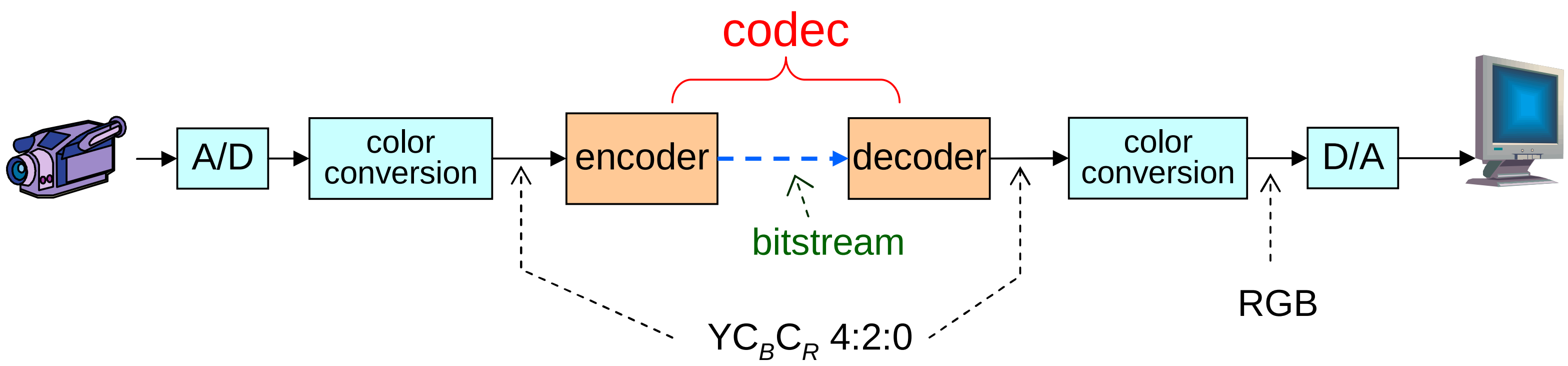
MPEG-4 Simple Profile Video Codec Introduction

National Chiao Tung University
Chun-Jen Tsai

3/3/2011

Video Systems

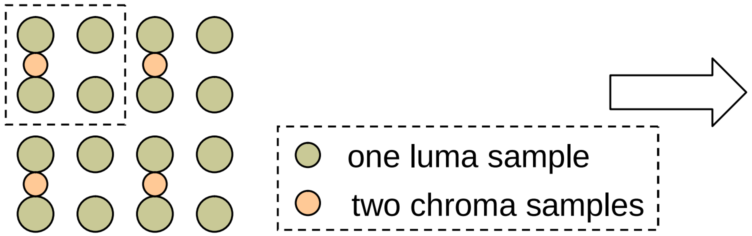
- A complete end-to-end video system is as follows



- Most popular codecs are from MPEG:
  - MPEG-1 video,1992, 1 ~ 2 mbps
  - MPEG-2 video, 1994, 2 ~ 20 mbps
  - MPEG-4 video
    - **Simple Profile, 1999, 64kbps ~ 1.5 mbps**
    - AVC/H.264, 2003, 32 kbps ~ 20 mbps

Video Frame Representation

- Video frame are represented in YC_BC_R 4:2:0 format
  - *RGB* format is well-known, but not suitable for video coding
  - YC_BC_R is used for video coding because:
    - Color components can be subsampled easily
    - Human eyes are less sensitive to color gradient
  - Some people refer to YC_BC_R space as *YUV* color space
- 4:2:0 stands for color subsampling



An *.yuv file stores video data frame-by-frame. Each frame stores complete luma sample before chroma samples.

