FORMATOS

DE AUDIO Y VIDEO.

Calidad de Aplicación

Objetivos

- Conocer y entender los formatos de transmisión para streaming de audio y vídeo.
- Conocer las formatos contenedores.
- Conocer los aspectos más importantes de calidad de aplicación de servicios streaming.

Competencias adquiridas

- Capacidad de entender y diseñar los formatos de transmisión de streaming que se aplican en la actualidad.
- Comprender la evaluación del funcionamiento de estos sistemas y su interacción con los niveles inferiores.
- Conocer y entender los formatos contendores de los sistemas audiovisuales.

Contenidos

- **UNIDAD DIDÁCTICA 3: FORMATOS DE AUDIO Y VIDEO**
  - AUDIO.
    1. Voz: G.711 … G.729.
    2. Encapsulado RTP.
  - **VIDEO.**
    1. **H.261, H.263 y H.264.**
    2. **Encapsulado RTP.**
  - CONTENEDORES.
    1.- MP3.
    2.- MPEG-4.
  - QoA. PSNR y PSQM, MOS.

Bibliografía

- *O. Hersent, D. Gurle IP Telephony: Packet-Based Multimedia Communications Systems Addison-Wesley, Hardcover, December 1999. ISBN 0201619105*

Standares ITU: P.800, P.860, P.861.

72G711 ITU-T Recommendation G.711: *Pulse Code Modulation* (PCM) of voice frequencies. 1972.

90G726 ITU-T Recommendation G.726: 40, 32, 24, 16 kbps *Adaptive Differential Pulse Code Modulation* (ADPCM). December 1990.

92G728 ITU-T Recommendation G.728: Coding of Speech at 16 kbps using Low-Delay Code Excited Linear Prediction. September 1992.

93H261 ITU-T Recommendation H.261: Video Codec for Audiovisual Services at p x 64 kbits. March 1993.

95BT601 ITU-R Recommendation BT.601-5: Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios. October 1995.

96G723 ITU-T Recommendation G.723.1: Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbps. March 1996.

96G729 ITU-T Recommendation G.729: C source code and test vectors for implementation verification of the G.729 8 kbps CS-ACELP speech coder. March 1996.

96H263 ITU-T Recommendation H.263: Video Coding for Low Bit Rate Communication. March 1996.

98H263 ITU-T Recommendation H.263: Video Coding for Low Bit Rate Communication. January 1998.

98T38 ITU-T Recommendation T.38: Procedures for real-time Group 3 facsimile communication over IP networks. June 1998.

99T38 ITU-T Recommendation T.38 (Amendment 1): Procedures for real-time Group 3 facsimile communication over IP networks. April 1999.

99T140 ITU-T Recommendation T.140: Protocol for multimedia application text conversation. February 1998 with addendum 1 to T.140 in February 2000.

00G711A2 ITU-T Recommendation G.711, Appendix II: A comfort noise payload definition for ITU-T G.711 use in packet-based multimedia communication systems. February 2000.

G723A96 ITU-T Recommendation G.723.1, Annex A: C reference code, test signals and test sequences for the fixed point 5.3 and 6.3 kbps dual rate speech coder and for the silence compression scheme, version 5.1. November 1996.

G729A96 ITU-T Recommendation G.729, Annex A: C source code and test vectors for implementation verification of the G.729 reduced complexity 8 kbps CS-ACELP speech coder. November 1996.

G729B97 ITU-T Recommendation G.729, Annex B: C source code and test vectors for implementation verification of the algorithm of the G.729 silence compression scheme. August 1997.

Contenidos

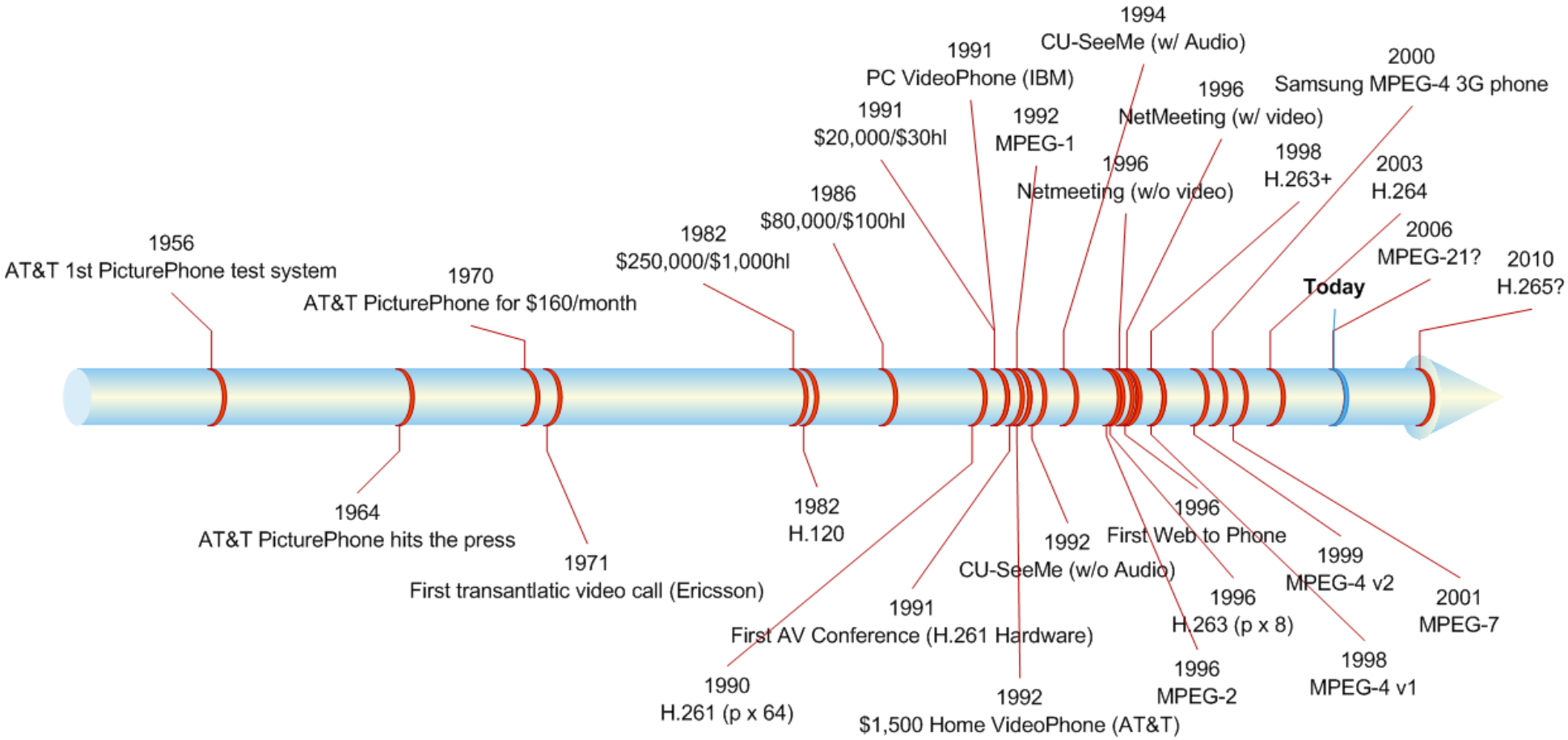
- **UNIDAD DIDÁCTICA 3: FORMATOS DE AUDIO Y VIDEO**
  - AUDIO.
    1. Voz: G.711 … G.729.
    2. Encapsulado RTP.
  - **VIDEO.**
    1. **H.261, H.263 y H.264.**
    2. **Encapsulado RTP.**
  - CONTENEDORES.
    1.- MP3.
    2.- MPEG4.
  - QoA. PSNR y PSQM, MOS.

Video Codec

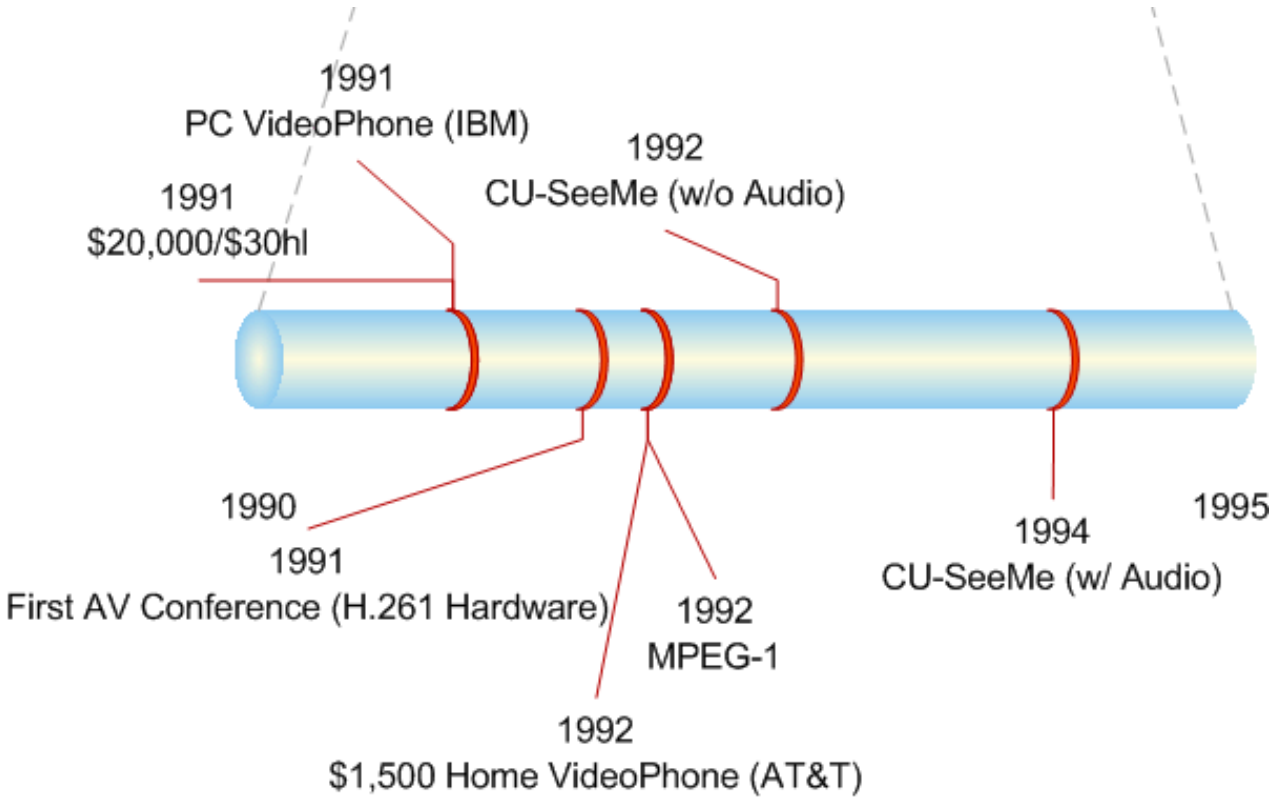
Overview

- **Codec de vídeo implica la codificación de vídeo y de audio** (no habla).
- Codificación CBR que no tiene en cuenta la variabilidad del canal inherentes a las aplicaciones multimedia.
  Gran demanda de streaming multimedia.
- **Tráfico VBR**.
- **Calidad variable**.

Video Codec Historia

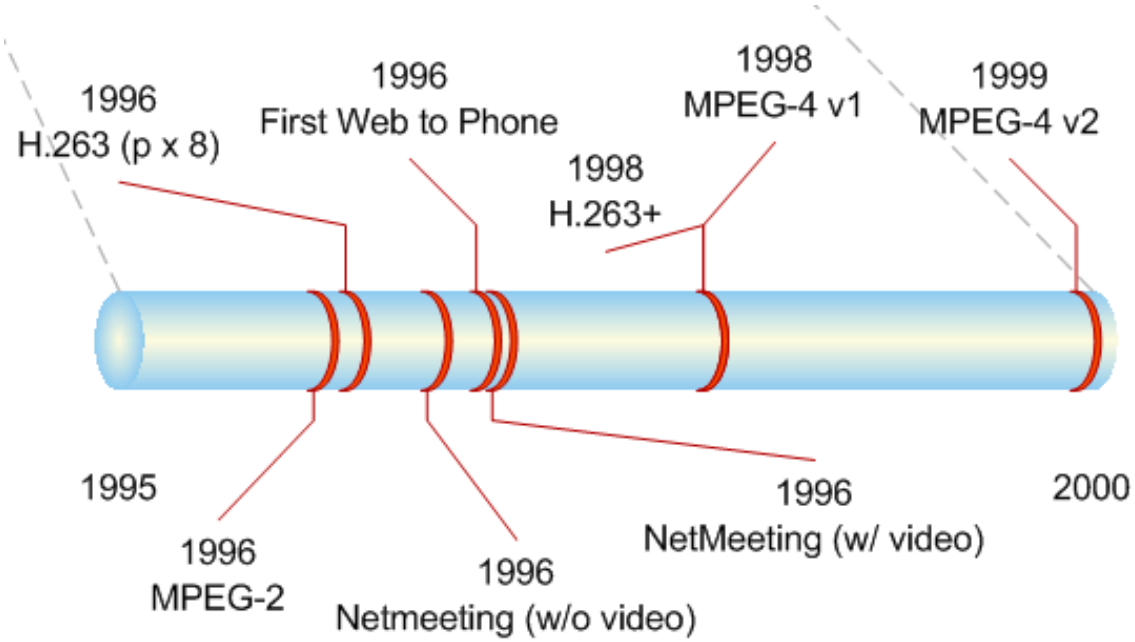


Video Codec Historia



- MPEG-1: “Codificación de cuadros de movimiento y audio asociado para el almacenamiento digital de medios” (1992)
  - Objetivo era calidad VHS a 1.5MBits/s
  - Basado en Video-CD
  - MP3 se utiliza todavía (MPEG-1 Layer 3)

Video Codec Historia

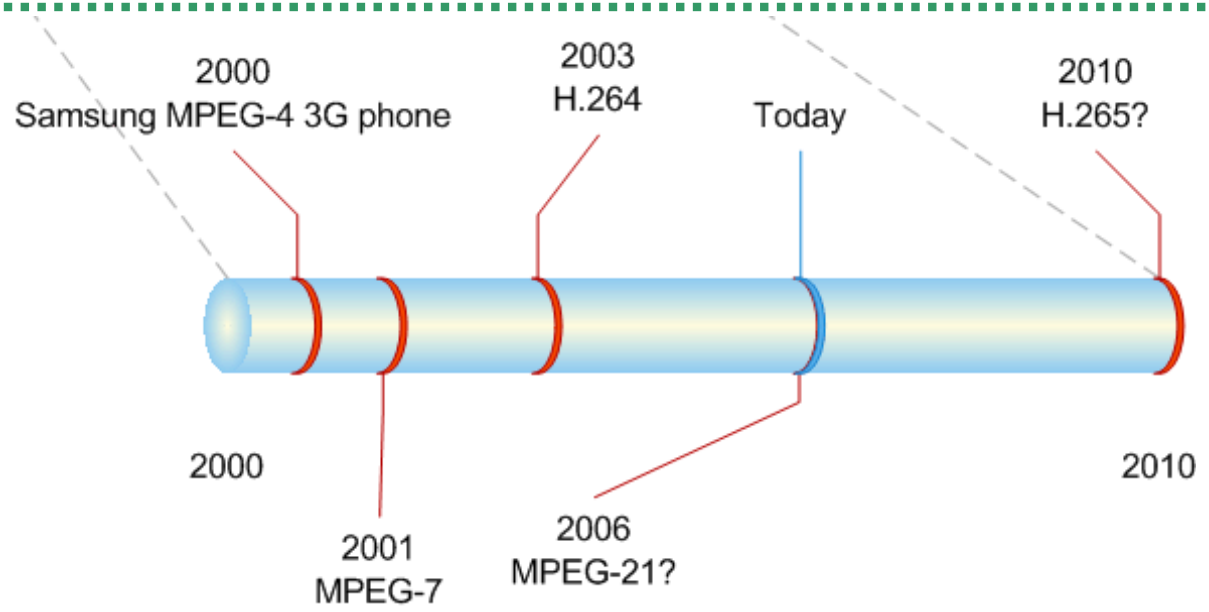


- MPEG-2: "Código genérico de codificación de imágenes en movimiento y audio asociado"
  - Broadcasting y almacenamiento
  - Bitrates: 4-9 MBits/s
  - Satellite TV, DVD
- MPEG-3?
  - Orientado a hacer TV de alta definición (HDTV)
  - MPEG-2 es también compatible con este formato
  - Se integra en MPEG-2

MPEG-4: "Codificación visual de objetos"

- Comenzó como proyecto de muy bajo almacenamiento
- Es mucho más ajustado en:
  - Codificación de *objectos media*
  - 64kbps hasta 240Mbps (Part 10/H.264)
  - Permite incluir objetos Sintéticos/Semi-sintéticos
  - XMT: Como HTML, para para vídeo
  - Primer estándard con Propiedad Intelectual

