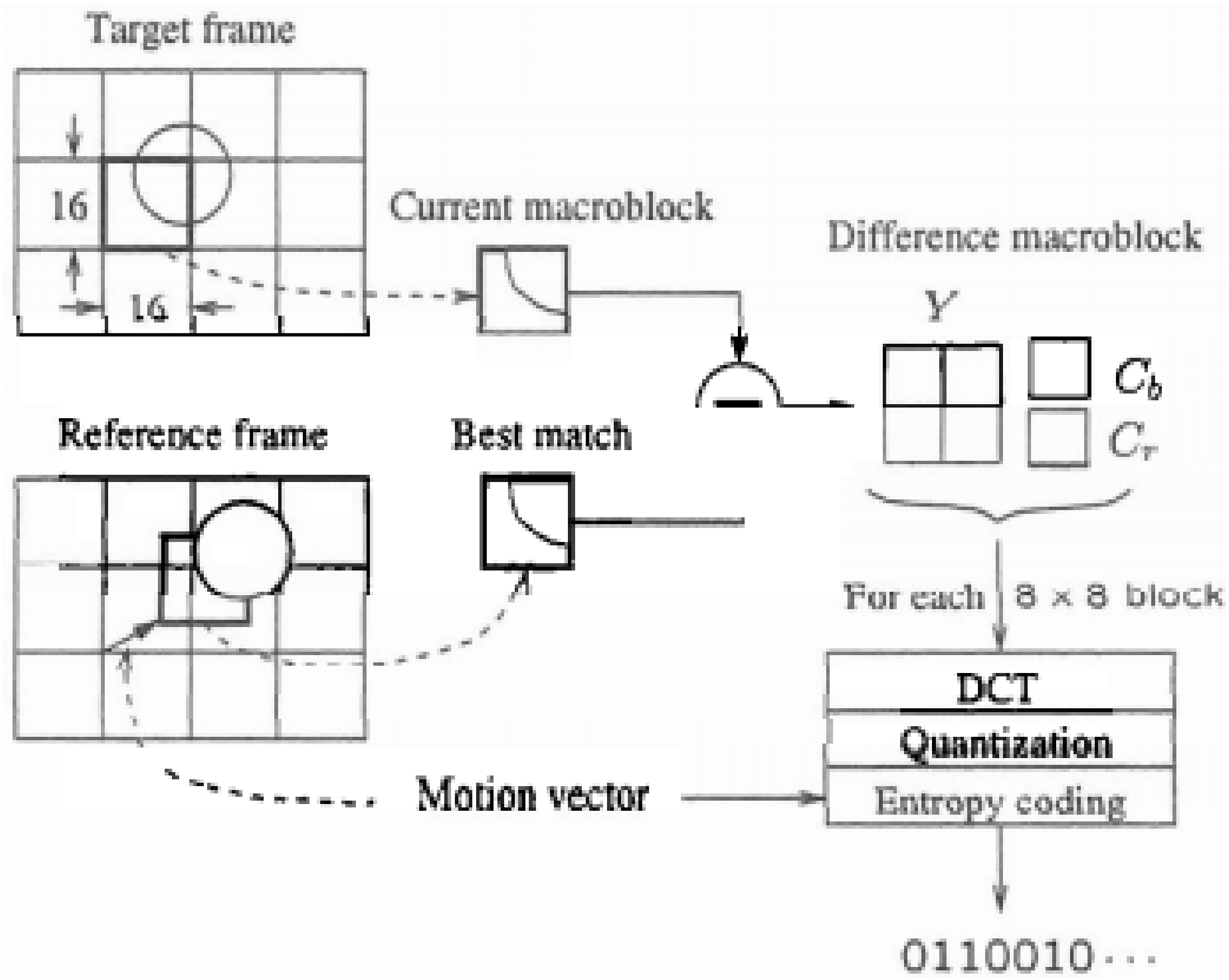


Fig. 10.6: H.261 P-frame Coding Based on Motion Compensation.

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].

H.261 Encoder and Decoder

- Fig. 10.7 shows a relatively complete picture of how the H.261 encoder and decoder work.

  A scenario is used where frames I, P_1, and P_2 are encoded and then decoded.

- Note: decoded frames (not the original frames) are used as reference frames in motion estimation.

- The data that goes through the observation points indicated by the circled numbers are summarized in Tables 10.3 and 10.4.

