

TV: TeleVisión – Plan 2010

Codificación híbrida: Recomendación ITU-T H.261



Contenido

1. Codificador híbrido

2. H.261

1. Codificador híbrido

- Definición:

- Codificador que resulta de la combinación de dos o más esquemas básicos de codificación.

- Módulos que contienen los codificadores actuales:

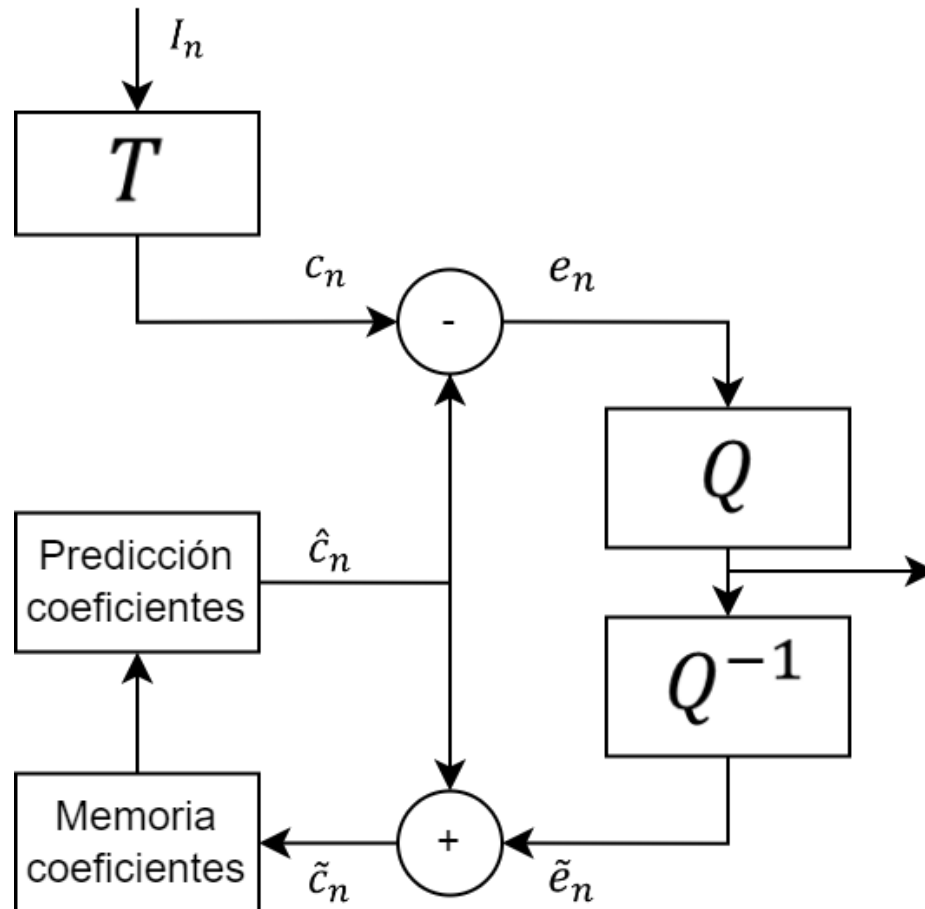
- Predicción → Explota la redundancia temporal de la secuencia.
- Transformación → Explota la redundancia espacial de las imágenes.
- Cuantificación → Necesaria para poder codificar eficientemente (alfabeto finito).

- Objetivo:

- Aunar las ventajas de la predicción y la transformación para mejorar el rendimiento global del codificador (mayor compresión).

1. Codificador híbrido

• Codificador híbrido transformación – predicción:



1. Cada imagen se divide en bloques 2D de $M \times N$ píxeles.

2. Sobre cada bloque se aplica una transformación 2D (e.g., DCT) que da lugar a bloques de coeficientes.

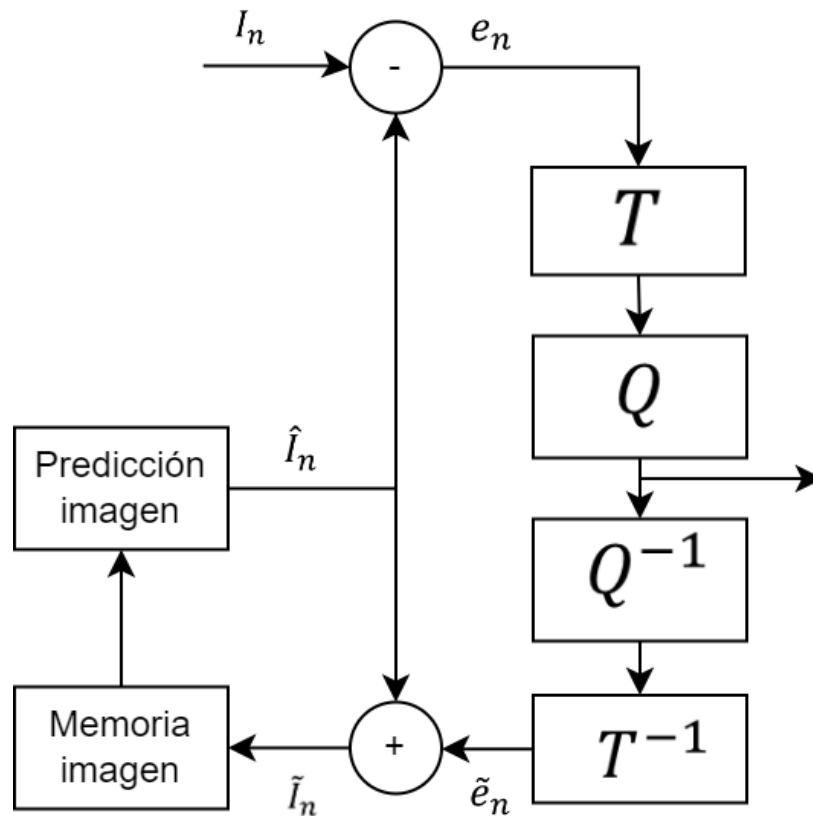
3. Se calcula una predicción a partir de coeficientes reconstruidos.

4. La predicción se resta a los coeficientes actuales y se obtienen coeficientes de error.

5. El error se cuantifica y transmite.

1. Codificador híbrido

• Codificador híbrido predicción – transformación:



1. Cada imagen se divide en bloques 2D de tamaño $M \times N$.

2. Se calcula una predicción de cada bloque a partir de imágenes previas decodificadas.

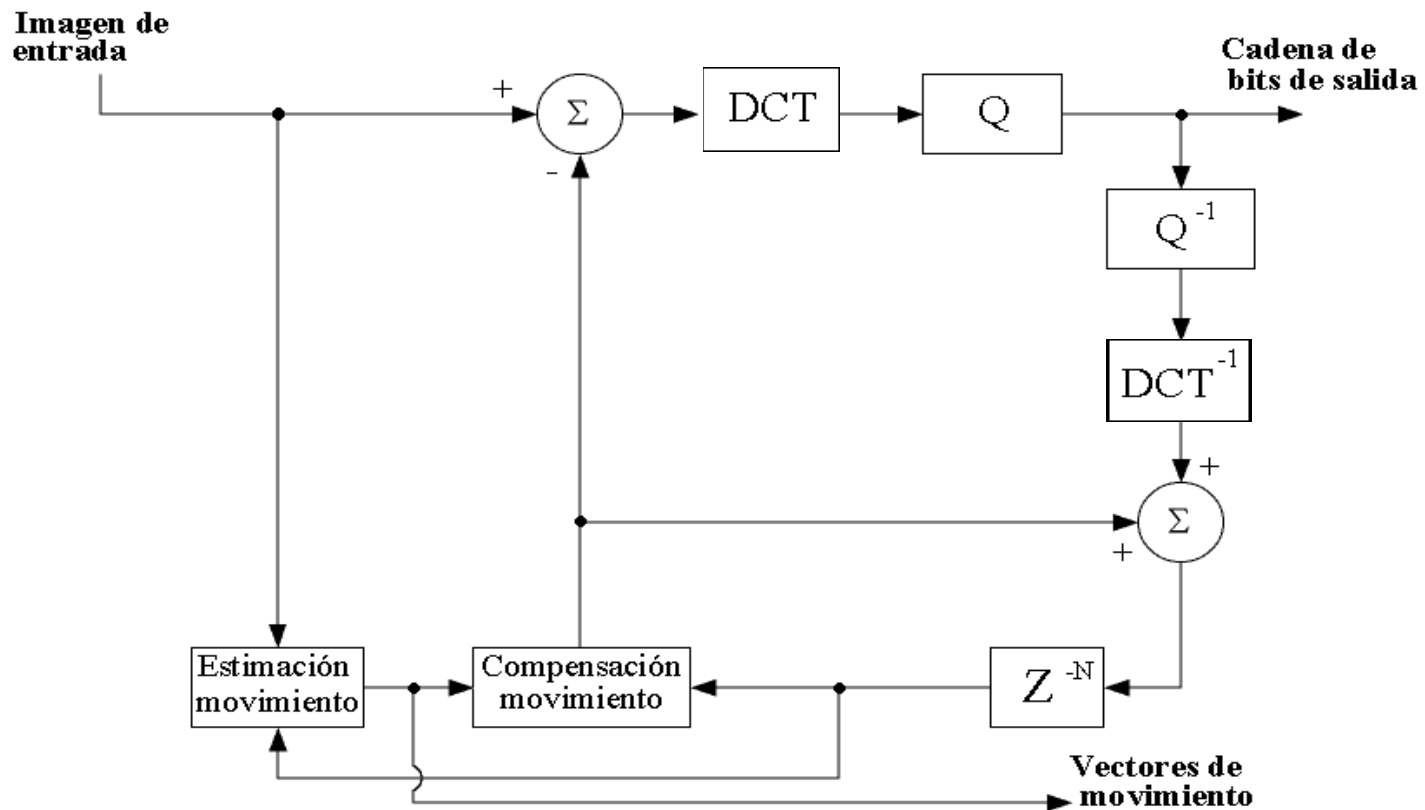
3. Se obtiene una imagen de error (resta de cada bloque con su predicción).