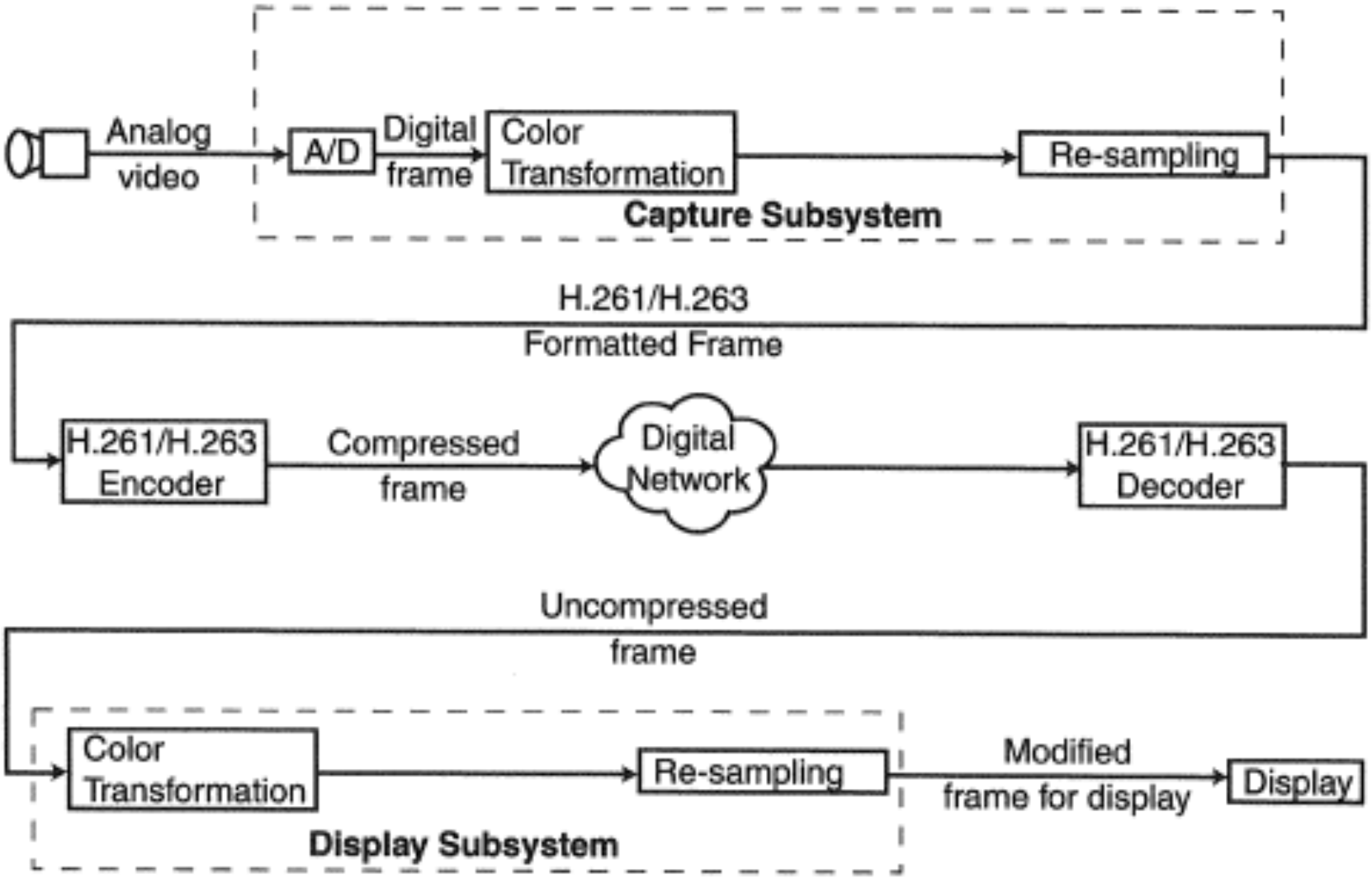
Video Codec Historia



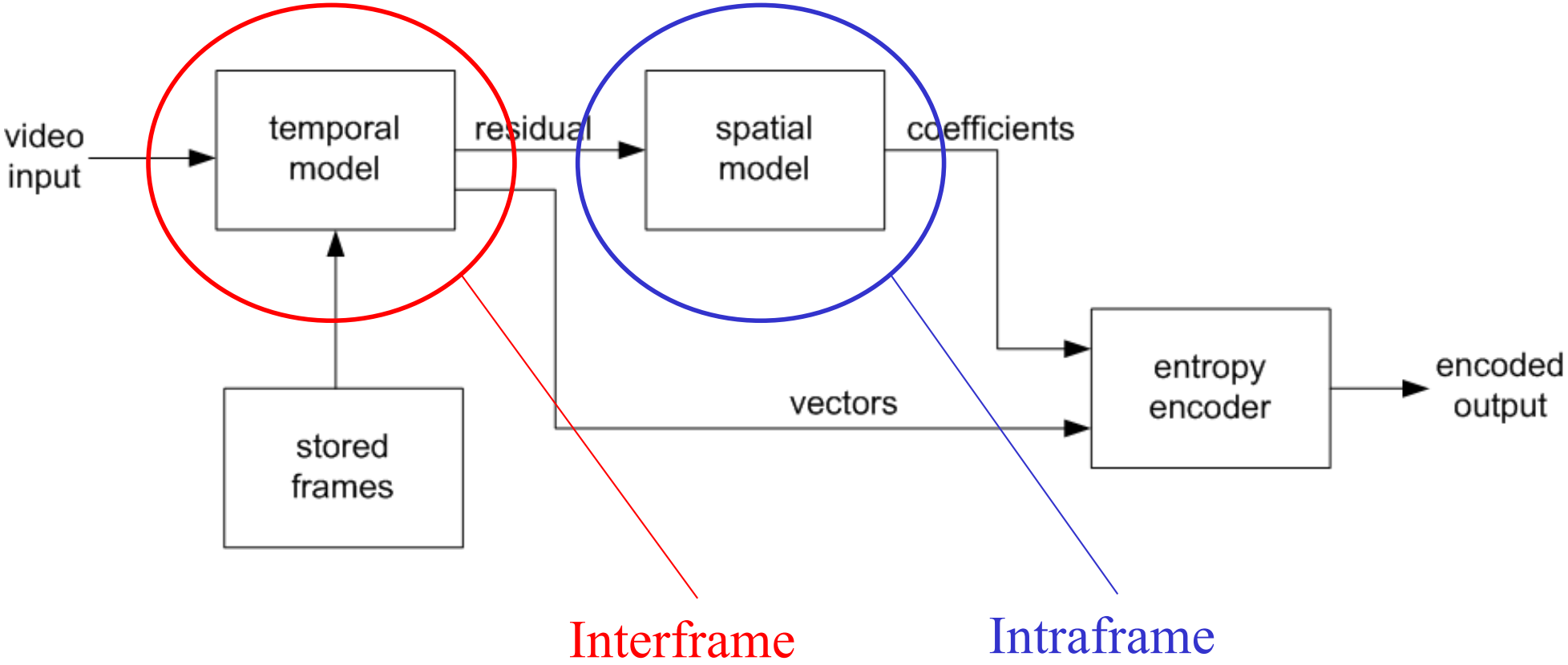
- MPEG-4 Part 10: Advanced Video Coding similar a H.264
  - Diseñado junto a MPEG y ITU-T group
  - Indica un ahorro 50% bitrate repecto a MPEG-2, 30% over MPEG-4!
- H.265 tiene mejor resultado frente al 50% de compresión

- MPEG-7: "Multimedia Content Description Interface" (2001)
  - Descripción audio/video
- Applicaciones
  - Indexación de video databases
  - Búsquedas & Recuperaciones
  - Navegación

Video Codec


Analog
video
A/D
Digital
frame
Color
Transformation
Re-sampling
Capture Subsystem
H.261/H.263
Formatted Frame
H.261/H.263
Encoder
Compressed
frame
Digital
Network
H.261/H.263
Decoder
Uncompressed
frame
Color
Transformation
Re-sampling
Modified
frame for display
Display
Display Subsystem

Video Codec

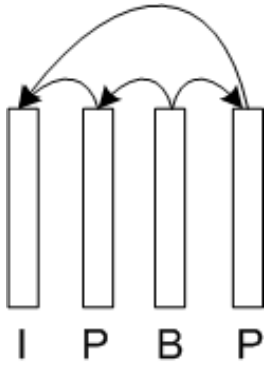

video input
temporal model
residual
spatial model
coefficients
entropy encoder
encoded output
vectors
stored frames
Interframe
Intraframe

Motion Compensation

- Su objetivo es reducir los datos transmitidos por la detección del movimiento de los objetos
- Utiliza la imagen anterior como referencia (Imágenes P).
- Pasos:
    - Dividir la imagen (marco) actual en bloques. Para cada uno de ellos:
        - Encuentrar el mejor bloque respecto al marco de referencia ¿Cómo? Buscamos la zona en el marco de referencia y comparar, con
            - Suma de diferencias absolutas (SAD)
            - La media de SAD (MSAD)
    - El mejor juego de bloque (vector) se codifica y se transmite
    - El resultado se puede utilizar una referencia pero no demasiado

Video Codec

- Imágenes (frames) Bidireccionales
  - Al realizar la estimación de movimiento, cada bloque se le puede asignar un juego que va en función de las imágenes anterior y posterior.



- Global Motion Compensation
  - Un conjunto de vectores que describen el plano del objeto todo se puede transmitir

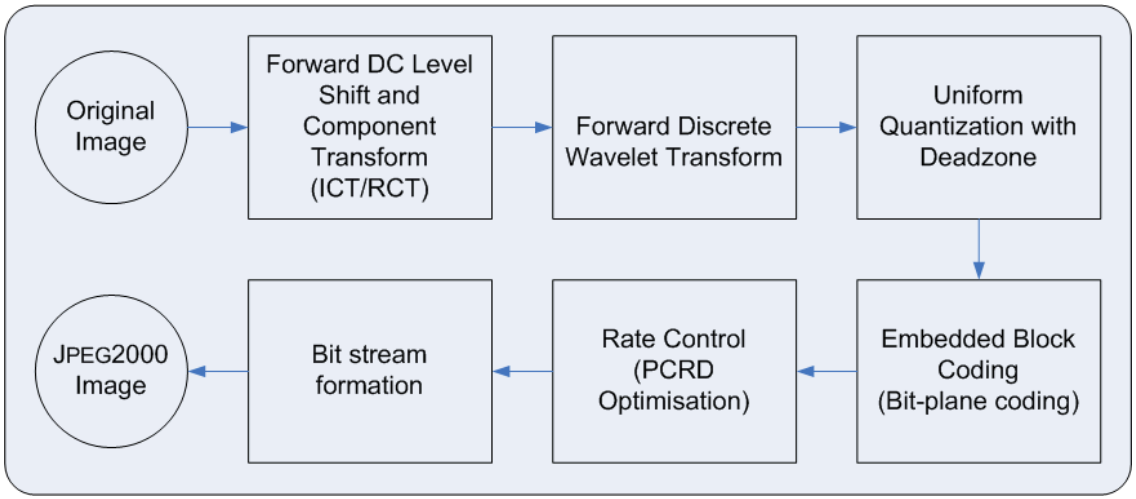
Video Codec

- Otras herramientas
  - Movimiento predictivo de compensación a nivel pixel (no bloques)
    - Grandes vectores, pero una mejor predicción
  - Predicción espacial (intra-nivel)
    - Predicción de píxeles espacialmente
  - Vectores de movimiento fuera de la imagen
  - Bloques 4x4
    - Más rápido, reversible, explota la correlación espacial mejor
  - Filtro de desbloqueo (Imagen congelada)
    - Ejecutado en el codificador, reduce el bloqueo
  - Varios marcos de referencia

JPEG2000

- Resolución y calidad escalable
  - Codifica una vez, pero la decodificación se realiza a resolución y calidad apropiada en el cliente, e.g.
    - Telefonía móvil: baja resolución /calidad
    - PC Altas prestaciones: Alta resolución/calidad
- Acceso aleatorio
  - Cualquier parte del flujo puede ser decodificada independientemente
- Codificación basada en pérdidas
- Rápida, altamente eficiente en términos de compresión
- Datos son organizados en capas con diferentes tareas

Video Codec



- Transforma compresiones de RGB a YUV
- Utiliza la transformada Wavelet dando una presentación multi-resolución de la imagen
- Cuantificación es opcional

- Bytes son organizados en planos de bits.
- Post-compresión se realiza en una optimación R-D "truncando" los planos para alcanzar una nivel óptimo de distorsión para la tasa especficada.

Bit 7 (MSB) Bit-plane 0	Bit 6 Bit-plane 1	Bit 5 Bit-plane 2	Bit 4 Bit-plane 3	Bit 3 Bit-plane 4	Bit 2 Bit-plane 5	Bit 1 Bit-plane 6	Bit 0 Bit-plane 7
1	0	1	1	1	0	1	0
1	1	0	1	1	1	0	0
1	1	0	0	0	1	0	0
0	1	0	0	0	0	1	0
1	1	1	1	1	0	1	1
0	1	1	0	1	0	1	1

DCT

$$D(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) \cos\left[\frac{(2x+1)i\pi}{2N} \right] \cos\left[\frac{(2y+1)j\pi}{2N} \right]$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{if } u > 0 \end{cases}$$

Para bloques 8x8

$$D(i,j) = \frac{1}{4} C(i)C(j) \sum_{x=0}^{7} \sum_{y=0}^{7} p(x,y) \cos\left[\frac{(2x+1)i\pi}{16} \right] \cos\left[\frac{(2y+1)j\pi}{16} \right]$$

DCT Matrix

$$T_{i,j} = \begin{cases} \frac{1}{\sqrt{N}} & \text{if } i = 0 \\ \sqrt{\frac{2}{N}} \cos\left[\frac{(2j+1)i\pi}{2N}\right] & \text{if } i > 0 \end{cases}$$

$$T = \begin{bmatrix} .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 \\ .4904 & .4157 & .2778 & .0975 & -.0975 & -.2778 & -.4157 & -.4904 \\ .4619 & .1913 & -.1913 & -.4619 & -.4619 & -.1913 & .1913 & .4619 \\ .4157 & -.0975 & -.4904 & -.2778 & .2778 & .4904 & .0975 & -.4157 \\ .3536 & -.3536 & -.3536 & .3536 & .3536 & -.3536 & -.3536 & .3536 \\ .2778 & -.4904 & .0975 & .4157 & -.4157 & -.0975 & .4904 & -.2778 \\ .1913 & -.4619 & .4619 & -.1913 & -.1913 & .4619 & -.4619 & .1913 \\ .0975 & -.2778 & .4157 & -.4904 & .4904 & -.4157 & .2778 & -.0975 \end{bmatrix}$$

DCT Matrix

$$Original = \begin{bmatrix} 154 & 123 & 123 & 123 & 123 & 123 & 123 & 136 \\ 192 & 180 & 136 & 154 & 154 & 154 & 136 & 110 \\ 254 & 198 & 154 & 154 & 180 & 154 & 123 & 123 \\ 239 & 180 & 136 & 180 & 180 & 166 & 123 & 123 \\ 180 & 154 & 136 & 167 & 166 & 149 & 136 & 136 \\ 128 & 136 & 123 & 136 & 154 & 180 & 198 & 154 \\ 123 & 105 & 110 & 149 & 136 & 136 & 180 & 166 \\ 110 & 136 & 123 & 123 & 123 & 136 & 154 & 136 \end{bmatrix}$$

$$M = \begin{bmatrix} 26 & -5 & -5 & -5 & -5 & -5 & -5 & 8 \\ 64 & 52 & 8 & 26 & 26 & 26 & 8 & -18 \\ 126 & 70 & 26 & 26 & 52 & 26 & -5 & -5 \\ 111 & 52 & 8 & 52 & 52 & 38 & -5 & -5 \\ 52 & 26 & 8 & 39 & 38 & 21 & 8 & 8 \\ 0 & 8 & -5 & 8 & 26 & 52 & 70 & 26 \\ -5 & -23 & -18 & 21 & 8 & 8 & 52 & 38 \\ -18 & 8 & -5 & -5 & -5 & 8 & 26 & 8 \end{bmatrix}$$

$$D = TMT'$$

DCT Matrix

$$D = \begin{bmatrix} 162.3 & 40.6 & 20.0 & 72.3 & 30.3 & 12.5 & -19.7 & -11.5 \\ 30.5 & 108.4 & 10.5 & 32.3 & 27.7 & -15.5 & 18.4 & -2.0 \\ -94.1 & -60.1 & 12.3 & -43.4 & -31.3 & 6.1 & -3.3 & 7.1 \\ -38.6 & -83.4 & -5.4 & -22.2 & -13.5 & 15.5 & -1.3 & 3.5 \\ -31.3 & 17.9 & -5.5 & -12.4 & 14.3 & -6.0 & 11.5 & -6.0 \\ -0.9 & -11.8 & 12.8 & 0.2 & 28.1 & 12.6 & 8.4 & 2.9 \\ 4.6 & -2.4 & 12.2 & 6.6 & -18.7 & -12.8 & 7.7 & 12.0 \\ -10.0 & 11.2 & 7.8 & -16.3 & 21.5 & 0.0 & 5.9 & 10.7 \end{bmatrix}$$

Quantization

$$Q_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Quantization

$$C_{i,j} = round\left(\frac{D_{i,j}}{Q_{i,j}}\right)$$

$$C = \begin{bmatrix} 10 & 4 & 2 & 5 & 1 & 0 & 0 & 0 \\ 3 & 9 & 1 & 2 & 1 & 0 & 0 & 0 \\ -7 & -5 & 1 & -2 & -1 & 0 & 0 & 0 \\ -3 & -5 & 0 & -1 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Video Codec

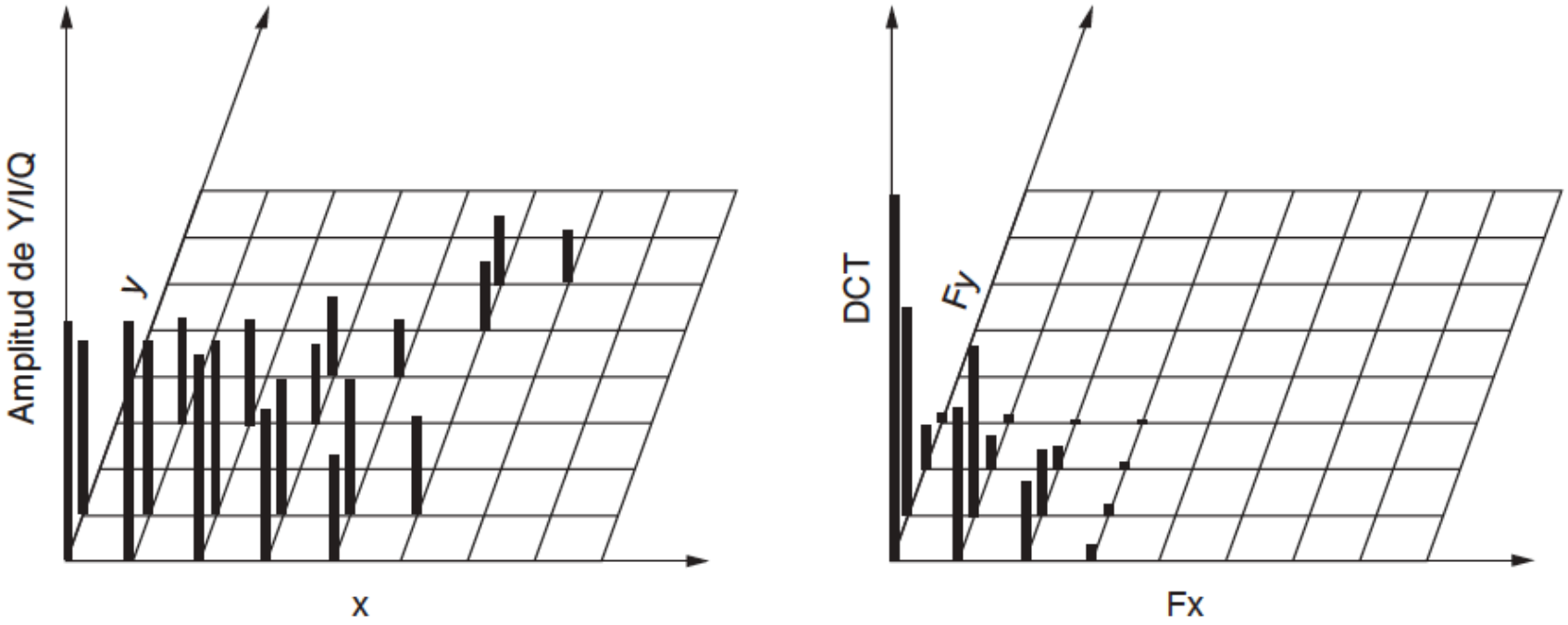
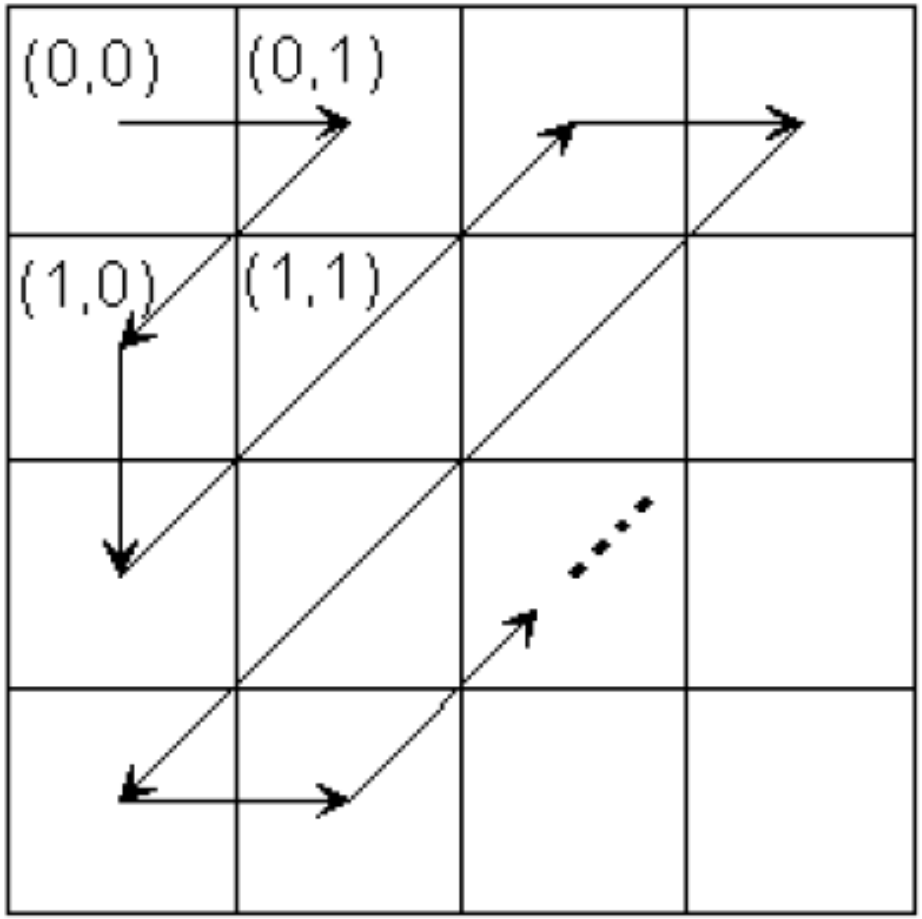


Figura 7-7. (a) Un bloque de la matriz *Y*. (b) Los coeficientes DCT.