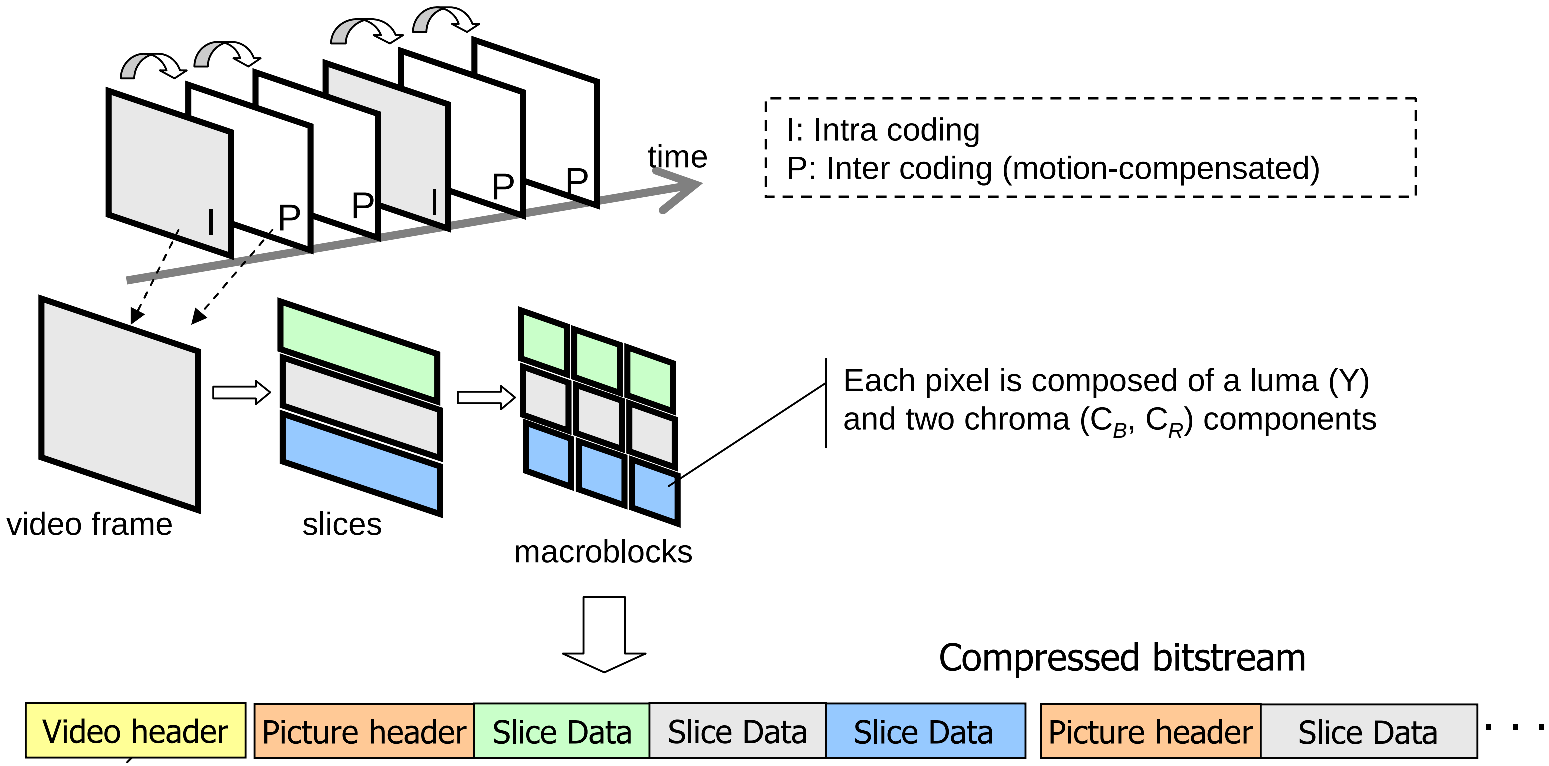
Y: luma; C_B, C_R: chroma
For more info., see Charles Poynton's website: http://www.poynton.com/

Color Space Conversion

- $RGB \rightarrow YC_BC_R$:
  - $Y = \alpha_{\text{red}} \times Red + \alpha_{\text{green}} \times Green + \alpha_{\text{blue}} \times Blue$
  - $C_B = (Blue - Y) / (2 - 2 \times \alpha_{\text{blue}})$
  - $C_R = (Red - Y) / (2 - 2 \times \alpha_{\text{red}})$
- $YC_BC_R \rightarrow RGB$:
  - $Red = C_R \times (2 - 2 \times \alpha_{\text{red}}) + Y$
  - $Green = (\alpha_{\text{blue}} \times Blue - \alpha_{\text{red}} \times Red) / \alpha_{\text{green}}$
  - $Blue = C_B \times (2 - 2 \times \alpha_{\text{blue}}) + Y$
- Coefficients table:

Recommendation	α_{red}	α_{green}	α_{blue}
BT-601	0.2990	0.5870	0.1140
BT-709	0.2126	0.7152	0.0722

From Video Frame to Bitstream



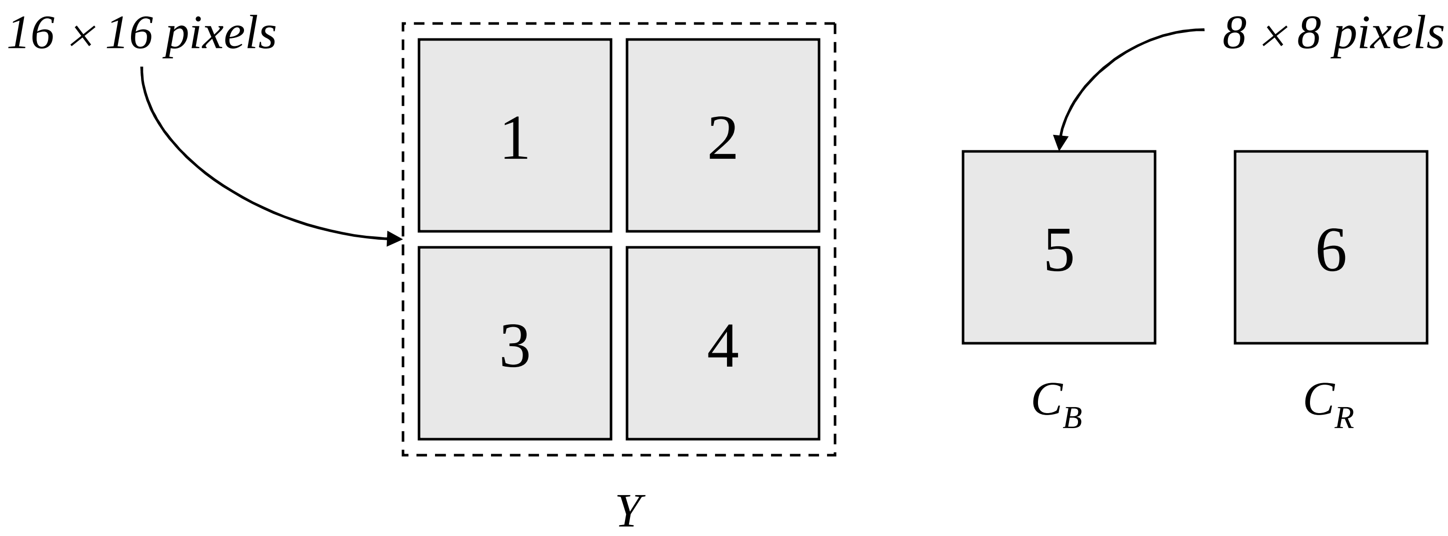
In MPEG-4, this is composed of three small headers. Among which the only important one (the third one) is the Video Object Layer (VOL) header.