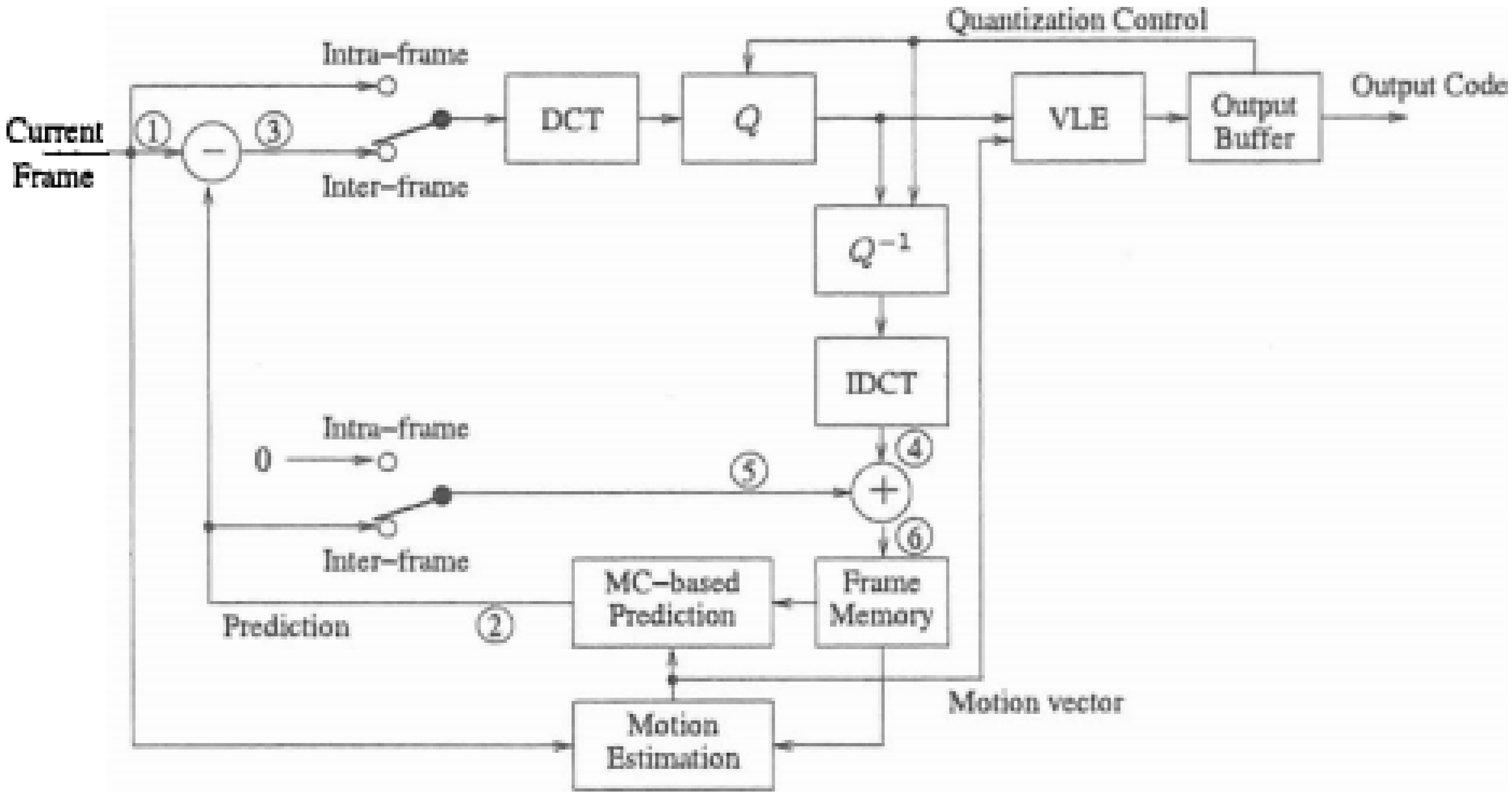
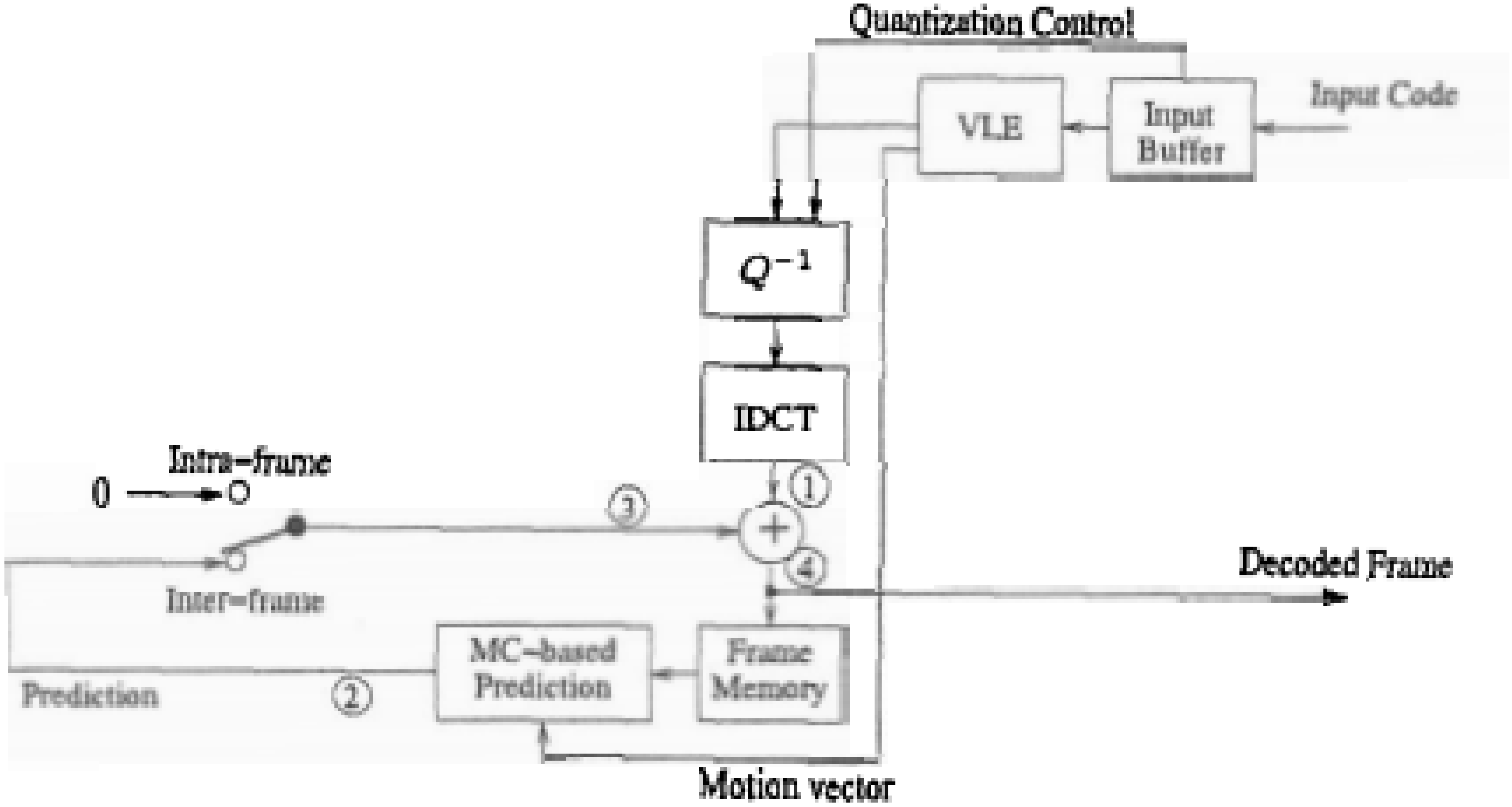