Video Codec

Codec



Descompression

$$R_{i,j} = Q_{i,j} \times C_{i,j}$$

$$R = \begin{bmatrix} 160 & 44 & 20 & 80 & 24 & 0 & 0 & 0 \\ 36 & 108 & 14 & 38 & 26 & 0 & 0 & 0 \\ -98 & -65 & 16 & -48 & -40 & 0 & 0 & 0 \\ -42 & -85 & 0 & -29 & 0 & 0 & 0 & 0 \\ -36 & 22 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

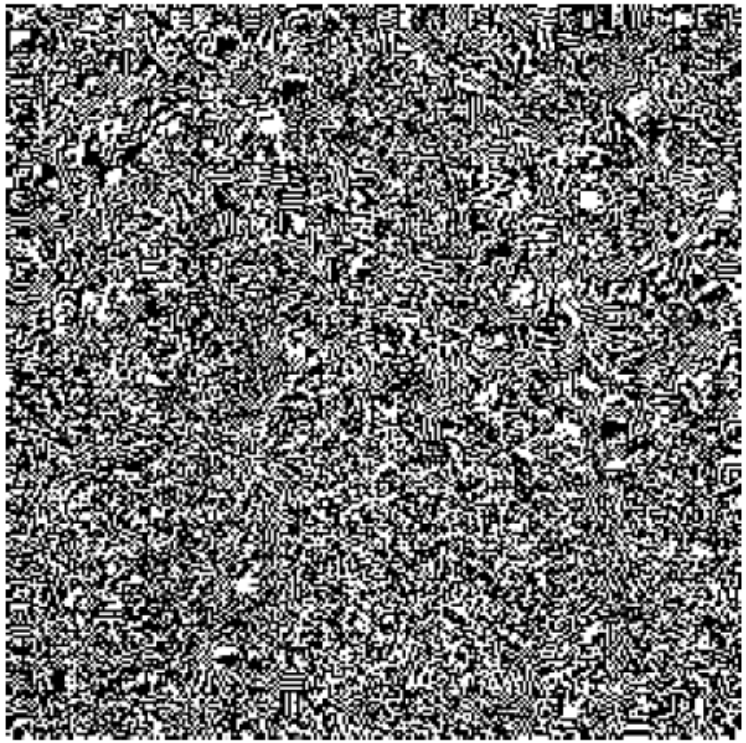
Descompression

$$N = round(T' \; R \; T) + 128$$

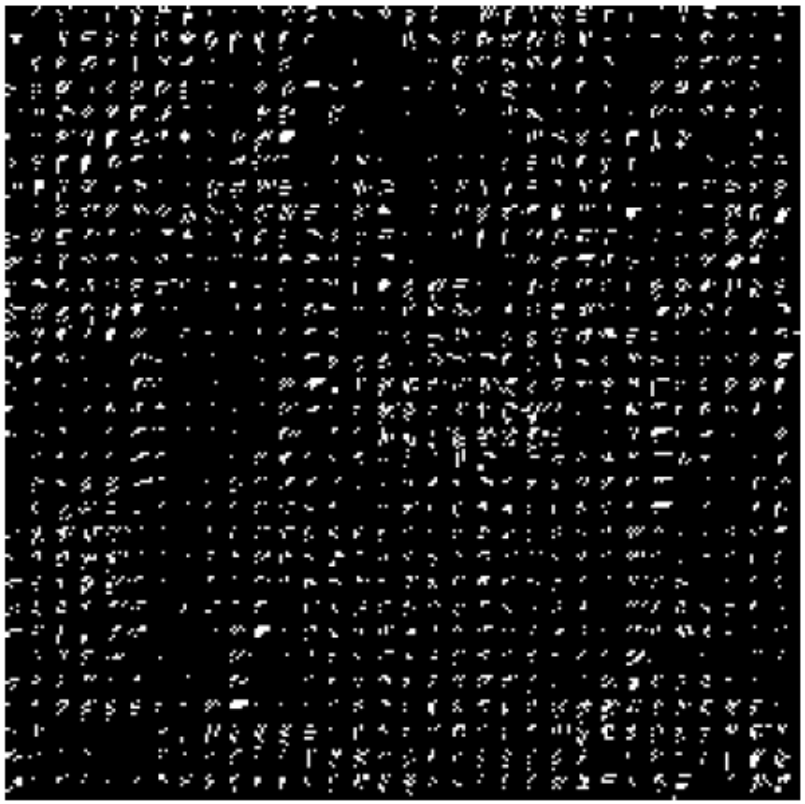
$$Original = \begin{bmatrix} 154 & 123 & 123 & 123 & 123 & 123 & 123 & 136 \\ 192 & 180 & 136 & 154 & 154 & 154 & 136 & 110 \\ 254 & 198 & 154 & 154 & 180 & 154 & 123 & 123 \\ 239 & 180 & 136 & 180 & 180 & 166 & 123 & 123 \\ 180 & 154 & 136 & 167 & 166 & 149 & 136 & 136 \\ 128 & 136 & 123 & 136 & 154 & 180 & 198 & 154 \\ 123 & 105 & 110 & 149 & 136 & 136 & 180 & 166 \\ 110 & 136 & 123 & 123 & 123 & 136 & 154 & 136 \end{bmatrix}$$

$$Decompressed = \begin{bmatrix} 149 & 134 & 119 & 116 & 121 & 126 & 127 & 128 \\ 204 & 168 & 140 & 144 & 155 & 150 & 135 & 125 \\ 253 & 195 & 155 & 166 & 183 & 165 & 131 & 111 \\ 245 & 185 & 148 & 166 & 184 & 160 & 124 & 107 \\ 188 & 149 & 132 & 155 & 172 & 159 & 141 & 136 \\ 132 & 123 & 125 & 143 & 160 & 166 & 168 & 171 \\ 109 & 119 & 126 & 128 & 139 & 158 & 168 & 166 \\ 111 & 127 & 127 & 114 & 118 & 141 & 147 & 135 \end{bmatrix}$$

Video Codec



Video Codec



H.261

Motivación

Errores del vídeo/audio sin comprimir son enormes

La Relación de compresión de métodos sin pérdida no son lo suficientemente alta

Las redes de destino son p * 64 Kbps, 1 < p < 30

64 kbps (p = 1) <velocidad de datos <1920Kbps (p = 30)

Cubre la transmisión de la tasa básica RDSI (64 Kbps) hasta más allá de la T-1 la velocidad de datos (1,54 Mbps)

Retardo máximo de 150 ms

Algoritmo de codificación es un híbrido de:

Inter-imagen de predicción - elimina la redundancia temporal
Una codificación por transformación - elimina la redundancia espacial
La compensación de movimiento - utiliza vectores de movimiento para ayudar a los codec compensar el movimiento.

La velocidad de datos se puede establecer entre 40 Kbit / s y 2 Mbit / s

Formato de la señal de entrada

CIF (Formato Intermedio Común) y QCIF (Quarter CIF).

H.261

Bit rate

The objetivo es ~ 64Kbps to 1920Kbps

Formatos de imagen

CIF (Common Intermediate Format) – NTSC & PAL

QCIF (Quarter Common Intermediate Format)

Picture Formats Supported

Picture format	Luminance pixels	Luminance lines	H.261 support	Uncompressed bitrate (Mbit/s)			
				10 frames/s		30 frames/s	
				Grey	Colour	Grey	Colour
QCIF	176	144	Yes	2.0	3.0	6.1	9.1
CIF	352	288	Optional	8.1	12.2	24.3	36.5

A tasa 29,97 imágenes por segundo con 4:2:0 submuestreo de crominancia ($Y:C_B:C_R$)

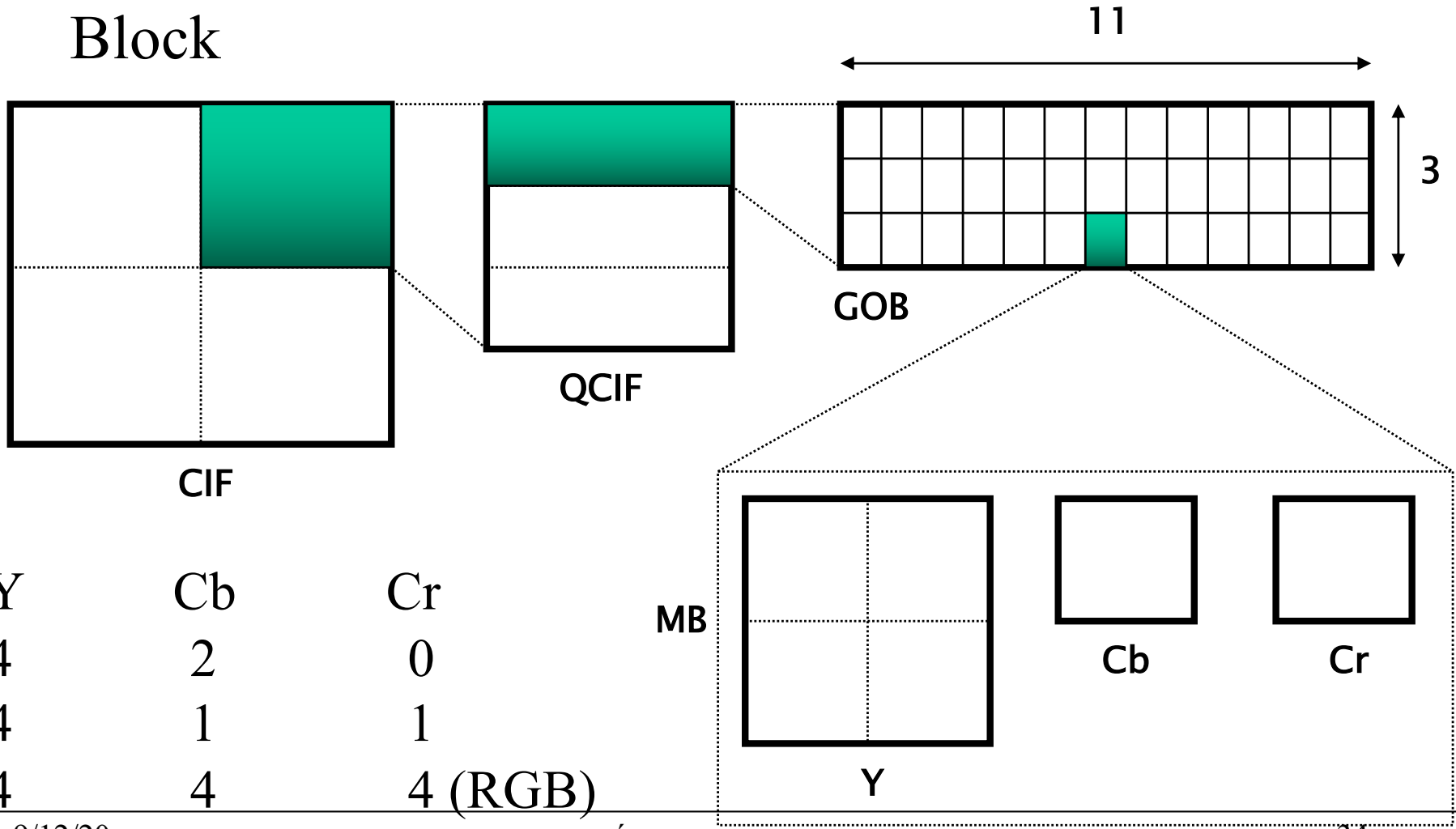
H.261

Group of blocks structure

- Picture – codificado como luminancia y dos componentes de diferencia de color (Y, C_B and C_R)
- Group of blocks (GOB)
- MacroBlock (MB)
- Block

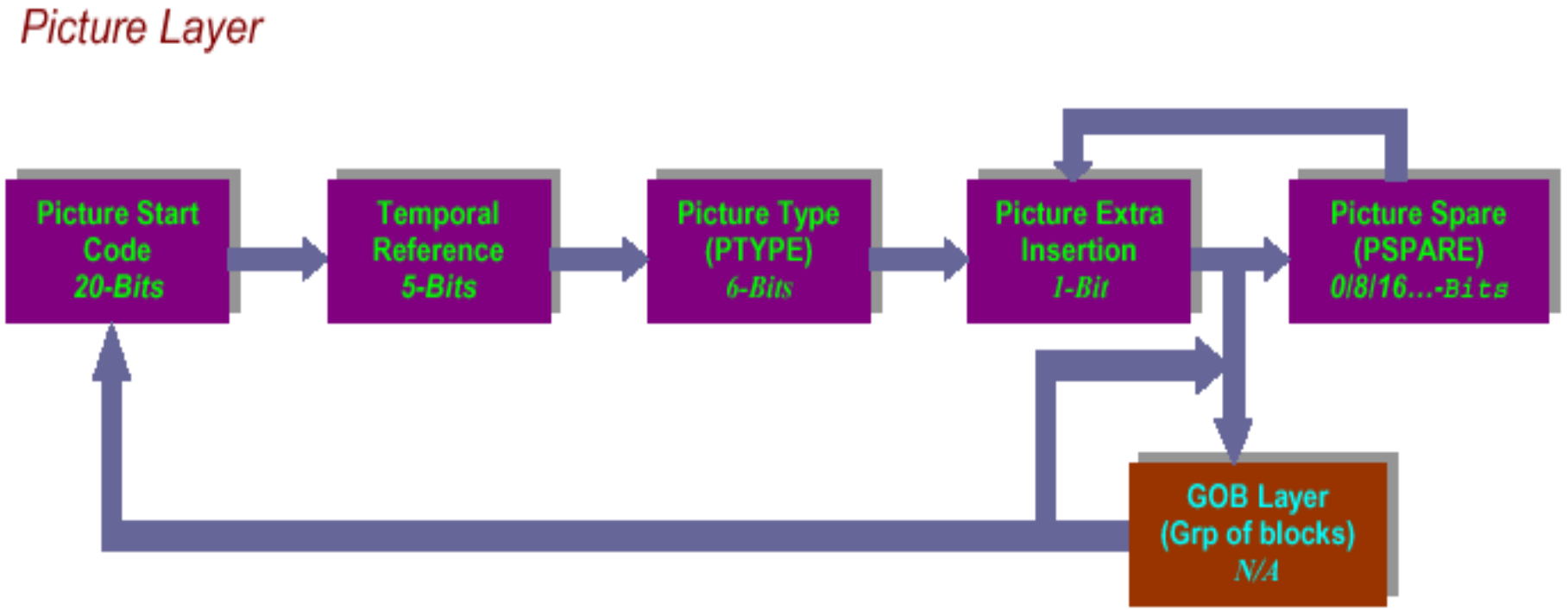
H.261- blocks

- 4 capas en el flujo comprimido
  - Picture, GOB (Group Of Block), MB (MacroBlock), Block

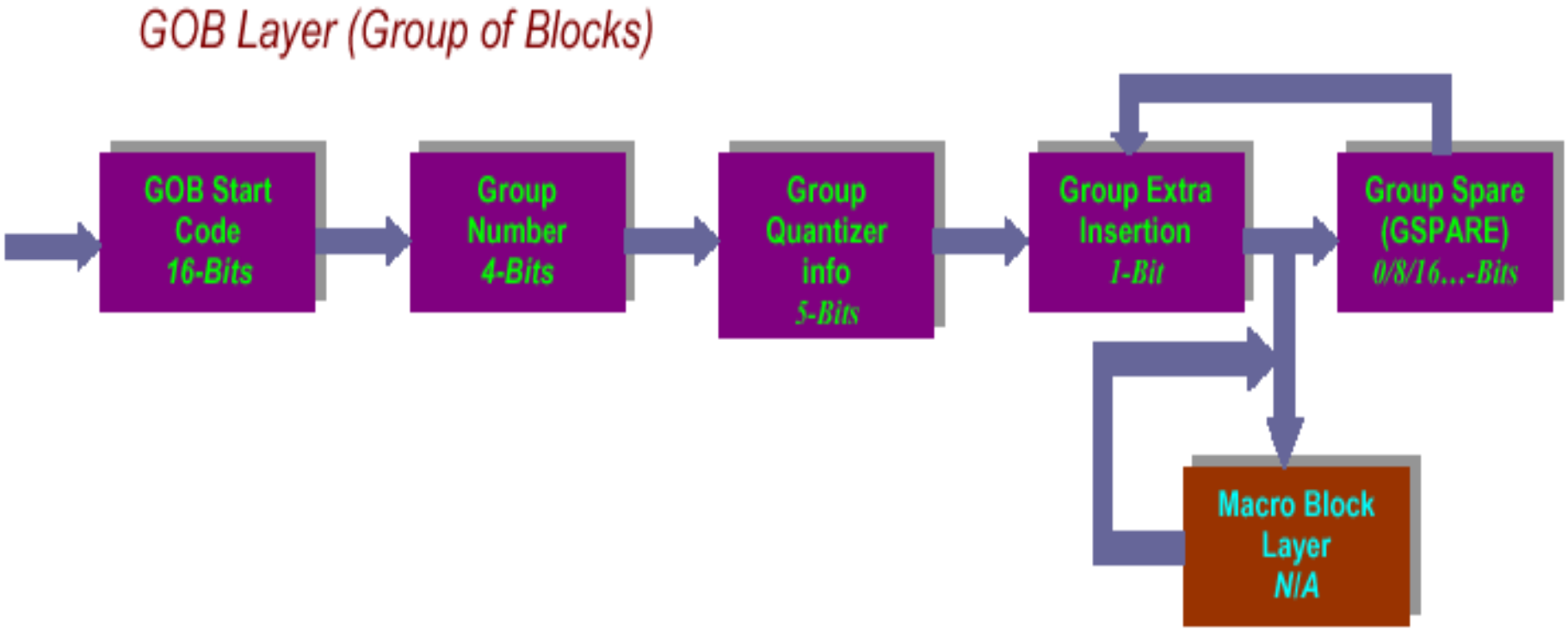


Y	Cb	Cr
4	2	0
4	1	1
4	4	4 (RGB)