4. Sobre cada bloque de error se aplica una transformación 2D que da lugar a bloques de coeficientes.

5. Los bloques transformados se cuantifican y transmiten.

1. Codificador híbrido

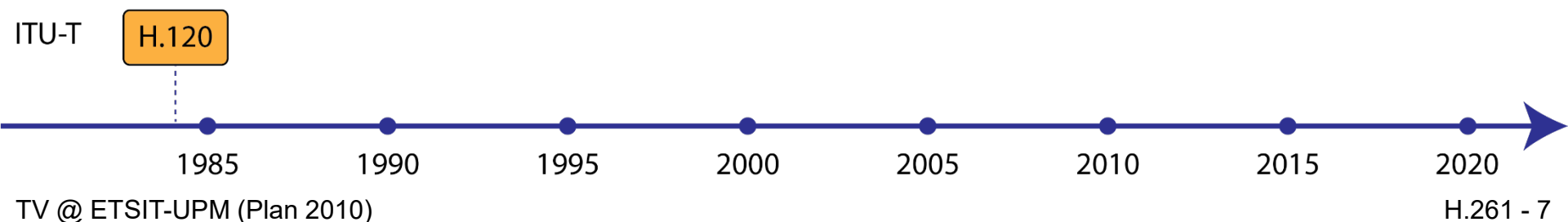
Esquema típico de un codificador híbrido predictivo transformacional:



1. Codificador híbrido

H.120 (1984):

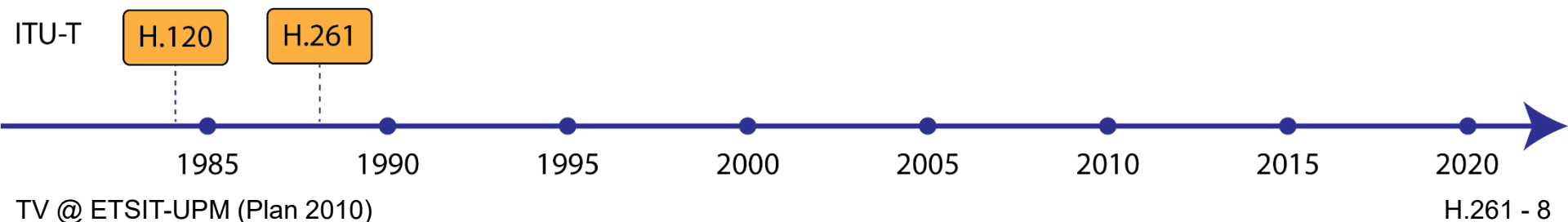
- ITU-T: *International Telecommunication Union (Telecommunication Standardization Sector)*.
- Primer estándar de compresión de video digital.
- El video tenía poca calidad → Se hicieron pocas implementaciones.
 - No hay codecs que lo utilicen.