Components of MPEG-4 Video Codec

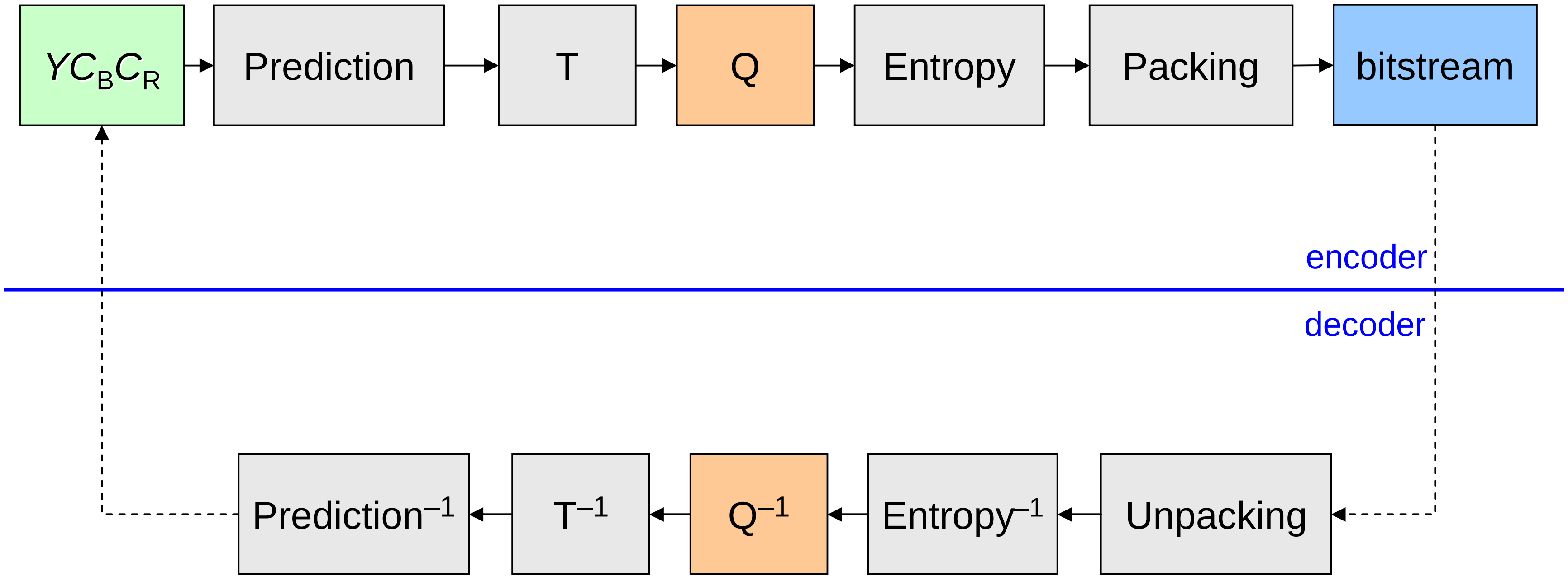
- MPEG-4 Simple Profile (SP) video codec is a “block-based motion-compensated transform” codec
- This type of codec is composed of the following modules:
  - Predictive coder (loss-less)
  - Transform coder (loss-less, theoretically)
  - Quantizer (lossy)
  - Entropy coder (loss-less)

MPEG-4 Macroblocks (MB)

- A picture is divided into macroblocks (MB) before coding. Each MB is composed of 16×16 pixels
- The smallest coding unit is a 8×8 block. There are 6 blocks per macroblocks



Macroblock Coding Decoding Loop



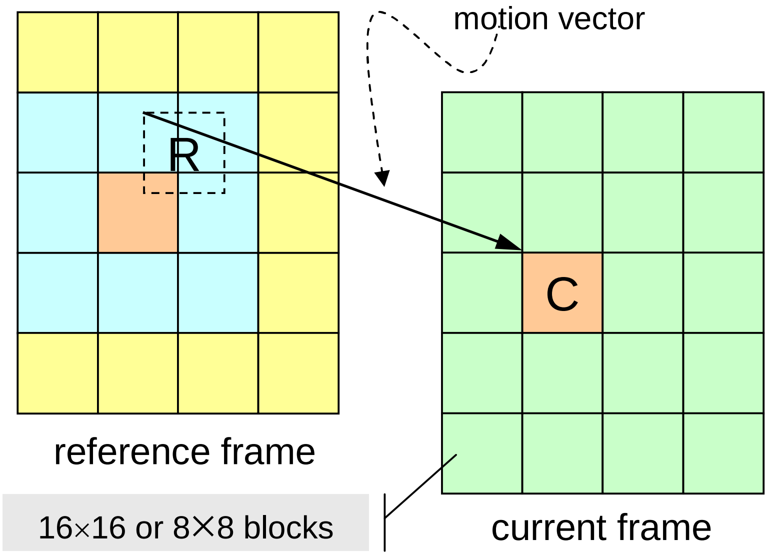
* T – transform; Q – Quantization

Prediction Modes

- Each macroblock is encoded using one of the following predictive coding mode:
  - Skip
  - Intra coding (*similar to JPEG*)
  - Intra coding with quantizer adjustment
  - Inter coding (*using Motion Compensated Prediction*)
  - Inter coding with quantizer adjustment
  - Inter coding with 4 motion vectors