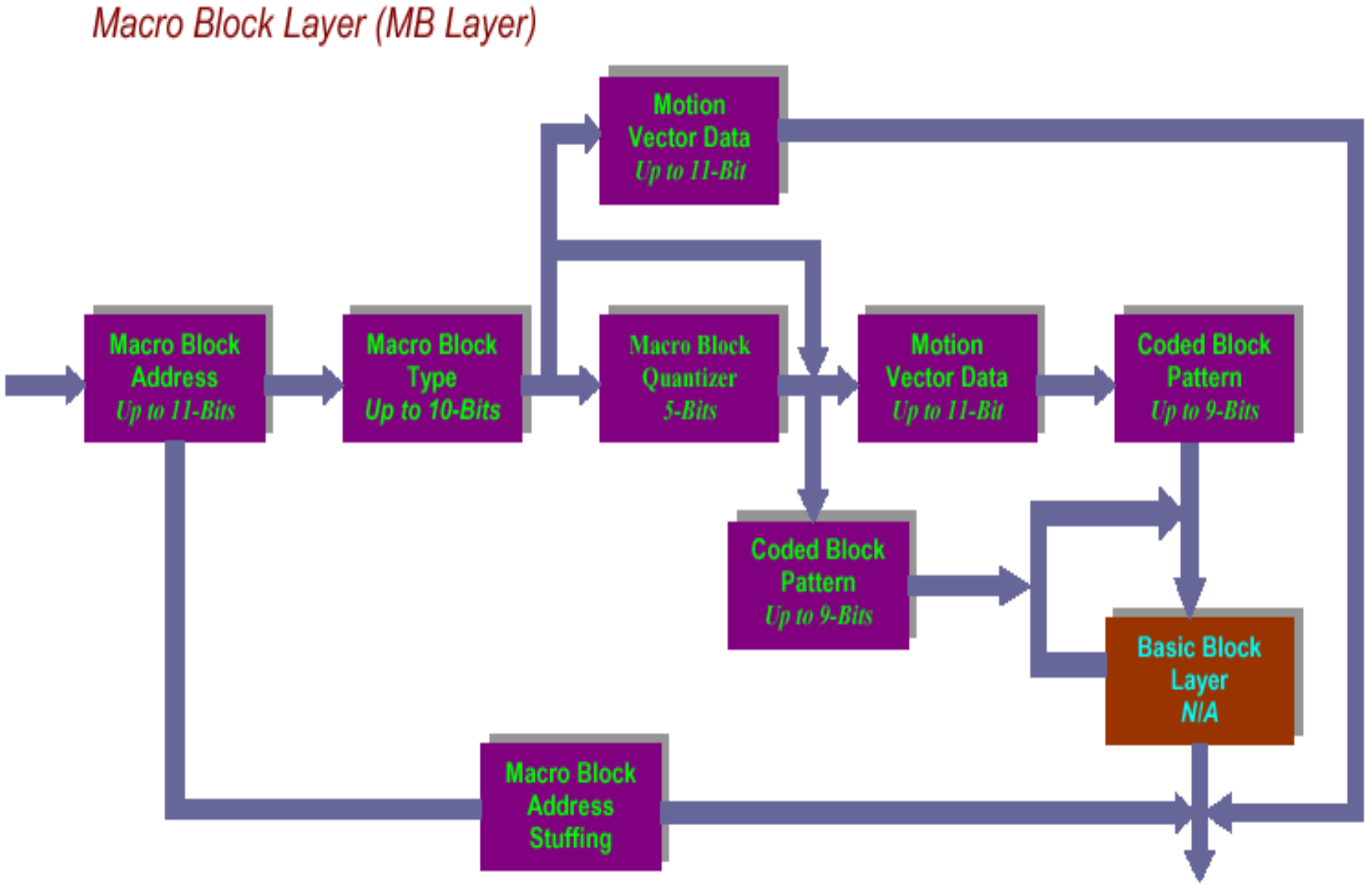
H.261- Formatos



H.261- Formatos

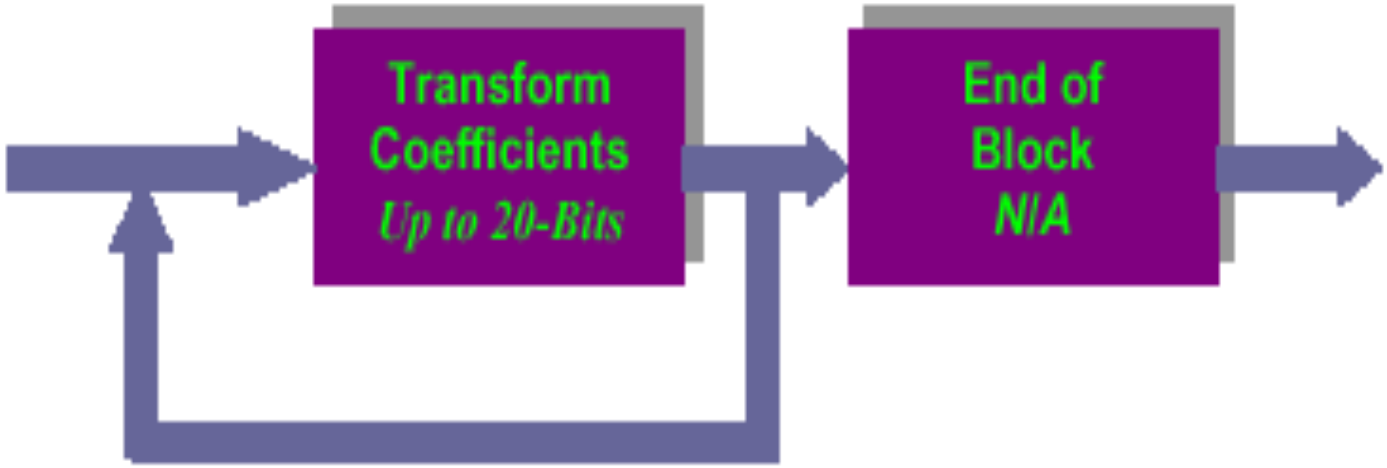


H.261- Formatos



H.261- Format

Basic Block Layer



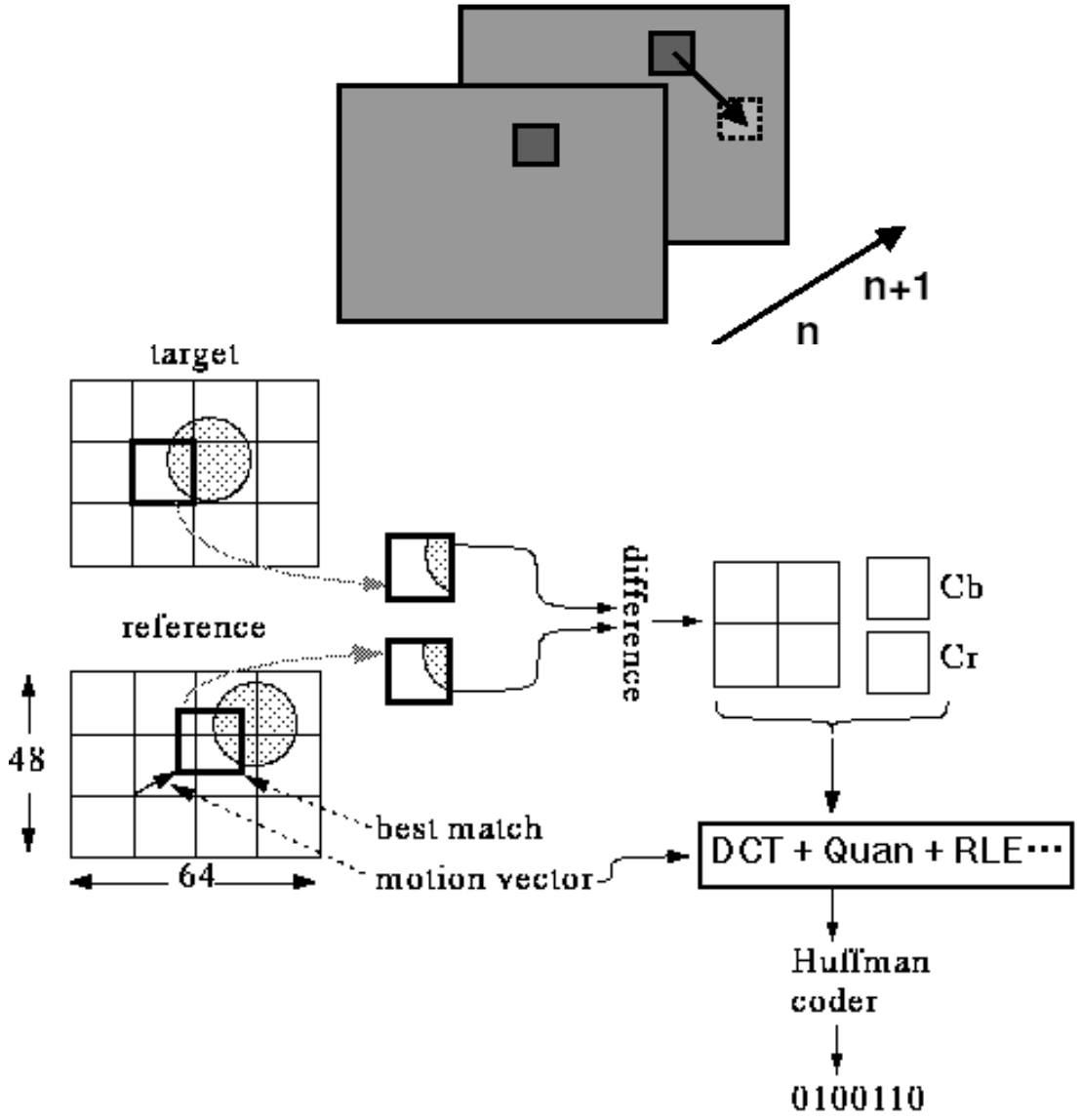
H.261- Formatos

- Intra-Encoded Frames (I-Frames)
  - Similares a la compresión JPEG
  - Filtrado espacial – cada objeto es un objeto (transformada de código coding)
    - Ej.: patrón de blanco y negro del fondo de la imagen
- Predicted Frames (P-Frame)
  - Se predicen en función de la imagen (frame) anterior
  - Utilizan filtrado temporal – Predicción entre imágenes
    - Ej.: Individuo en una imagen

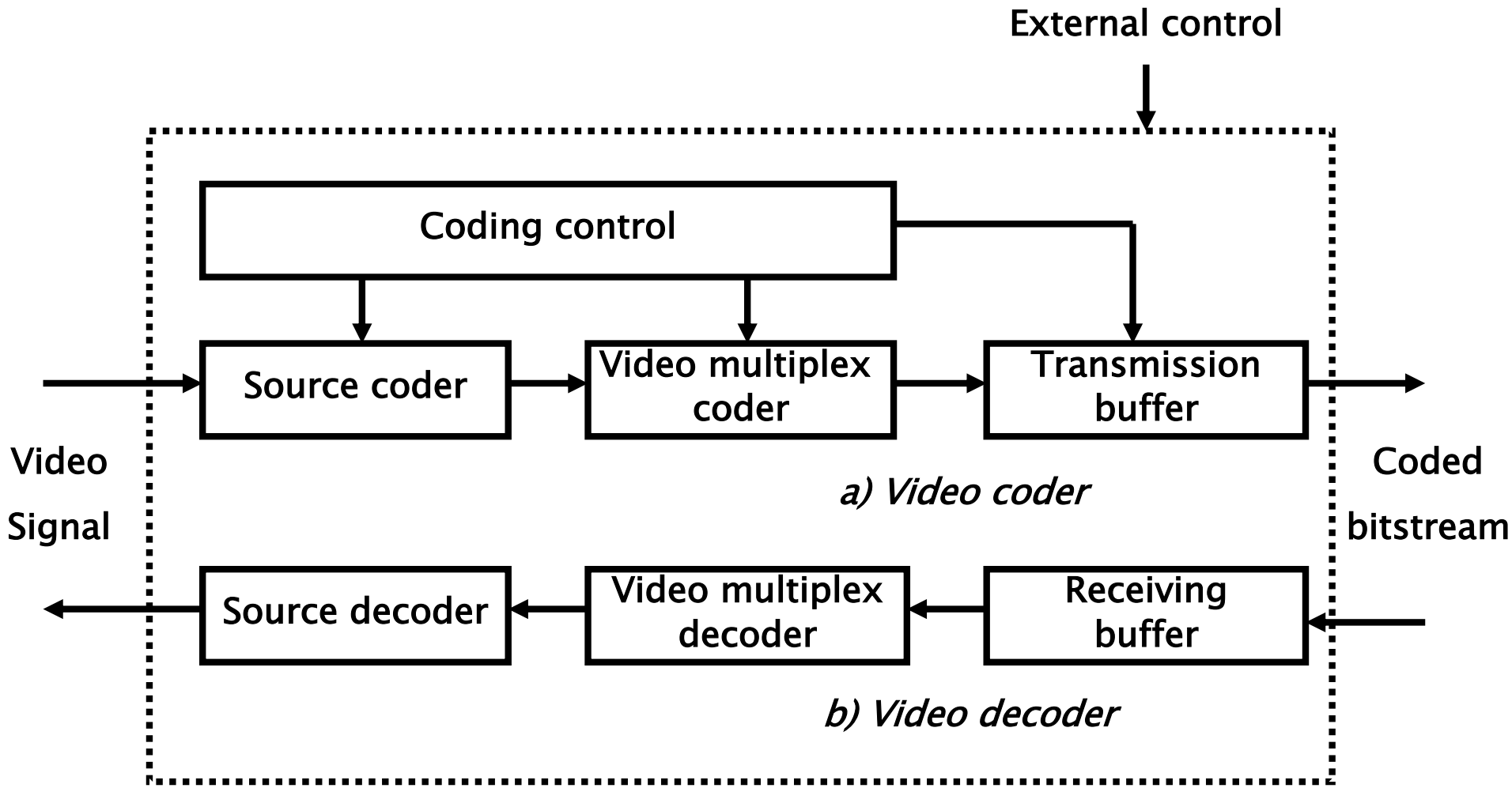
H.261- Format

- Asunciones
  - Cada parte de la imagen se mueve , su color está casi constante

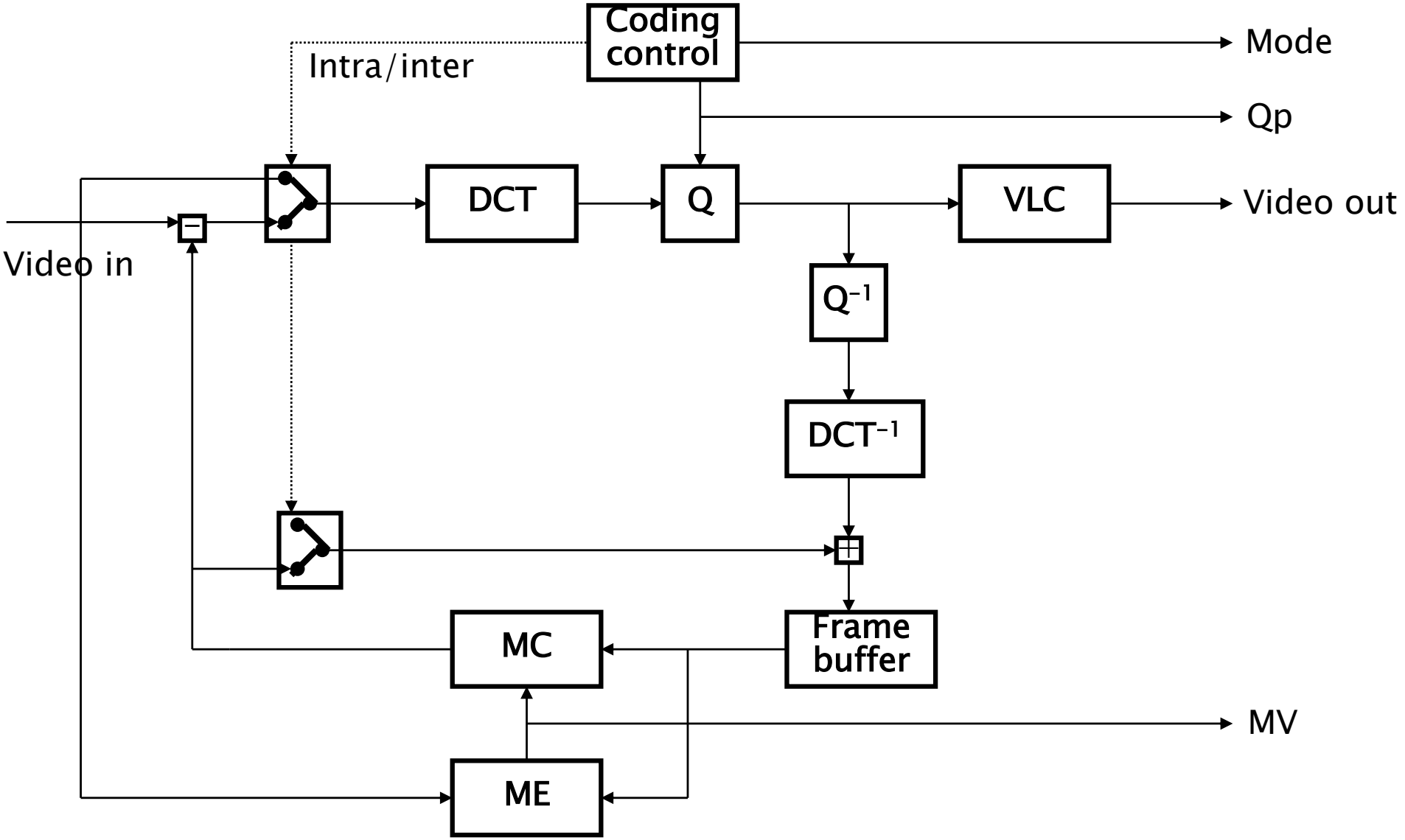
- Idea
  - Encontrar partes similares en otras imágenes
  - Codificar donde se ha encontrado (i.e. vector de movimiento)
  - Código imagen previa – Imagen de referncia
  - Imagen a código – Imagen objetivo
  - Codifica sólo el residual



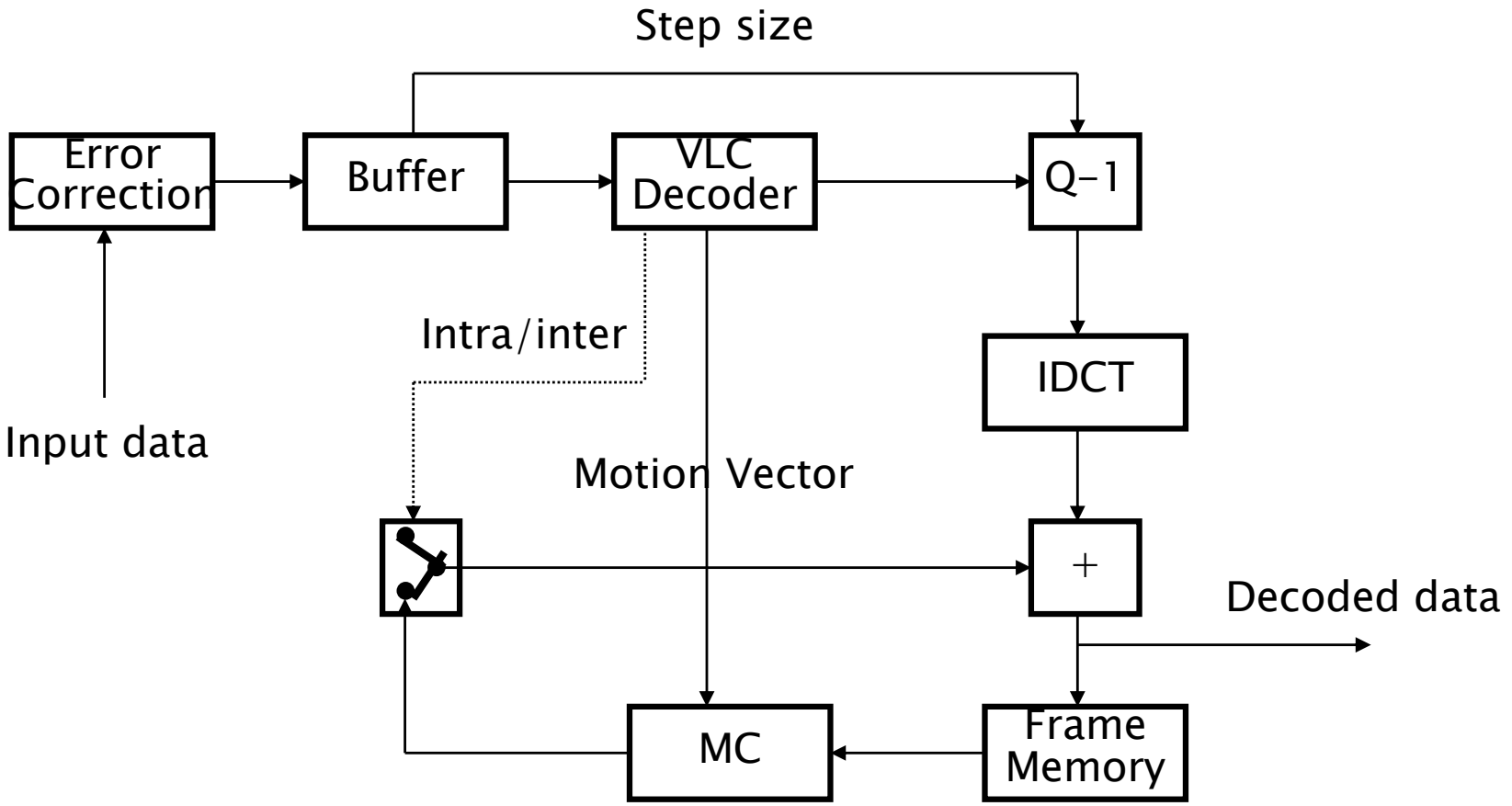
H.261- Coders


External control
Coding control
Source coder
Video multiplex coder
Transmission buffer
Video Signal
Coded bitstream
a) Video coder
Source decoder
Video multiplex decoder
Receiving buffer
b) Video decoder

H.261- Encoder



H.261- Decoder



H.261

The color representation of NTSC and PAL is defined in [95BT601], and we refer to it as YC_rC_b with a 4:2:0 pixel subsampling ratio. A simple way to describe this representation is to first assume that the digital video is initially represented in a *Red, Green, and Blue* (RGB) format where each picture or frame consists of red, green, and blue color planes. Similarly, each pixel is represented by a red, green, and blue pixel. In the case of digitized NTSC, each video picture has three color planes, each of which have a resolution of 720-by-480. The Y (or luminance) plane of the YC_rC_b format represents the intensity value of each color pixel in the picture. The C_b plane, referred to as a color difference or chrominance plane, represents the values of the blue plane subtracted from Y, and the C_r plane similarly represents the values of the red plane subtracted from Y. The color transformation is implemented as a 3-by-3 matrix transform of RGB pixel values to YC_rC_b pixel values. Recommendations H.261 and H.263 also require the color difference planes to be subsampled by two, relative to the luminance plane, both in vertical and horizontal directions. This process constitutes a first step toward data compression by exploiting the fact that the human visual system has lower spatial resolution sensitivity to color than to luminance. Assuming that the luminance plane is still 720-by-480, the chrominance planes are now each 360-by-240.