Quantization Control
Intra-frame
DCT
Q
VLE
Output Buffer
Output Code
Current Frame
1
3
Inter-frame
Q^{-1}
IDCT
Intra-frame
0
5
4
6
Inter-frame
MC-based Prediction
Frame Memory
Prediction
2
Motion vector
Motion Estimation

Table 10.3: Data Flow at the Observation Points in H.261 Encoder

Current Frame	Observation Point					
	1	2	3	4	5	6
I	I			$\tilde{I}$	0	$\tilde{I}$
P_1	P_1	P_1'	D_1	$\tilde{D}_1$	P_1'	$\tilde{P}_1$
P_2	P_2	P_2'	D_2	$\tilde{D}_2$	P_2'	$\tilde{P}_2$

Table 10.4: Data Flow at the Observation Points in H.261 Decoder

Current Frame	Observation Point			
	1	2	3	4
I	$\tilde{I}$		0	$\tilde{I}$
P_1	$\tilde{D}_1$	P_1'	P_1'	$\tilde{P}_1$
P_2	$\tilde{D}_2$	P_2'	P_2'	$\tilde{P}_2$

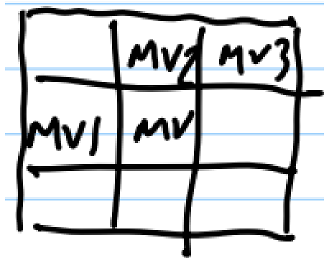

Quantization Control
VLE
Input Buffer
Input Code
Q^{-1}
IDCT
Intra-frame
0
1
3
+
4
Decoded Frame
Inter-frame
MC-based Prediction
Frame Memory
Prediction
2
Motion vector

(b) Decoder

The H.263 Standard

- Similar to H.261
    - It supports a few more video formats (QCIF, CIF, GCIF, 16CIF)
    - Motion compensation slightly different than H.261
    - Supports half pixel prediction based on bilinear interpolation
    - Supports forward and backward prediction (PB frames)

<u>Motion Compensation H.263</u>



- MV(u,v): Motion vector for current macroblock
- $MV_1(u_1,v_1)$, $MV_2(u_2,v_2)$, $MV_3(u_3,v_3)$ are motion vector for neighboring macroblocks

- Obtain MV,MV_1,MV_2,MV_3 as in H.261
- Calculate: u_p = median (u_1,u_2,u_3)
  v_p=median (v_1,v_2,v_3)

For MV code $\Delta u = u - up, \quad \Delta v = v - vp$

- Half pixel based motion prediction
  - Upscale image by 2 using bilinear interpolation before you calculate motion vectors