Motion Estimation/Mode Decision

- The most important prediction coding technique for video is called motion-compensated prediction:

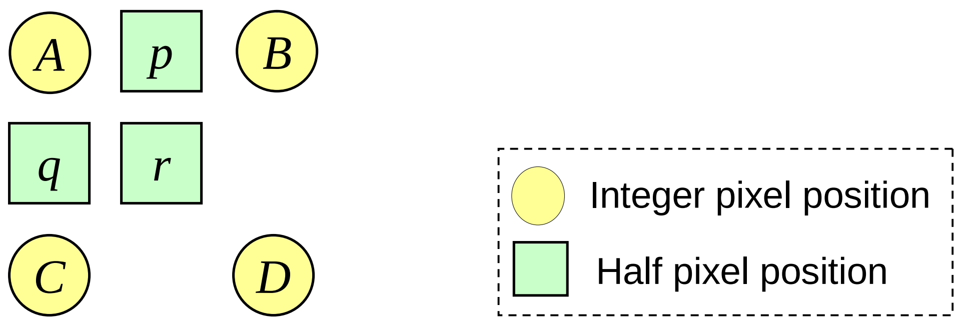


'R' is a predictor block;
'C' is the current block for coding

- The differences of pixel values, C – R, are encoded into the bitstream after DCT transform (to be discussed later)

½-Pixel Motion Compensation

- Since MV resolution is half-a-pixel, when the MV is, say, (-10.5, 4.5), we must “make up” the predictor block “R” by interpolation:

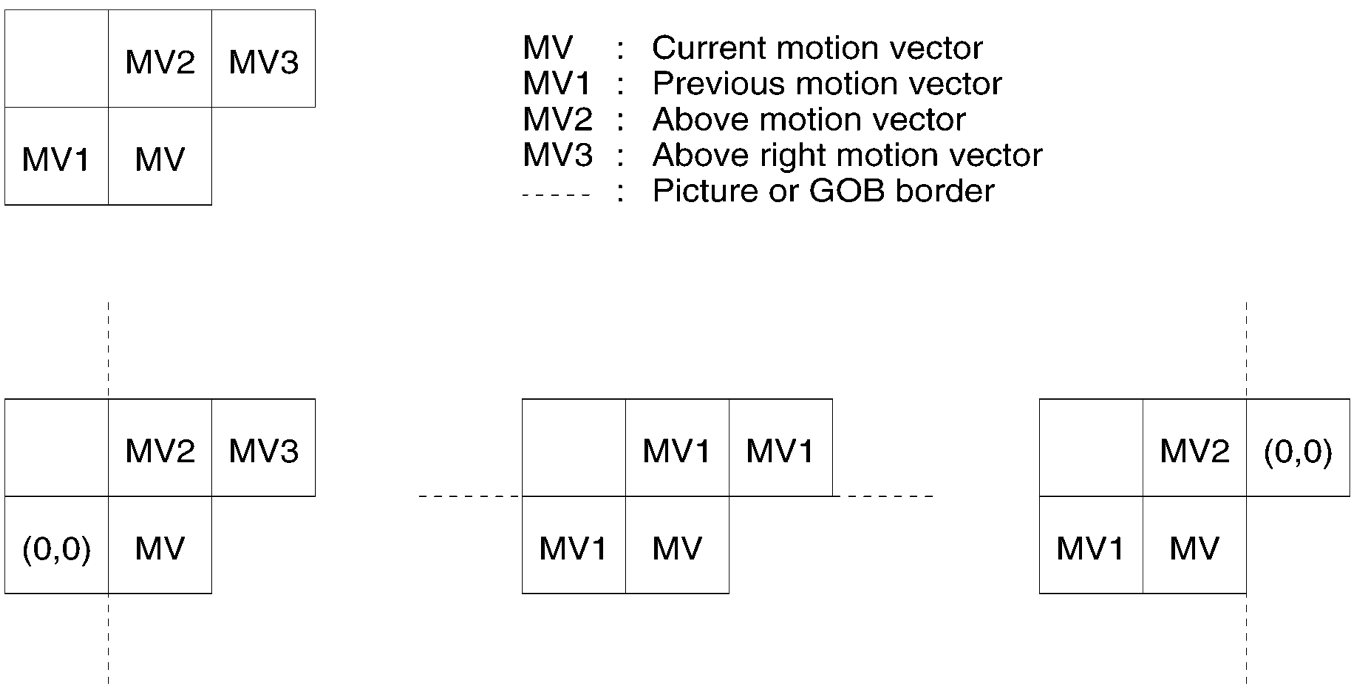


$$p = (A + B + 1 - \delta)/2$$
$$q = (A + C + 1 - \delta)/2$$
$$r = (A + B + C + D + 2 - \delta)/4$$

Note: $\delta = 0$ or 1, is a “rounding control” parameter.
δ is data-dependent and determined by the encoder.