H.261

The color representation of NTSC and PAL is defined in [95BT601], and we refer to it as YC_rC_b with a 4:2:0 pixel subsampling ratio. A simple way to describe this representation is to first assume that the digital video is initially represented in a *Red, Green, and Blue* (RGB) format where each picture or frame consists of red, green, and blue color planes. Similarly, each pixel is represented by a red, green, and blue pixel. In the case of digitized NTSC, each video picture has three color planes, each of which have a resolution of 720-by-480. The Y (or luminance) plane of the YC_rC_b format represents the intensity value of each color pixel in the picture. The C_b plane, referred to as a color difference or chrominance plane, represents the values of the blue plane subtracted from Y, and the C_r plane similarly represents the values of the red plane subtracted from Y. The color transformation is implemented as a 3-by-3 matrix transform of RGB pixel values to YC_rC_b pixel values. Recommendations H.261 and H.263 also require the color difference planes to be subsampled by two, relative to the luminance plane, both in vertical and horizontal directions. This process constitutes a first step toward data compression by exploiting the fact that the human visual system has lower spatial resolution sensitivity to color than to luminance. Assuming that the luminance plane is still 720-by-480, the chrominance planes are now each 360-by-240.

H.261

The output from the color transformation subsystem is fed into the resampling subsystem that resizes the frames to one of the formats that are acceptable to H.261 or H.263. These formats are nominally specified as some multiple or fraction of *Common Intermediate Format* (CIF). CIF specifies the frame resolution ([width-by-height], [horizontal-by-vertical], or [pixels/line-by-the number of lines]) to be 352-by-288 for luminance and 176-by-144 for chrominance. Moreover, it specifies that these pixels occupy a display window where the ratio of horizontal to vertical dimensions is 4:3. Because 352 horizontal pixels are not related to 288 vertical pixels by a 4:3 ratio, a display system must use nonsquare pixel representations. Recommendation H.261 accepts frame resolutions of CIF and *Quarter Common Intermediate Format* (QCIF), which is 176-by-144. Recommendation H.263 Version 1 accepts frame resolutions of SQCIF (128-by-96), QCIF (176-by-144), CIF (352-by-288), 4CIF (704-by-576), and 16CIF (1408-by-1152). The frame resolution expressed in pixels for the C_B and C_R components consists of Sub-QCIF (64-by-48), QCIF (88-by-72), CIF (176-by-144), 4CIF (352-by-288), and 16CIF (704-by-576), respectively. Recommendation H.263 Version 2 relaxes the constraints on input dimensions to any multiple of four in both horizontal and vertical dimensions from a minimum of 4-by-4 to a maximum of 2048-by-1152.

H.261

The output from the capture subsystems is so-called uncompressed video, which still requires a high bandwidth network for transmission. As an example, video with a frame rate of 10 frames/s, QCIF resolution, and 12 bits/pixel requires a bandwidth of about 4.5 Mbps. Today's access networks do not provide such a high bandwidth as required by uncompressed video. The *Public-Switched Telephone Network* (PSTN) modem provides an access bandwidth of only 28.8 kbps to 56 kbps. Even the 128 kbps to 384 kbps access bandwidth of the *Integrated Services Digital Network* (ISDN) is not sufficient. This situation is one reason why the encoder (H.261 or H.263) compresses video before transmission on the network.

H.261

Intra-frame (also known as I-frame) coding takes place on a frame without referring to any other frames in the video sequence. An I-frame is required at the beginning of a video sequence because there are no prior frames to which you can refer. You can insert I-frames periodically in a coded video sequence for use as random access key frames and to eliminate errors that might have accumulated in the previous sequence of frames. I-frame coding is similar to the JPEG still-picture compression standard in that an 8-by-8 pixel *Discrete Cosine Transform* (DCT) is employed (as well as quantization and variable-length coding).

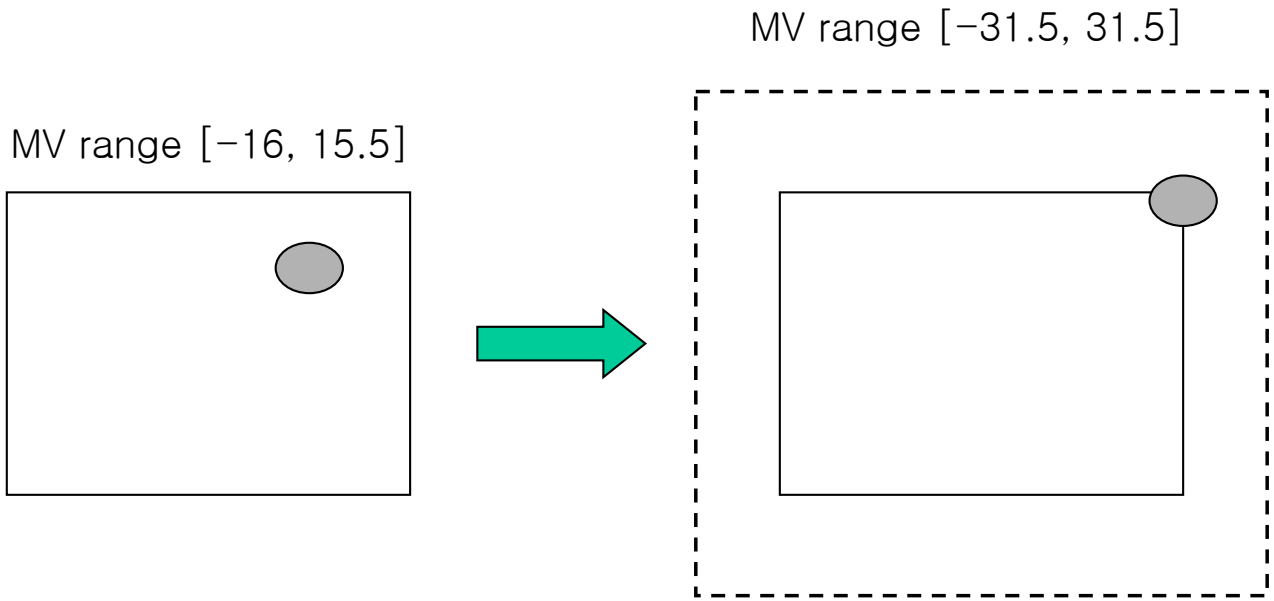
Inter-frame (also known as Predictive-frame or P-frame) coding occurs on the current frame that is being encoded based on a prediction by using the prior frame. P-frame coding relies on the fact that the content changes little from frame to frame. The encoder codes differences between frames and transmits them over the network to the decoder. These inter-frame differences can be significantly reduced (and in turn, coded much more efficiently) by taking advantage of the fact that you can approximate most changes by displacing 16-by-16 or 8-by-8 pixel blocks in the previous frame by a few pixels and subtracting them from the corresponding blocks in the target frame. The use of spatial displacement is known as motion compensation, and a motion vector specifies the spatial displacement for each block. Coding of this difference due to displacement is known as *Displaced Frame-Difference* (DFD) coding. The frame differences are then coded in a manner similar to the I-frames; that is, employing DCT, quantization, and variable-length encoding.

Bidirectional frame (also known as B-frame) coding is very similar to P-frame encoding with the addition of a prediction from the frame located immediately before the current frame to be encoded. You use the previous frame, the future frame, or a weighted average of both in order to predict each block to be encoded in the target frame. In some cases, the predictions fail to improve the coding efficiency for a given block—at which time the block is coded as if it were in an I-frame. Each block is thus coded along with a representation of zero, one, or two motion vectors. As with the P-frame encoding, the displaced frame difference is then encoded by using DCT, quantization, and variable-length encoding.

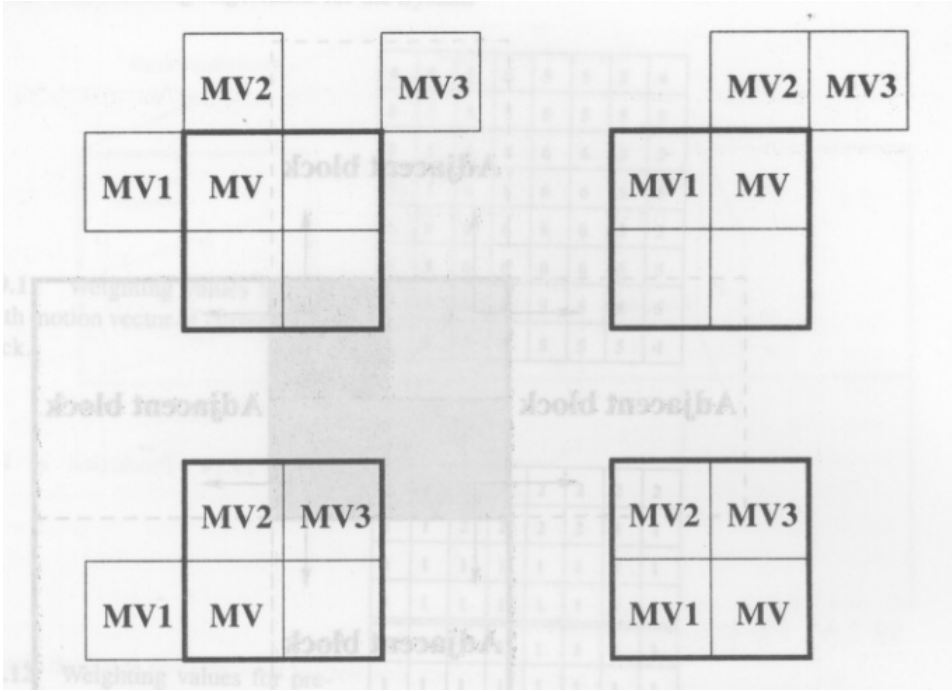
The bit-stream syntax for each picture is divided into smaller syntactical elements. The picture is divided into *Groups of Blocks* (GOBs), GOBs into *Macro Blocks* (MBs), MBs into blocks, and blocks into pixels. Normally, the encoder is restricted to producing its output within a certain bit rate based on the bandwidth of the network. On a 28.8 kbps modem, the encoder might be limited to a bit rate of, say, 15 kbps. Such a low bit rate is possible on a low frame resolution and on a frame rate of, for example, QCIF at 5Hz.

H.263 (+)

- Overview:
  - H.261+Unrestricted Motion Vector mode

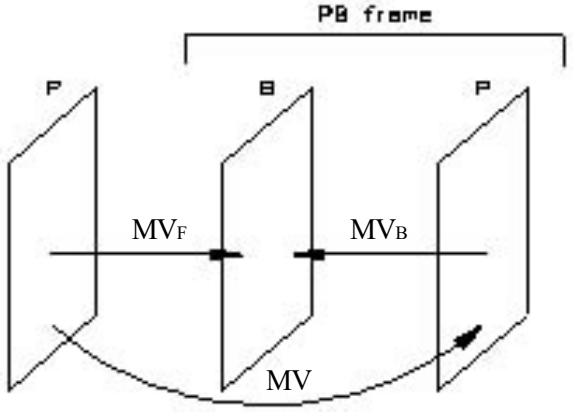


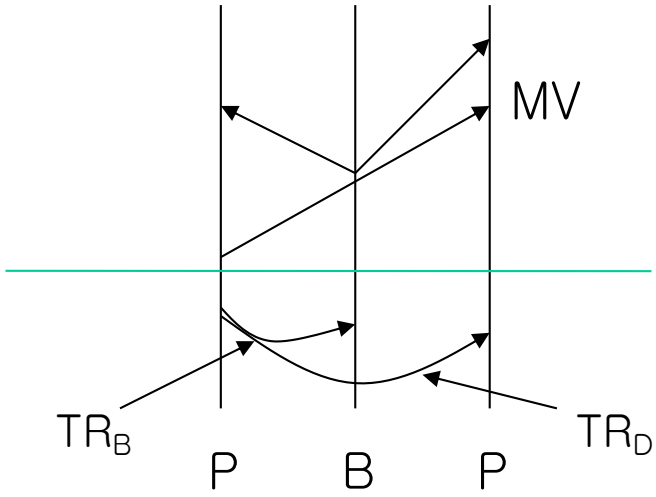
- Advance:
  - Advanced Prediction mode : 4MV por Macroblock



$$MVD_Y = MV_X - P_X$$
$$MVD_Y = MV_Y - P_Y$$
$$P_X = \text{Median}(MV_{1X}, MV_{2X}, MV_{3X})$$
$$P_Y = \text{Median}(MV_{1Y}, MV_{2Y}, MV_{3Y})$$

- Advance 3:
  - PB-frame mode





- $MV_F=(TR_B * MV) / TR_D + MV_D$
  >> $TR_D : TR_B = MV : MV_F$

- if MV_D is unequal to 0
  $MV_B = MV_F - MV - MV_D$
  $MV_F = (TR_B / TR_D) * MV + MV_D$

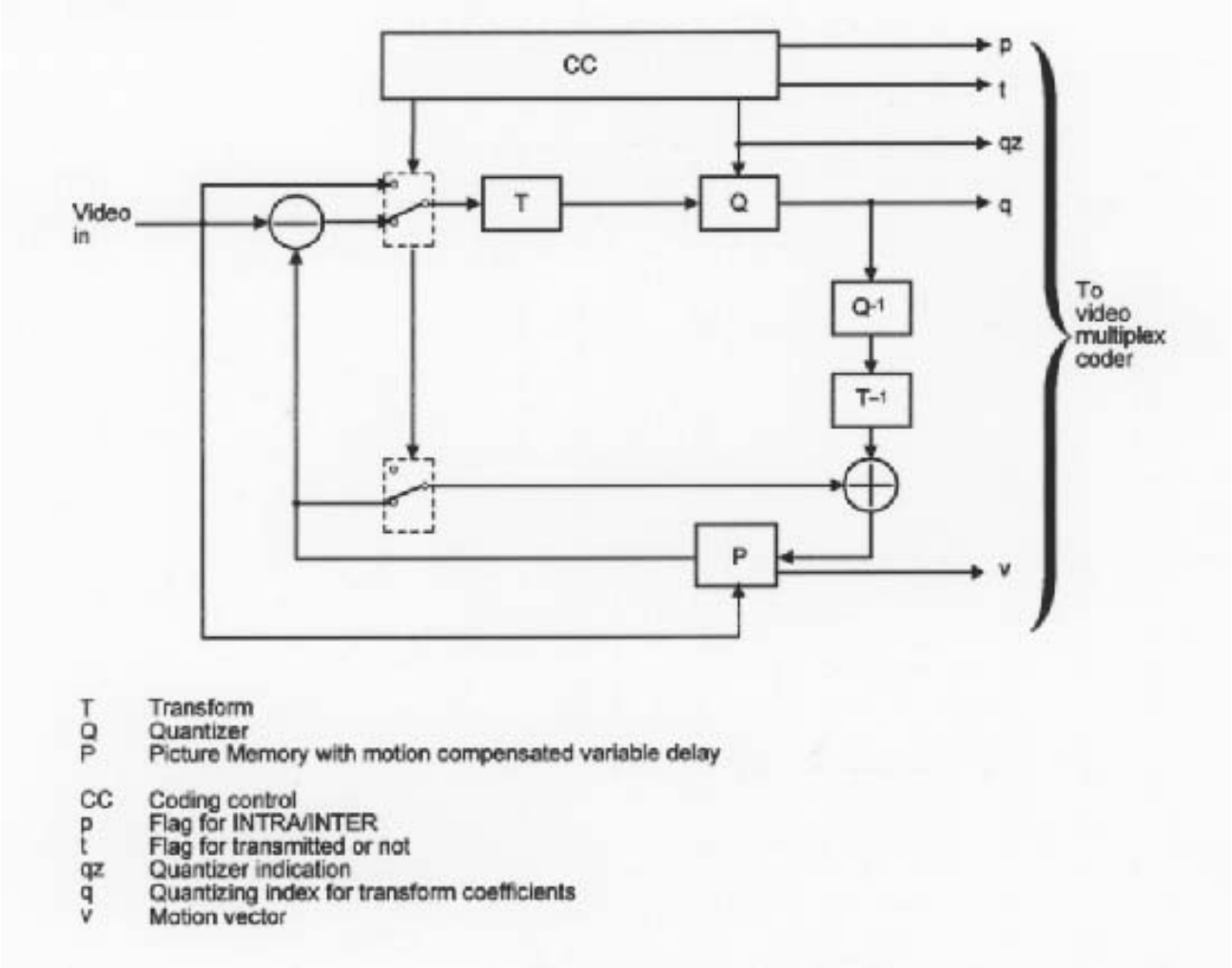
- if MV_D is equal to 0
  $MV_B = MV_F - MV$