Motion Vector Coding

- Motion vectors are also predictively coded
- The predictor is the median of MV1, MV2, and MV3



MPEG-4 Transform Coder

- 2D Discrete Cosine Transform (DCT) is used:
  - Forward transform (for encoder):

$$F(u,v) = C(u)C(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N}$$

  - Backward transform (for decoder):

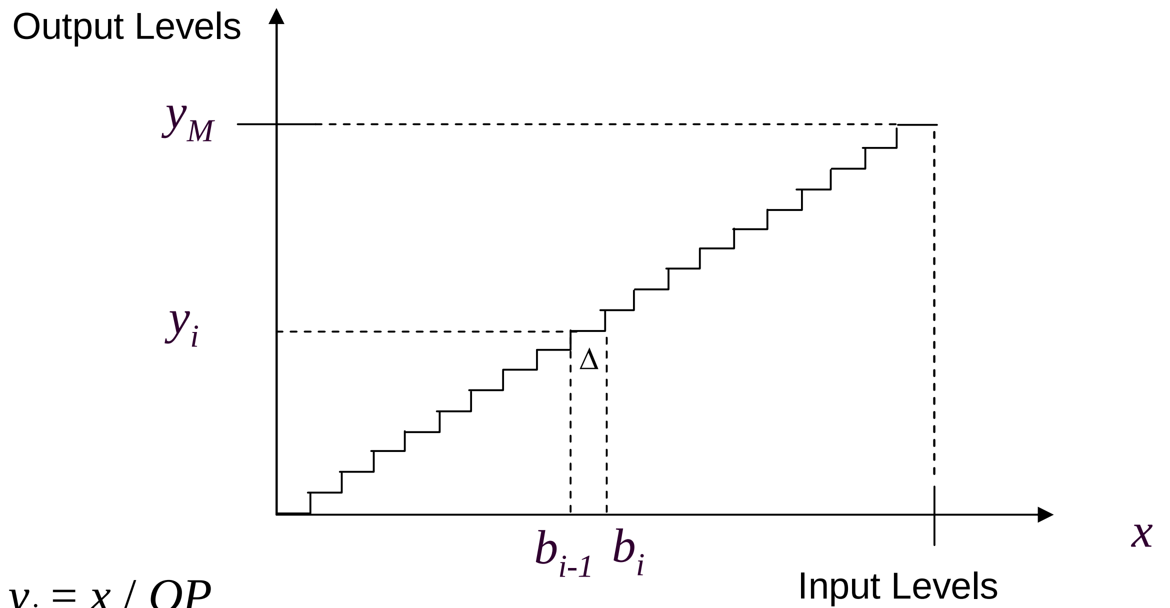
$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} C(u)C(v)F(u,v)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N}$$

Note: In both equations, N is the size of block (8 for MPEG-4 SP), and

$$C(t) = \begin{cases} \frac{1}{\sqrt{N}}, & t = 0 \\ \sqrt{\frac{2}{N}}, & t \neq 0 \end{cases}$$

Quantization Module

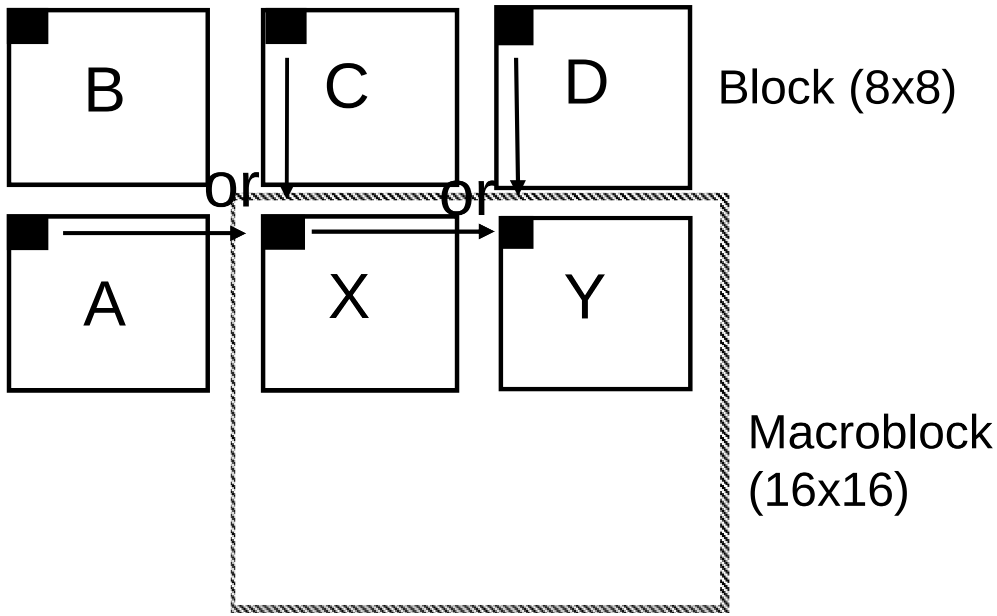
- The 1-D array of coefficients are quantized using a uniform quantizer (with quantizer stepsize $QP = \Delta$):



$y_i = x / QP$

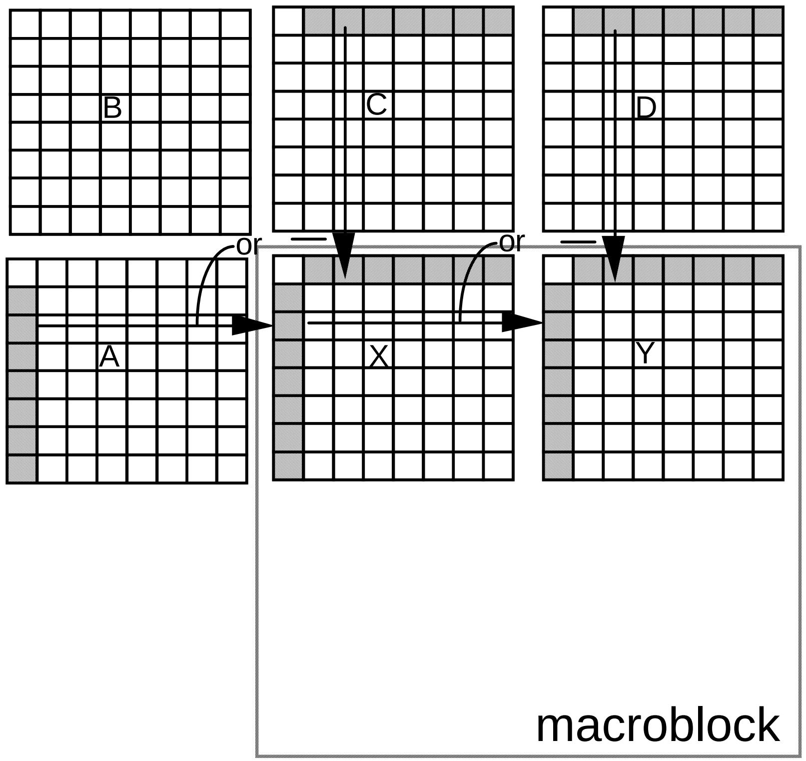
Adaptive DC Prediction

- Pick DC predictor based on gradients of the DC's:
  - If $(|DC_A - DC_B| < |DC_B - DC_C|)\ DC_X = DC_C$
  - else $DC_X = DC_A$



Adaptive AC Prediction

- Coefficients are predicted from previous coded blocks.
- The best direction is chosen based on the DC prediction.



DCT Coefficients Coding

- The DCT of the error residuals are coded using an entropy coder
- The entropy coder is composed of three steps:
  - Convert 2-D data to 1-D array of coefficients (zigzag scan)
  - Convert 1-D array of coefficients to 1-D array of symbols (run-length coding)
  - Variable-length coding (VLC) of the run-length symbols

Zigzag Scan

- Zigzag scan is used to map the 2-D array of DCT coefficients to an 1-D array:

0	1	5	6	14	15	27	28
2	4	7	13	16	26	39	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Entropy Coding Module

- Each nonzero coefficient in the 1-D array is converted to a (Last, Run, Level) symbol:
  - Last: is this the last non-zero coefficient?
  - Run: the number of zeros precede this coefficient
  - Level: the value of this coefficient
- These symbols are coded using variable length codes (VLC)

Coefficient Coding Example

78	0	0	0	0	0	0	0
-9	0	22	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	3	0	0	-9	0	0
0	8	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

zig-zag scan ⇒ 78, 0, -9, 0, 0, 0, 0, 22, 0, 0, … , 0, 0, -9

convert to symbols ⇒ (0, 0, 78), (0, 1, -9), (0, 4, 22), (0, 14, 8), (0, 0, 3), (1, 21, -9)

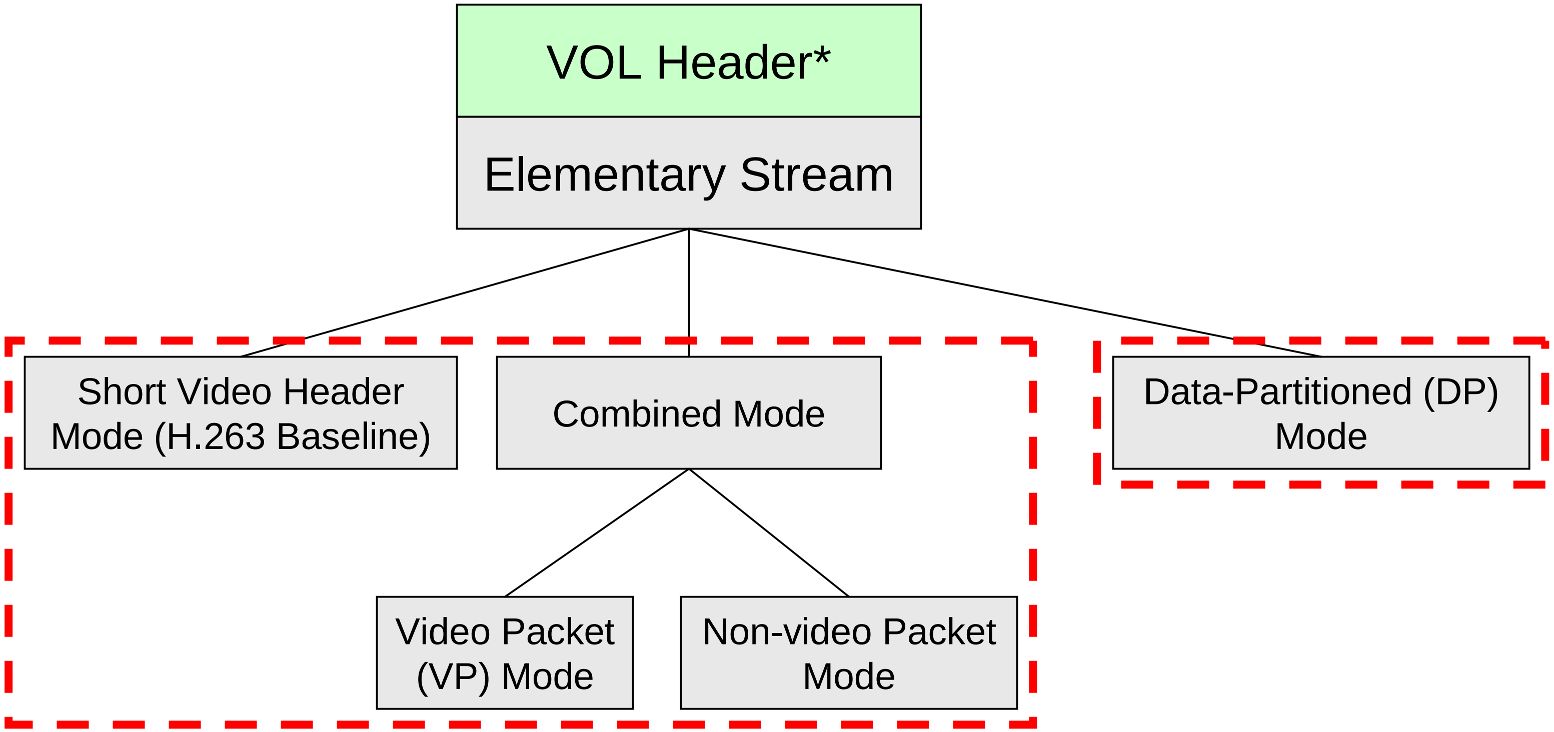
VLC coding ⇒ Convert symbols to VLC codes by table-lookup.