H.263-Source Format

Picture Format

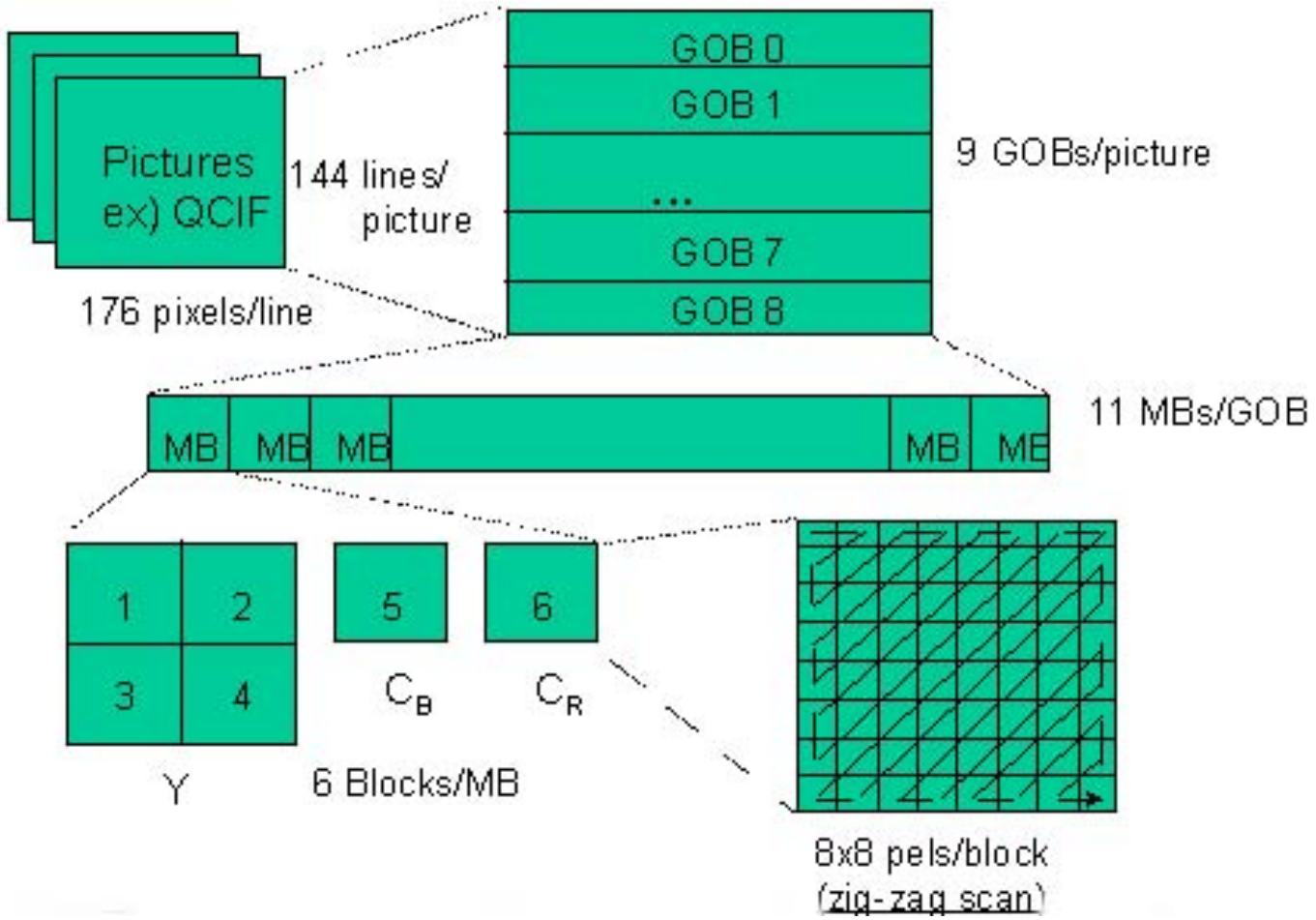
Picture Format	number of pixels for luminance (dx)	number of lines for luminance (dy)	number of pixels for chrominance (dx/2)	number of lines for chrominance (dy/2)
sub-QCIF	128	96	64	48
QCIF	176	144	88	72
CIF	352	288	176	144
4CIF	704	576	352	288
16CIF	1408	1152	704	576

H.263-Coder



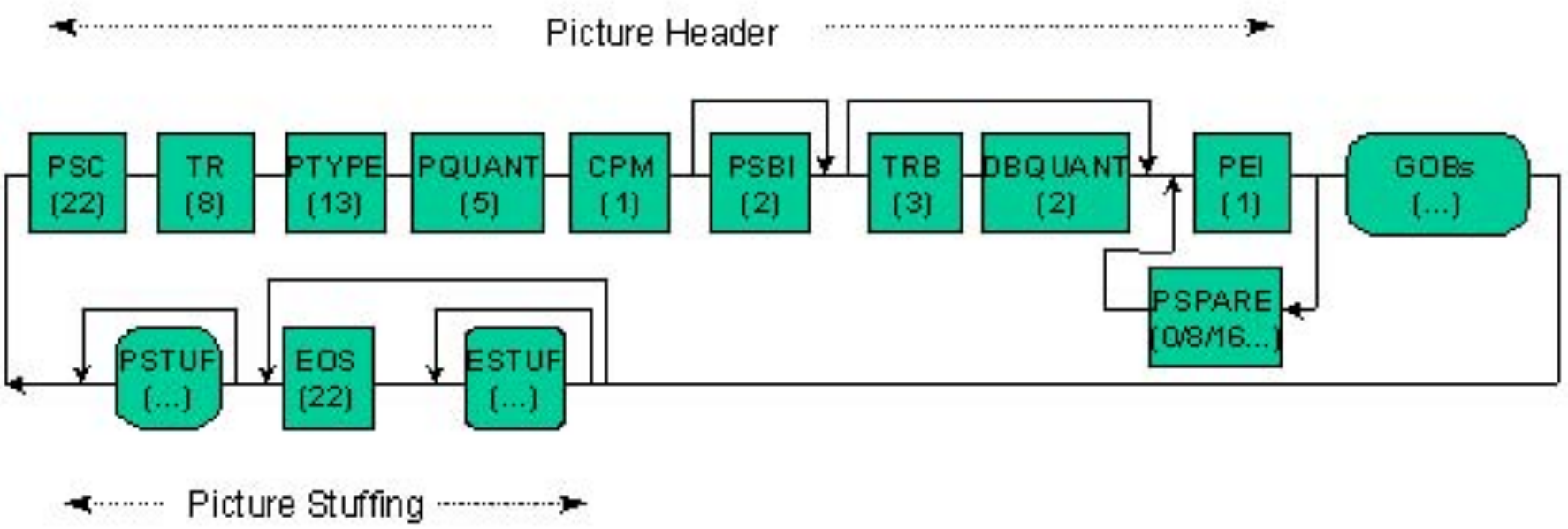
T	Transform
Q	Quantizer
P	Picture Memory with motion compensated variable delay
CC	Coding control
p	Flag for INTRA/INTER
t	Flag for transmitted or not
qz	Quantizer indication
q	Quantizing index for transform coefficients
v	Motion vector

H.263-format


Pictures
ex) QCIF
144 lines/
picture
176 pixels/line
GOB 0
GOB 1
...
GOB 7
GOB 8
9 GOBs/picture
MB
MB
MB
MB
ME
11 MBs/GOB
1
2
3
4
Y
5
C_B
6
C_R
6 Blocks/MB
8x8 pels/block
(zig-zag scan)

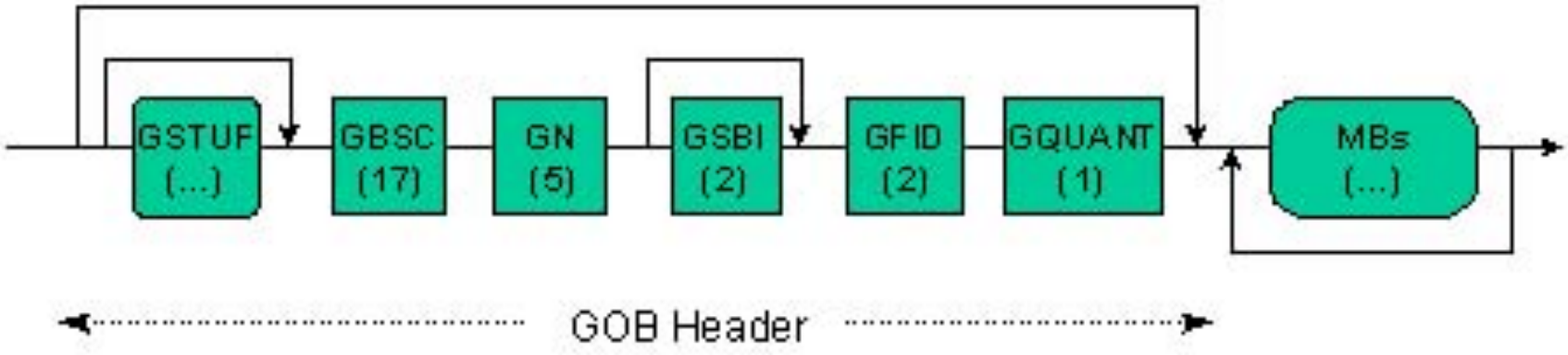
H.263-format

- Picture Layer



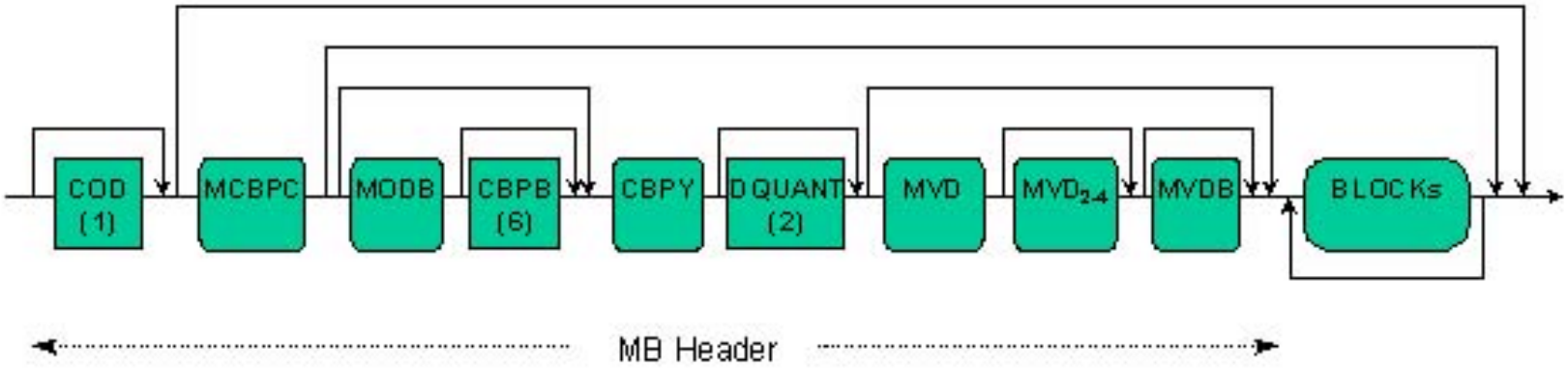
H.263-format

- Group of Blocks Layer



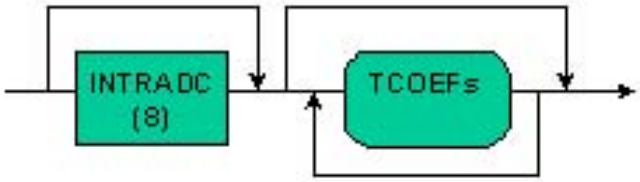
H.263-format

- Macroblock Layer



H.263-format

- Block Layer



INTRADC DC Coefficient for INTRA blocks
TCOEFF: Transform Coefficient

H.263+

The following options of H.263+ address the needs of wireless and packet-based networks:

Slice Structured Mode. This mode provides flexibility in partitioning the frame and transmitting the partitions in any order. The partitions must be on MB boundaries. You can compare this mode with fixed partitions and the fixed order of transmission of the GOBs in earlier coders. This mode can enhance error resilience and reduce video delay.

Independent Segment Decoding Mode. In this mode, the segments of the frame are encoded in such a way that you can independently decode them. This feature prevents errors in a segment from propagating to other segments of the frame.

Reference Picture Selection Mode. Normally, the most recently encoded frame is used as the reference frame for inter-frame prediction. But this mode avoids the use of an erroneous frame for reference by enabling an older error-free frame to be used instead. In this mode, there is a back channel through which the decoder provides the encoder information about which reference frame to use.

Bit-Stream Scalability Mode. In this mode, the bit stream consists of layers that represent different levels of video quality. The base layer guarantees a minimum level of video quality, and the enhancement layers progressively add higher levels of quality to the base layer. This mode enables receivers that have varying processing and bandwidth capabilities to receive different qualities of video. For example, if the bit rates of the base layer, enhancement layer 1, and enhancement layer 2 are 10 kbps, 20 kbps, and 60 kbps (respectively), then a receiver that is connected to a 28.8 kbps link will only pick the base layer with minimum video quality. The receiver on a 56 kbps link will additionally pick the enhancement layer 1 with higher video quality, and the receiver on a 128 kbps link will pick all three video layers that have the highest video quality.

We define three types of scalability modes as follows. You can use these three modes separately or together in order to create a layered, scalable bit stream:

Temporal Scalability. In this mode, the base layer consists of I- and P-frames and the enhancement layer consists of B-frames. This mode provides a considerably higher frame rate with little increase in bit rate. As an example, the base layer consists of QCIF, 10 frames/s, and 60 kbps while the enhancement layer consists of QCIF, 20 frames/s, and 80 kbps.

Spatial scalability. In this mode, the base layer consists of I- and P-frames that are down-sampled horizontally and vertically by two before being coded. The enhancement layer frames are horizontally and vertically two times the base layer. This mode provides a higher frame resolution. As an example, the base layer consists of QCIF, 10 frames/s, and 60 kbps while the enhancement layer consists of CIF, 10 frames/s, and 300 kbps.

SNR scalability. In this mode, the base layer consists of I- and P-frames and the enhancement layer consists of the difference between the original frames and the base-layer frames. This mode provides higher-fidelity frames with the same frame resolution. As an example, the base layer consists of QCIF, 10 frames/s, and 60 kbps, and the enhancement layer consists of QCIF, 10 frames/s, and 100 kbps. The decoded enhancement plus the base-layer video would be noticeably better than the decoded base-layer video only.

H.264

- El H.264 Advanced Video Coding (H.264/AVC) es un estándar de compresión de vídeo avanzada.

•También conocido como MPEG-4 Part 10, MPEG-4 AVC, MPEG-4 o JVT H.26L (L es por mucho tiempo).

•Primera versión publicada en el año 2003.

•Desarrollado por el equipo de vídeo Conjunta (JVT), un esfuerzo colectivo de la VCEG UIT-T y MPEG ISO / IEC.

H.264

Objetivos:

•Tener un diseño de video de alto rendimiento utilizando técnicas de codificación simple y eficiente.

•Mejorar el rendimiento de compresión.

•Apoyar una amplia variedad de servicios / aplicaciones.

•Dar cabida a soluciones frente a las restricciones de ancho de banda.

•Asegurar el apoyo a las condiciones de red hostil.

•Desarrollar una nueva parte (parte 10) de la familia MPEG-4 de las normas y una nueva recomendación UIT-T (H.264).

Propiedades:

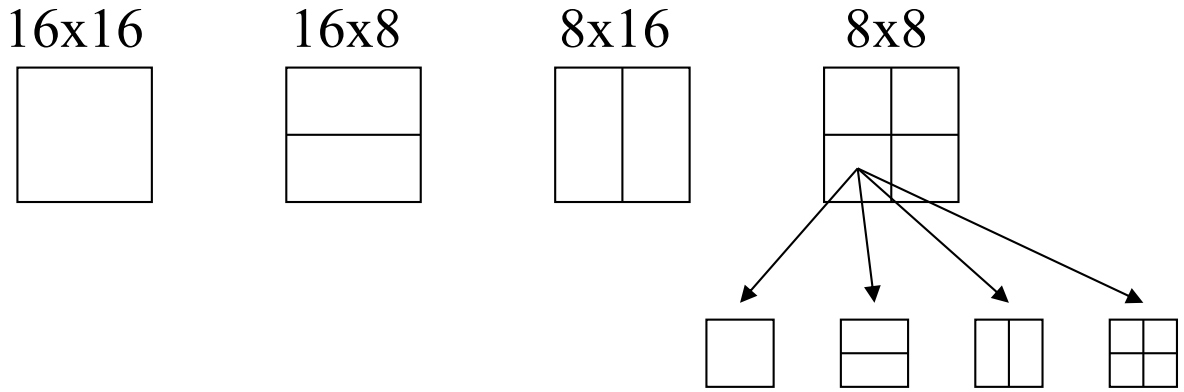
•Para aplicaciones de vídeo de tiempo real (bajo retardo, dependiente del contenido).
•La aplicación implementa para el mismo RD-algoritmo de optimización en todos los codificadores de vídeo.
•Configurados para el mejor funcionamiento RD-sin tener en cuenta de la complejidad de la parte de decodificación

H.264 vs H.263

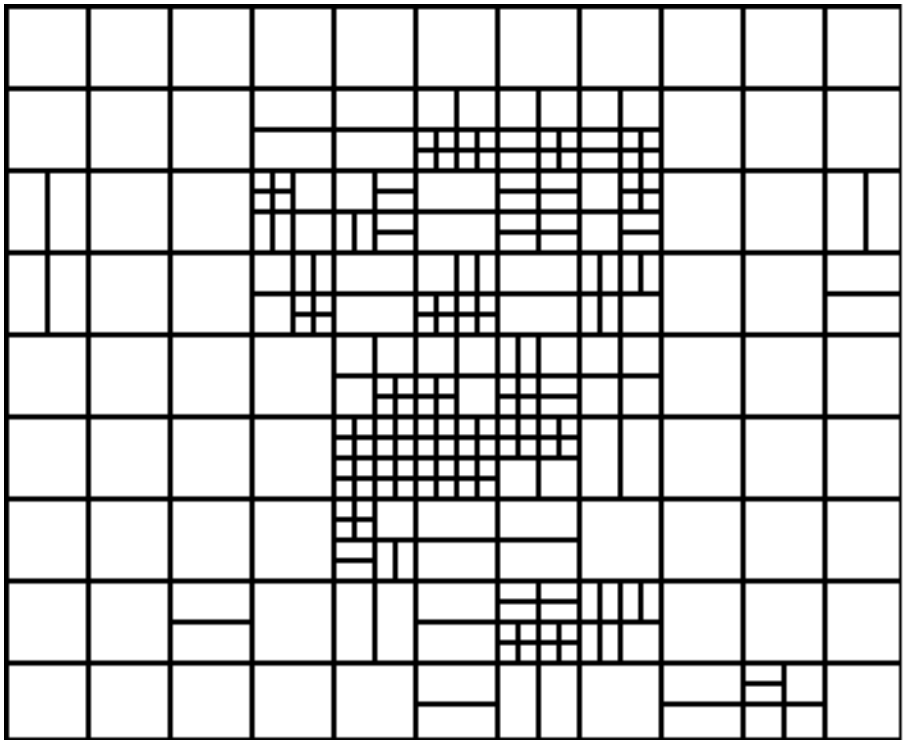
Component	H.263	H.264
Motion prediction	Bilinear 1/2 pixel interpolation, fixed block size, one reference frame	H.264 1/4 pixel interpolation, variable block size, multiple reference frames
Motion vector coding	BZIP2	Adaptive arithmetic coding
Transform	Floating point 8x8 DCT	Integer 4x4 DCT
Quantized coefficients	Single magic number	Table of empirically derived numbers

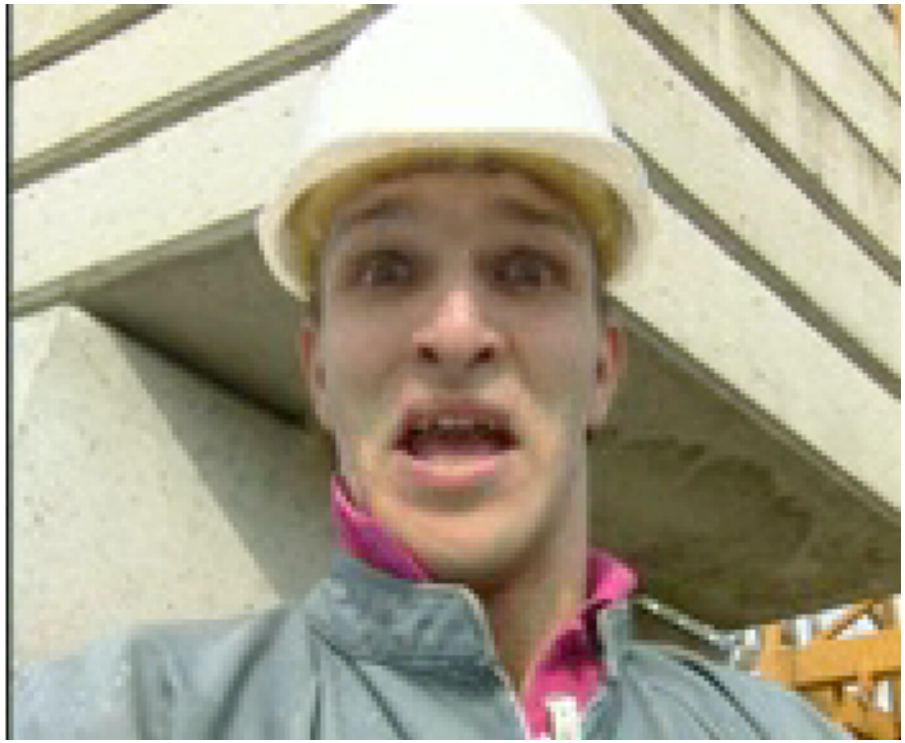
H.264

- Tamaño variable de MB (luminancia) mejora aún más la precisión de la predicción.
  - No hay MB de 16x16 fijos, como en H.263.
  - Los MB pueden ser divididos como se muestra a continuación.
- Varios bloques de referencia
  - Predecir a partir de los bloques más en el pasado.