Last	Run	Level	Bits	VLC Code
0	0	1	3	10s
0	0	2	5	1111s
0	0	3	7	0101 01s
• • •				
1	0	1	5	0111s
1	0	2	10	0000 1100 1s
• • •				

Packing the Compressed Data

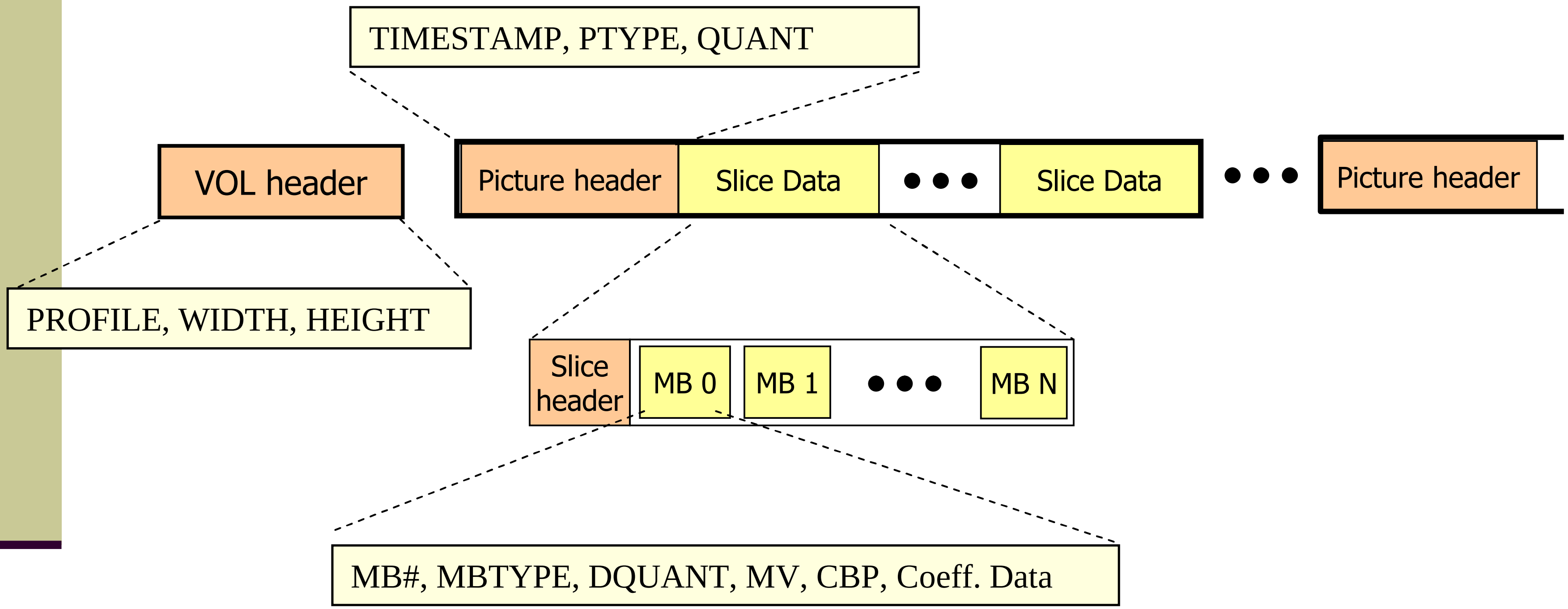
- In previous slides, we covered the core technologies inside a video codec. However, the compressed data has to be arranged into a bitstream, alone with some system information
- The packing mechanism is critical to the application scenario (e.g. video-over-IP)

MPEG-4 SP Bitstream Types



*Note: VOL header for “short video header mode” is an empty header

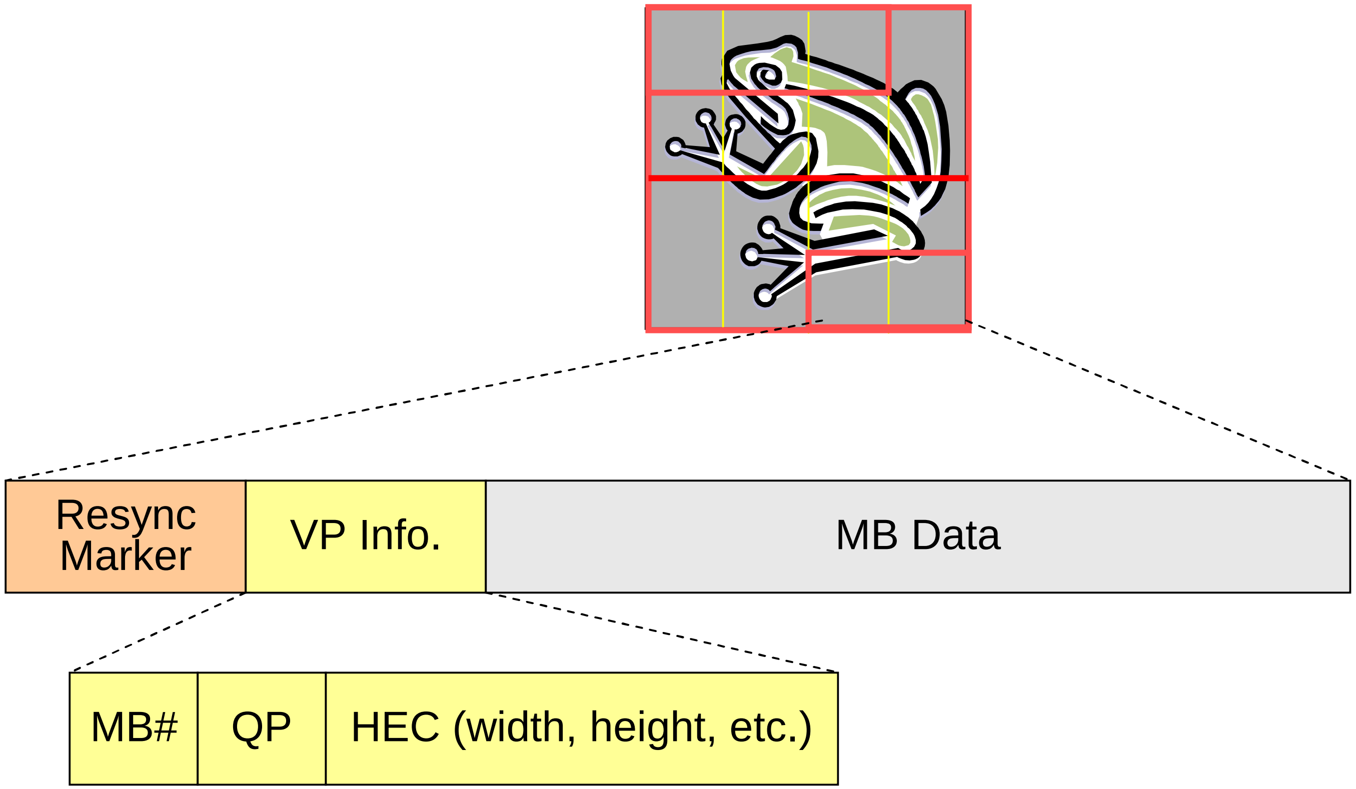
Combined Mode with VP Syntax



⇨ VLC is used to code these symbols

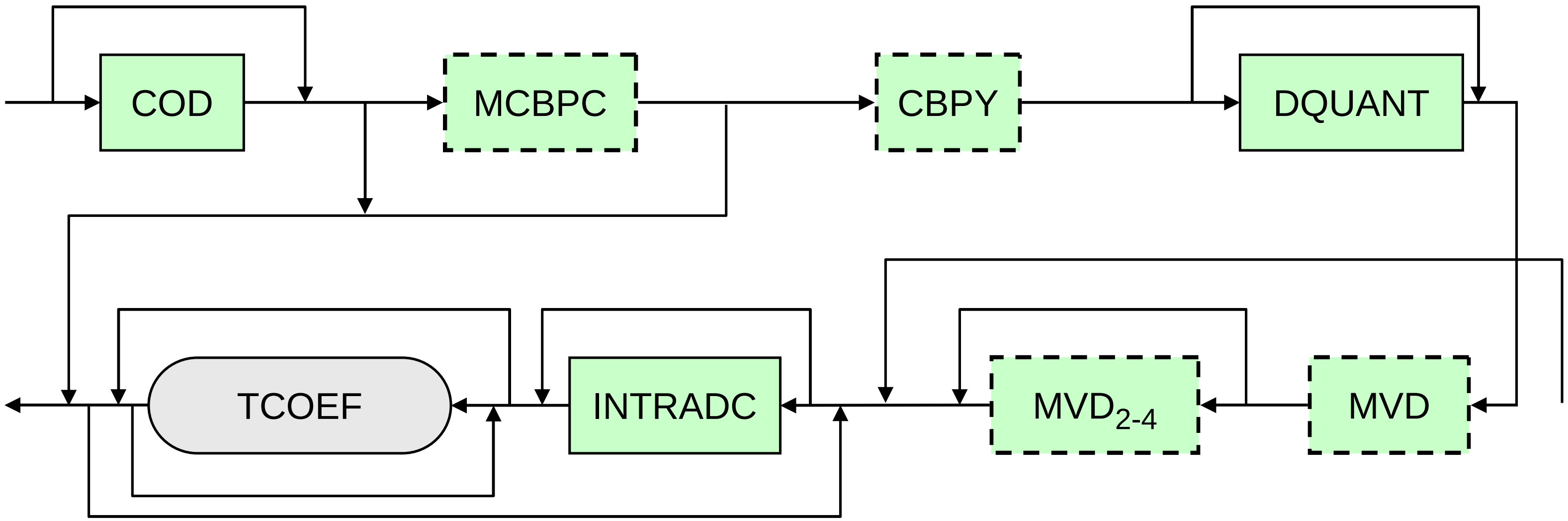
Video Packets (Slices)

- A video packet (slice) is a set of consecutive macroblocks in scan order:



Macroblock Syntax

- Each macroblock data contains the following information



* MCBPC combines “MB Prediction Type” and “chroma coded block pattern (CBP)” into one symbol; INTRADC is coded using 8-bit FLC, they are present for every blocks in an Intra MB