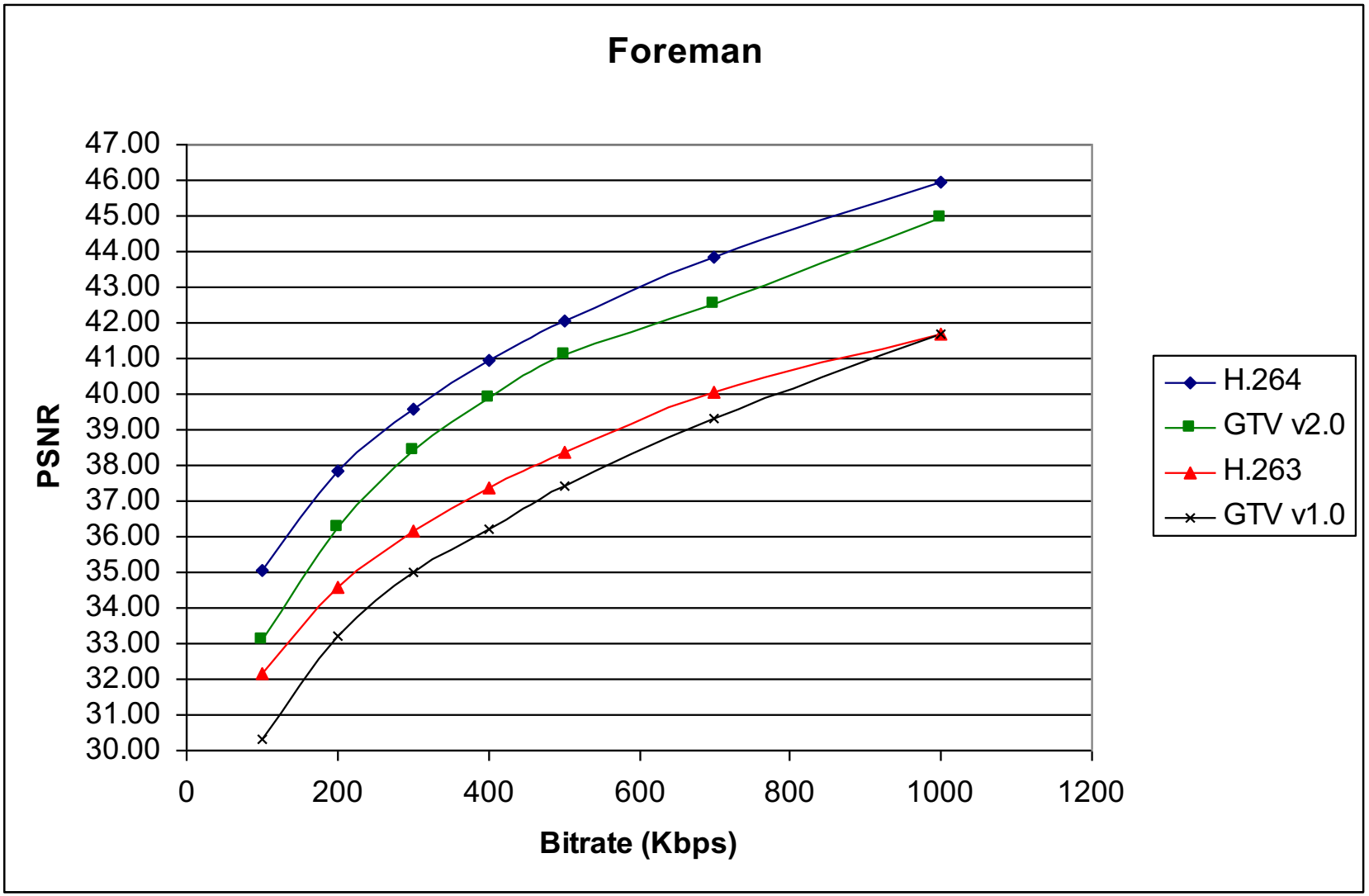


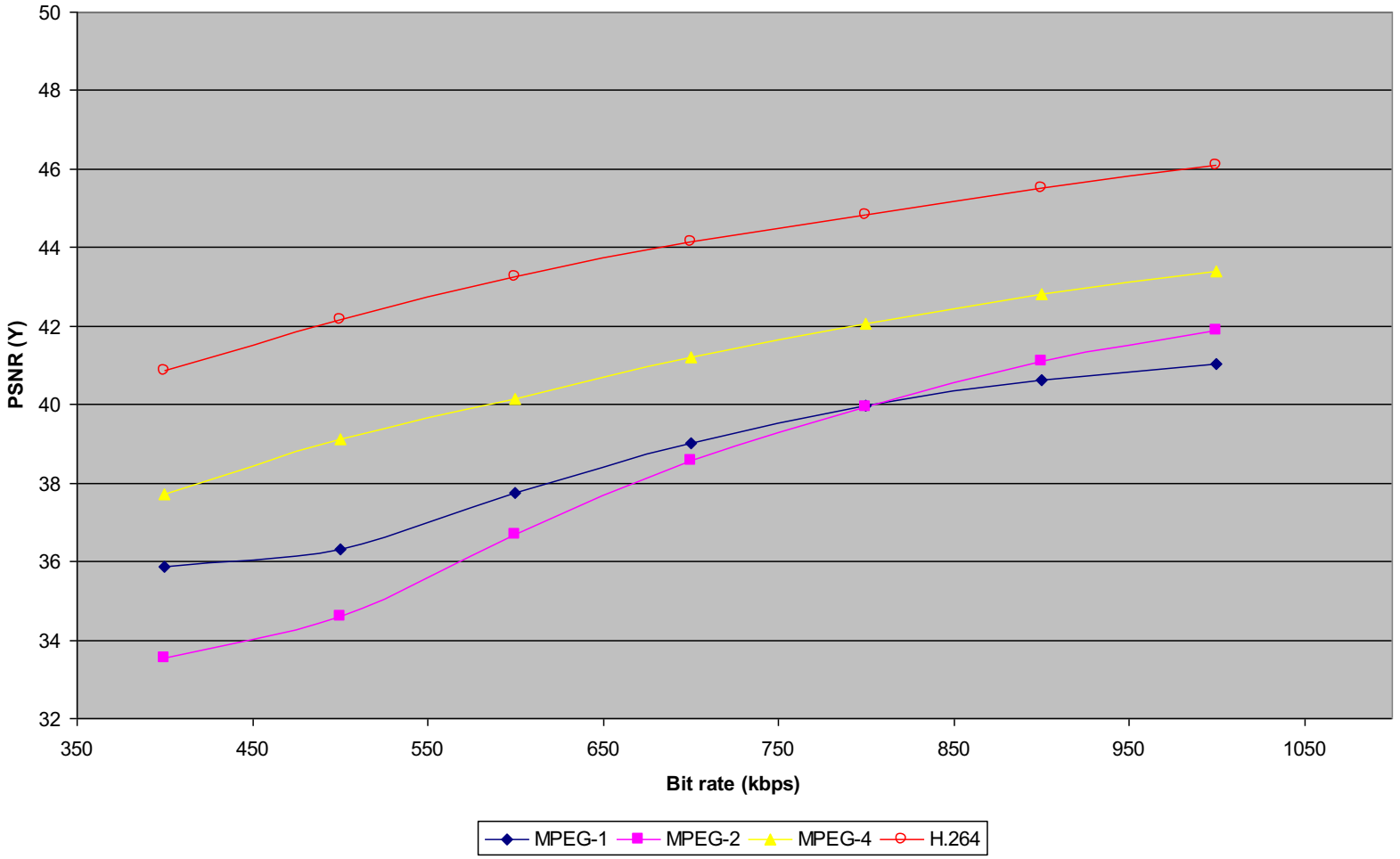
H.264





H.264


Foreman
PSNR
47.00
46.00
45.00
44.00
43.00
42.00
41.00
40.00
39.00
38.00
37.00
36.00
35.00
34.00
33.00
32.00
31.00
30.00
0
200
400
600
800
1000
1200
Bitrate (Kbps)
H.264
GTV v2.0
H.263
GTV v1.0


R-D Performance of MPEG Codecs
PSNR (Y)
50
48
46
44
42
40
38
36
34
32
350
450
550
650
750
850
950
1050
Bit rate (kbps)
MPEG-1
MPEG-2
MPEG-4
H.264

H.264 formats

Level	Max macroblocks per second	Max macroblocks per frame	Max video bit rate (VCL) BP, XP, MP (kbit/s)	HiP (kbit/s)	Hi10P (kbit/s)	Hi422P, Hi444PP (kbit/s)	Examples for high resolution @ frame rate (max stored frames)
1	1,485	99	64	80	192	256	128 × 96@30.9 (8) 176 × 144@15.0 (4)
1b	1,485	99	128	160	384	512	128 × 96@30.9 (8) 176 × 144@15.0 (4)
1.1	3,000	396	192	240	576	768	176 × 144@30.3 (9) 320 × 240@10.0 (3) 352 × 288@7.5 (2)
1.2	6,000	396	384	480	1,152	1,536	320 × 240@20.0 (7) 352 × 288@15.2 (6)
1.3	11,880	396	768	960	2,304	3,072	320 × 240@36.0 (7) 352 × 288@30.0 (6)
2	11,880	396	2,000	2,500	6,000	8,000	320 × 240@36.0 (7) 352 × 288@30.0 (6)
2.1	19,800	792	4,000	5,000	12,000	16,000	352 × 480@30.0 (7) 352 × 576@25.0 (6)
2.2	20,250	1,620	4,000	5,000	12,000	16,000	352 × 480@30.7(10) 352 × 576@25.6 (7) 720 × 480@15.0 (6) 720 × 576@12.5 (5)
3	40,500	1,620	10,000	12,500	30,000	40,000	352 × 480@61.4 (12) 352 × 576@51.1 (10) 720 × 480@30.0 (6) 720 × 576@25.0 (5)
3.1	108,000	3,600	14,000	17,500	42,000	56,000	720 × 480@80.0 (13) 720 × 576@66.7 (11) 1280 × 720@30.0 (5)
3.2	216,000	5,120	20,000	25,000	60,000	80,000	1,280 × 720@60.0 (5) 1,280 × 1,024@42.2 (4)

H.264 formats

Level	Max macroblocks per second	Max macroblocks per frame	Max video bit rate (VCL) BP, XP, MP (kbit/s)	HiP (kbit/s)	Hi10P (kbit/s)	Hi422P, Hi444PP (kbit/s)	Examples for high resolution @ frame rate (max stored frames)
4	245,760	8,192	20,000	25,000	60,000	80,000	1,280 × 720@68.3 (9) 1,920 × 1,080@30.1 (4) 2,048 × 1,024@30.0 (4)
4.1	245,760	8,192	50,000	62,500	150,000	200,000	1,280 × 720@68.3 (9) 1,920 × 1,080@30.1 (4) 2,048 × 1,024@30.0 (4)
4.2	522,240	8,704	50,000	62,500	150,000	200,000	1,920 × 1,080@64.0 (4) 2,048 × 1,080@60.0 (4)
5	589,824	22,080	135,000	168,750	405,000	540,000	1,920 × 1,080@72.3 (13) 2,048 × 1,024@72.0 (13) 2,048 × 1,080@67.8 (12) 2,560 × 1,920@30.7 (5) 3,680 × 1,536@26.7 (5)
5.1	983,040	36,864	240,000	300,000	720,000	960,000	1,920 × 1,080@120.5 (16) 4,096 × 2,048@30.0 (5) 4,096 × 2,304@26.7 (5)

H.263 on RTP

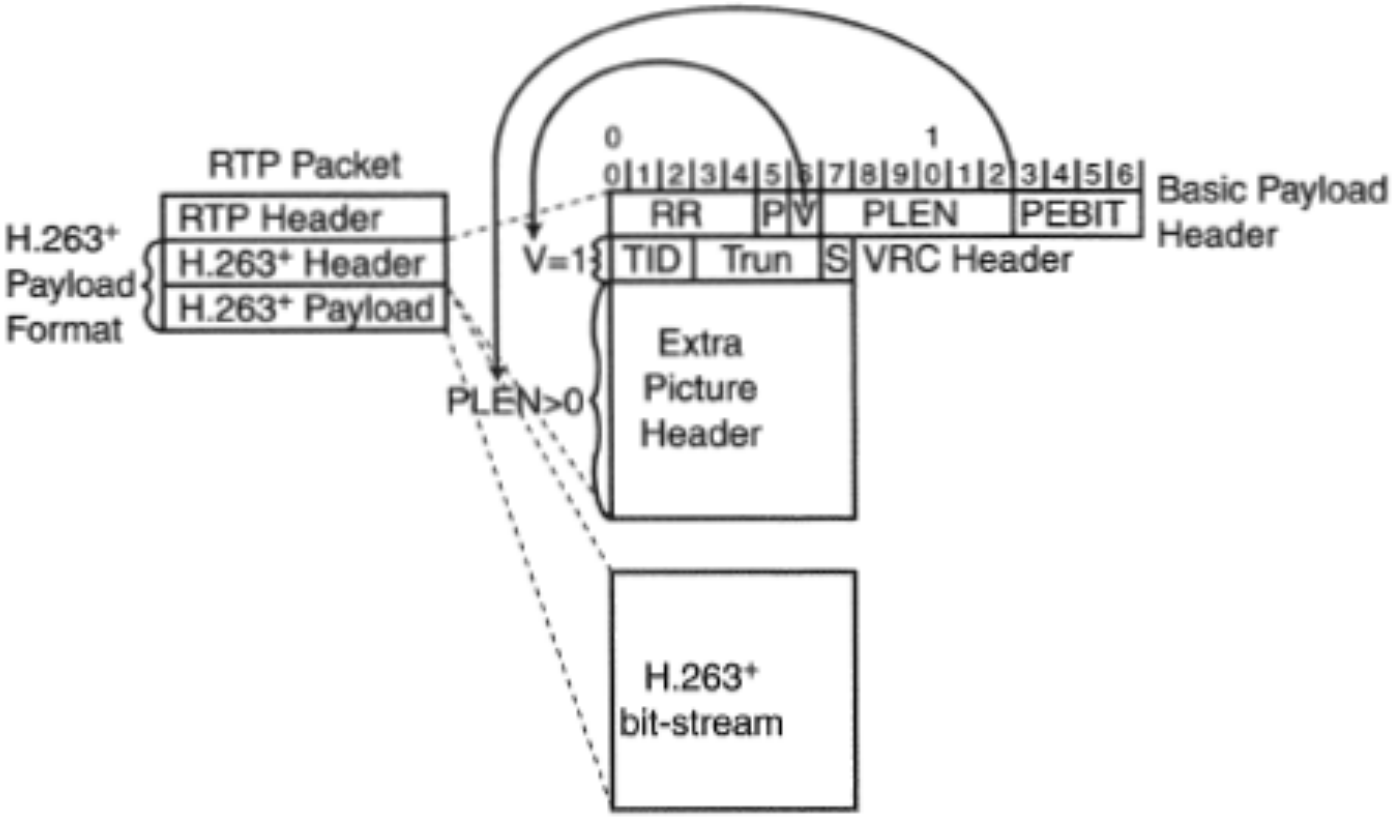
Payload Format for H.263

The H.263+ (H.263 version 2) payload format is specified in [WEN98] and includes the following components:

- The frequency of the RTP clock is 90kHz.
- The payload consists of the payload header followed by the H.263-bit stream.
- A static payload type is not assigned.

Figure 2.11 shows the payload header. The header contains a mandatory 16-bit basic header followed by an optional 8-bit *Video Redundancy Coding* (VRC) header, which is then followed by an optional variable-length extra picture header. The VRC mechanism provides error resilience at the packet level. This feature is in addition to the error resilience that is provided at the coder level in the H.263+ coder. VRC enables the transmission of multiple threads of independently coded P-frames so that errors in a frame cause distortions only within the thread that contains that frame. The effect of a packet loss, for example, would result in video being displayed at half the original frame rate (as opposed to no video displayed at all without VRC). The drawback of VRC is that coding efficiency is reduced, however, so you should only use it when you anticipate errors and when you find the other error-resiliency schemes of H.263 to be insufficient.

H.263 on RTP


RTP Packet
RTP Header
H.263+ Payload Format
H.263+ Header
H.263+ Payload
0 1
0|1|2|3|4|5|6|7|8|9|0|1|2|3|4|5|6
RR
P
V
PLEN
PEBIT
Basic Payload Header
V=1
TID
Trun
S
VRC Header
PLEN>0
Extra Picture Header
H.263+ bit-stream

H.263 on RTP

We specify the fields in the 16-bit basic header as follows:

RR (5 bits). This field is reserved for future use; we will assign it a value of zero.

P (1 bit). The start code in the H.263-bit stream begins with two bytes of zeros. The start code is unique and cannot occur anywhere else in the bit stream. This field specifies the beginning of a frame, GOB, slice, or the end of a video sequence. The P field, when set to 1, enables you to remove these two bytes of zeros—thus resulting in the reduction of the bit stream for transmission. The receiver recreates the bit stream by substituting two bytes of zeros in the start code when this field is set.

V (1 bit). If this field is set to 1, then it specifies the presence of the optional VRC header.

PLEN (6 bits). If this field is not zero, then it specifies the presence and size (in bytes) of the optional extra picture header. You can insert the extra picture header into packets that do not contain the start of a coded picture and that would otherwise not include the bit-stream picture header. Inserting extra picture header information in these cases improves the resiliency of the representation.

PEBIT (3 bits). This field is valid only if the extra picture header is present. This field specifies the number of least-significant bits of the last byte in the extra picture header that you should ignore.

H.263 on RTP

We specify the fields in the 8-bit optional VRC header as follows:

TID (3 bits). This field provides the identification number from 1 to 7 of up to seven threads. A thread that has a lower identification number should provide a better representation of the synchronization frame than the higher-numbered thread. We conventionally assign the identification number 0 to the thread from which the synchronization frame should be used. If thread 0 is corrupt, the decoder should use the next-higher error-free thread for the representation of the synchronization frame.

Trun (4 bits). This field provides a monotonically increasing modulo 16 count of the packet number within each thread. We use this field to detect packet loss within a thread.

S (1 bit). If this field is set to 1, then it specifies that the H.263+ bit stream contains a representation of the synchronization frame.

RTP redundancy

The main idea behind redundancy is to send information about the previous packet in the following packet. If the previous packet is lost, the following packet will provide the lost information. The disadvantage is in the increase in bandwidth and the extra delay. The payload format for redundancy is specified in [PERK97] and consists of multiple secondary or redundant payloads along with the primary nonredundant payload. An RTP packet containing a redundancy payload format appears in Figure 2.12. The RTP header provides information about the primary payload. The Payload Type field in the RTP header is of type Redundancy in order to indicate the format of the payload. The payload format consists of one or more headers for each secondary payload and the last header for the primary payload. The length of each secondary header is four bytes, and the primary header is one byte. Each header contains information about its media payload.

The fields in the header are as follows:

First (1 bit). If this field is set to 1, then another header follows. If this bit is 0, then this header is the last. The last header contains information about the primary payload.

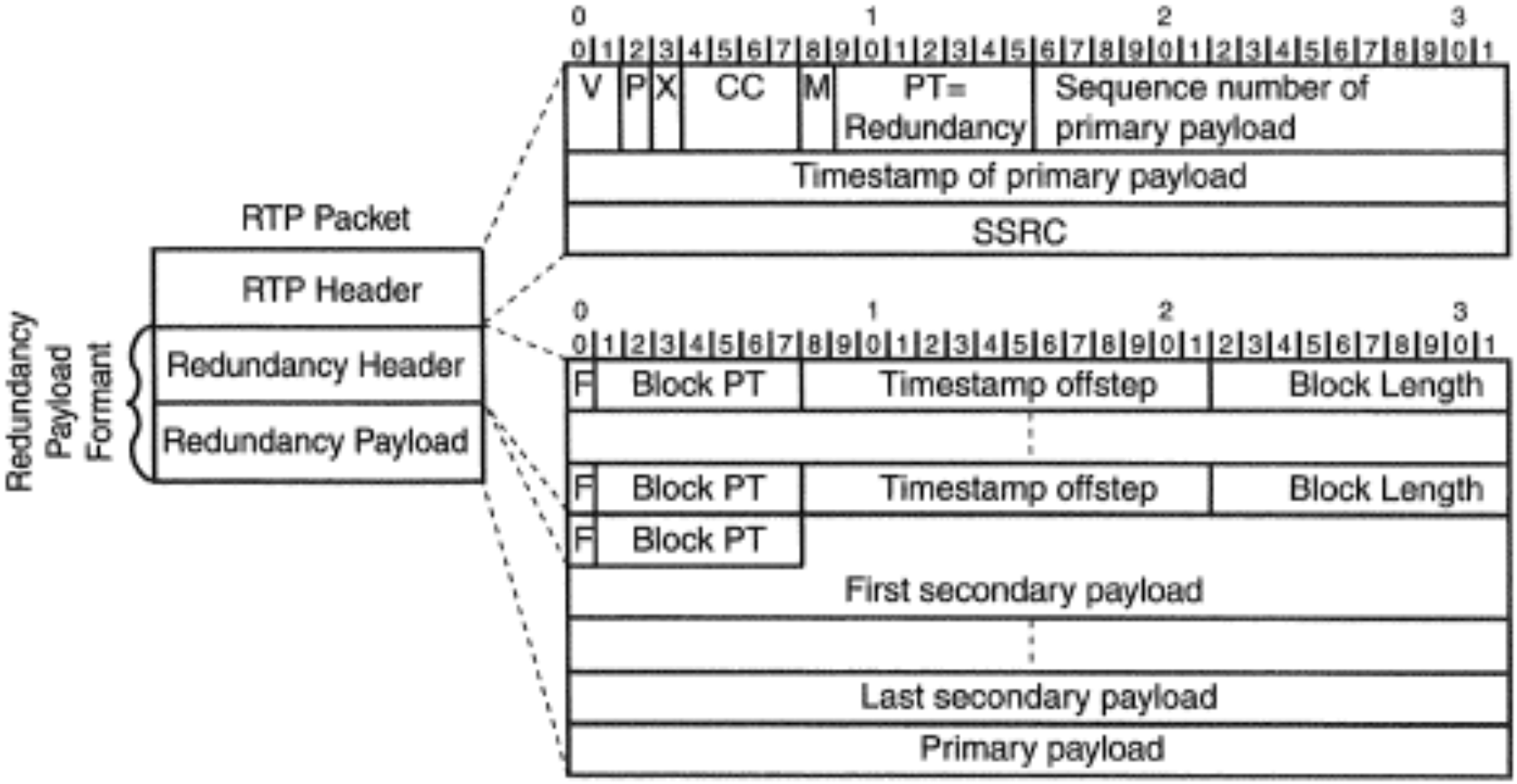
Block Payload Type (7 bits). This field specifies the type of the payload format used.

RTP redundancy

Timestamp Offset (14 bits). This field contains the offset from the timestamp in the RTP header. This value is an unsigned number, and you should subtract it from the RTP timestamp in order to determine the timestamp of the secondary media. The timestamp for the primary media is in the RTP timestamp.

Block Length (10 bits). This field contains the length (in bytes) of the payload that this header describes.

RTP redundancy



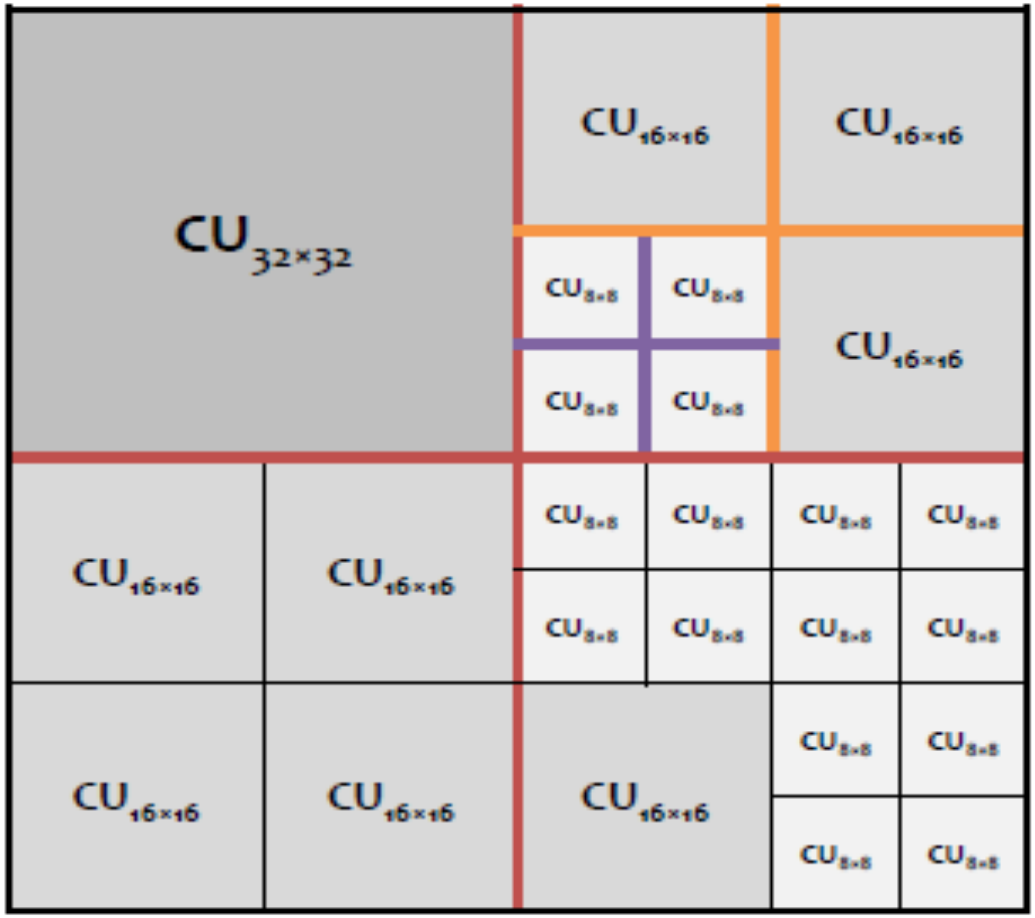
Coding Tree Block (CTB):
Picture is partitioned into square coding tree blocks (CTBs). The size N of the CTBs is chosen by the encoder (16x16, 32x32, 64x64). Luma CTB covers a square picture area of N × N samples and the corresponding chroma CTBs cover each (N/2) × (N/2) samples (in 4:2:0 format).

Coding Tree Units (CTU):
The luma CTB and the two chroma CTBs, together with the associated syntax, form a coding tree unit (CTU). The CTU is the basic processing unit similar to MB in prior standards.

Coding Block (CB):
Each CTB can be further partitioned into multiple coding blocks (CBs). The size of the CB can range from the same size as the CTB to a minimum size (8 × 8).

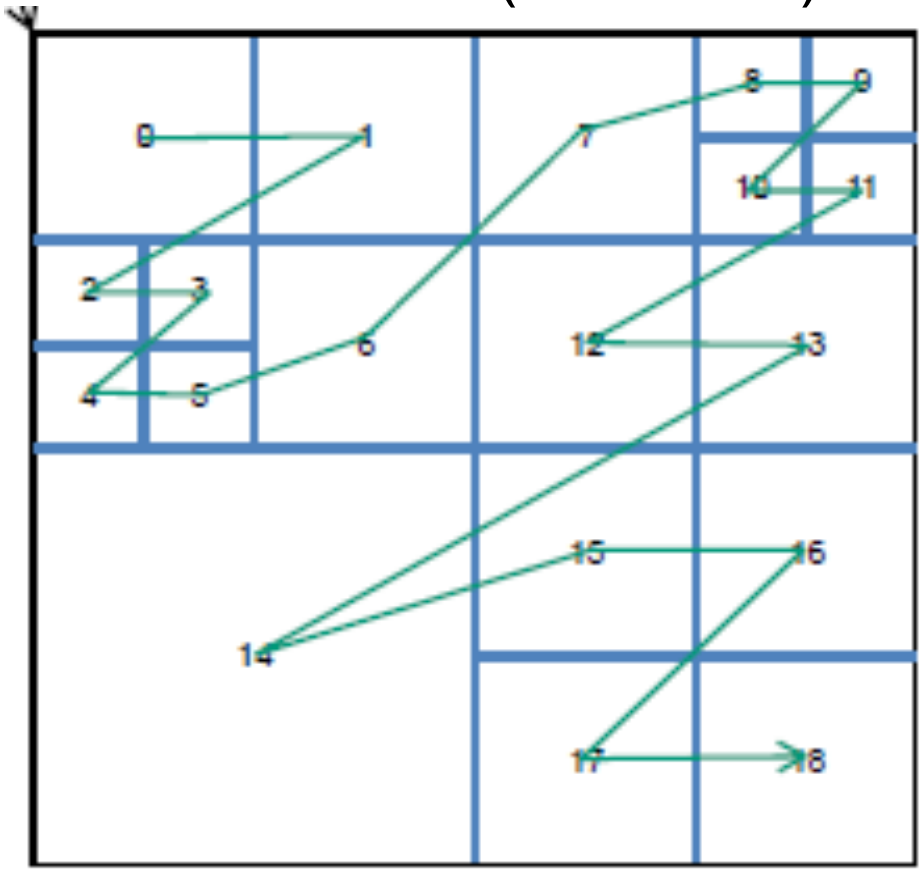
Coding Unit (CU)
The luma CB and the chroma CBs, together with the associated syntax, form a coding unit (CU). Each CU can be either Intra or Inter predicted.

CTU Syntax


CU32×32
CU16×16
CU16×16
CU8×8
CU8×8
CU8×8
CU8×8
CU16×16
CU16×16
CU16×16
CU8×8
CU8×8
CU8×8
CU8×8
CU8×8
CU8×8
CU8×8
CU8×8
CU16×16
CU16×16
CU16×16
CU8×8
CU8×8
CU8×8
CU8×8

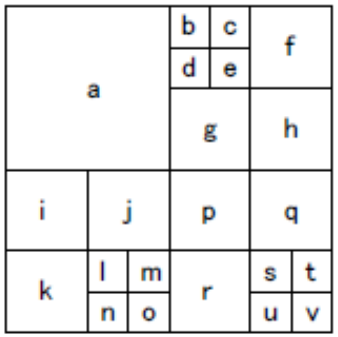
CTU Syntax

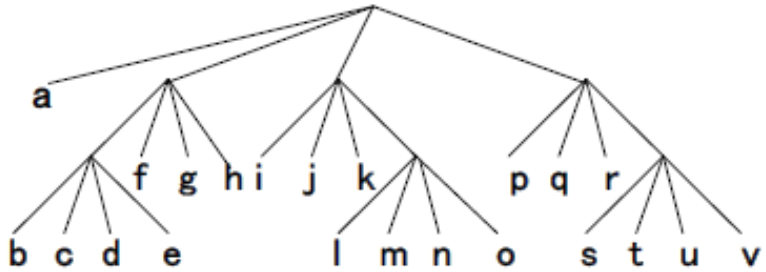
All CUs in a CTU are encoded (traversed) in **Z–Scan** order:



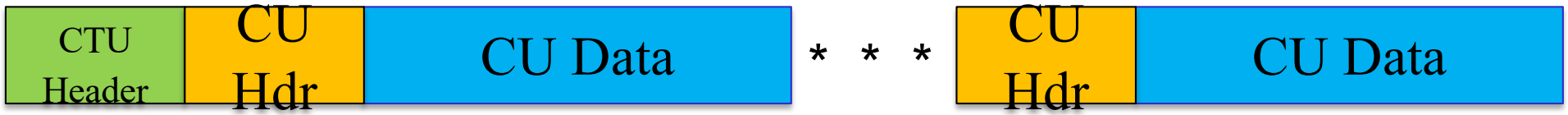
64x64 CTU

Formally CTU specifies quad-tree traversed in ‘inorder’.





Note: unlike to prior standards where MB header is followed by data, in HEVC ‘headers are dispersed’:

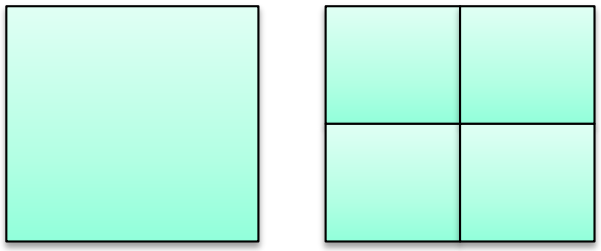


CTU

Prediction Block (PB):
Each CB is partitioned in 1, 2 or 4 prediction blocks (PBs).

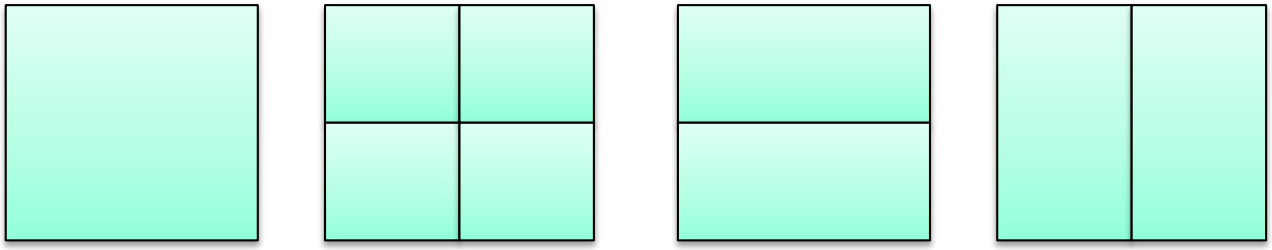
Prediction Unit (PU):
The luma PB and the chroma PBs, together with the associated syntax, form a prediction unit (PU).

Intra:



2Nx2N NxN (only if CB size is smallest CB size, i.e. CU = SCU)

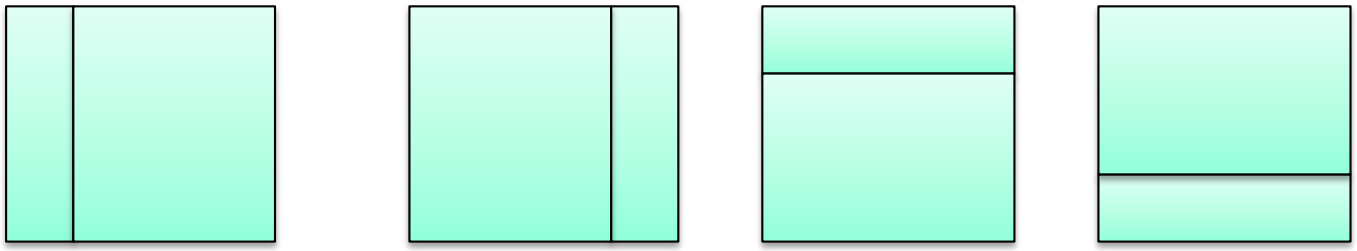
Inter:



2Nx2N NxN 2NxN Nx2N

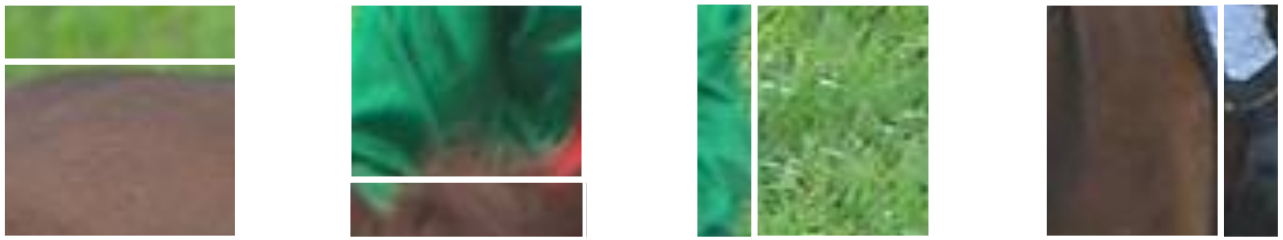
Sub-partitions (e.g. NxN) are allowed if CU = SCU

Inter Assymetric Partitions (conditioned by amp_enabled_flag in SPS**):**



nLx2N nRx2N 2NxnU 2NxnD

Why assymetric partitions are beneficial:



2NxnU 2NxnD nLx2N nRx2N

Notes:

- CUs are then divided into prediction units (PUs) of either intra-picture or inter-picture prediction type which can vary in size from 64 × 64 to 4 × 4.

- To limit worst-case memory bandwidth when applying motion compensation in the decoding process, prediction units coded using inter-picture prediction are restricted to a minimum size of 8 × 4 or 4 × 8 if they are predicted from a single reference (uniprediction) or 8 × 8 if they are predicted from two references (biprediction)

- Therefore, the smallest luma PB size is 4 × 8 or 8 × 4 samples (where 4x8 and 8x4 are permitted only for uni-directional predictions, no bi-prediction < 8x8 allowed).

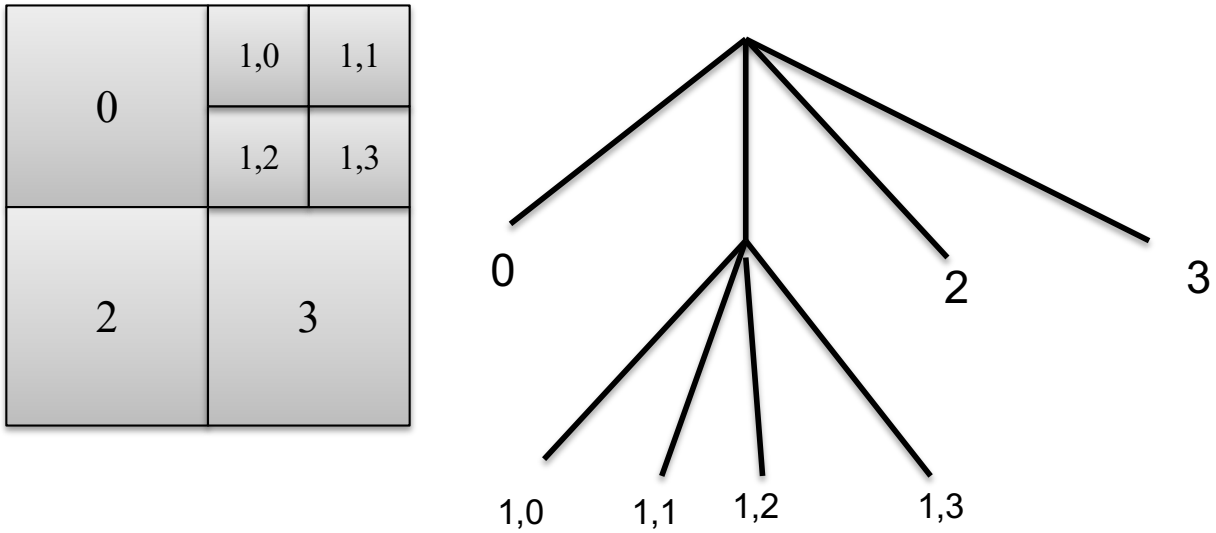
- Chroma PBs mimic corresponding luma partition with the scaling factor 1/2 for 4:2:0.

- Assymetric splitting is also applied to chroma CBs.

Transform Block (TB) :

To code the prediction residual, a CU is divided into a quadtree of transform units (TUs).TUs contain coefficients for spatial block transform and quantization. A TU can be 32×32, 16×16, 8×8, or 4×4 pixel block sizes

Example.
CU divided into two TU levels (the block #1 is split into four blocks):



H.265