1. Codificador híbrido

H.261 (1988):

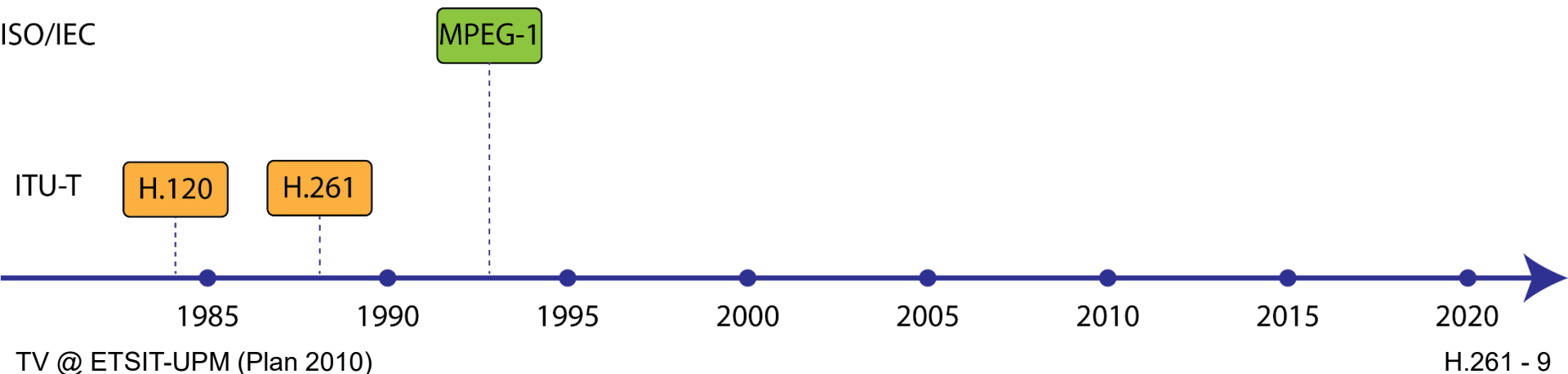
- ITU-T.
- Define métodos de codificación y decodificación de vídeo sobre redes RDSI.
- Aplicaciones con video “sencillo” (e.g., videoconferencias).
- Primer estándar considerado “práctico”.



1. Codificador híbrido

MPEG-1 – parte 2 (1993):

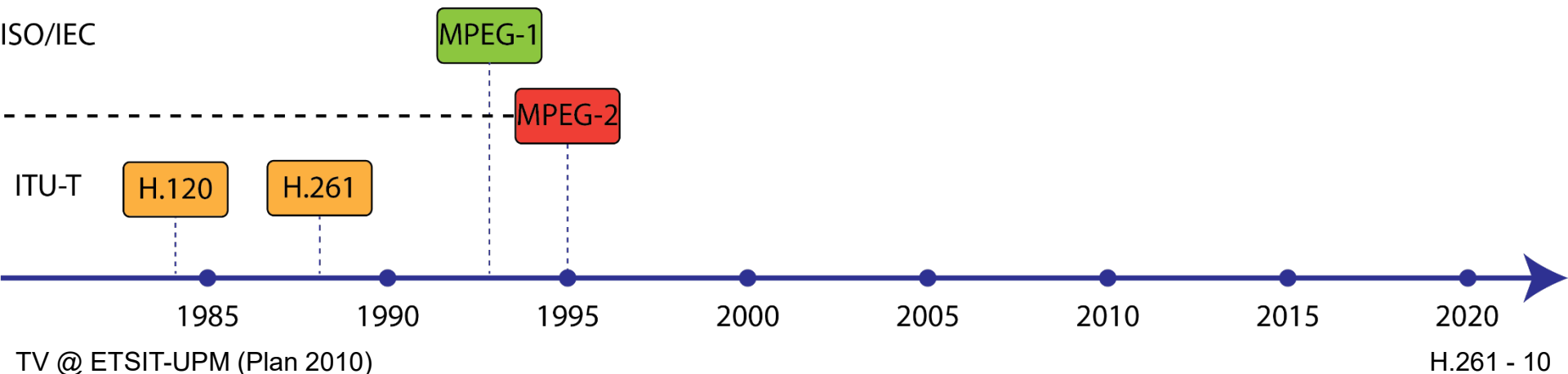
- MPEG: *Moving Picture Experts Group*
 - ISO: *International Organization for Standardization*
 - IEC: *International Electrotechnical Commission*
- Para ser utilizado en formato vídeo CD (calidad similar al VHS doméstico).
- Vídeo “complejo” (todo tipo de contenido).



1. Codificador híbrido

MPEG-2 – parte 2 / H.262 (1995):

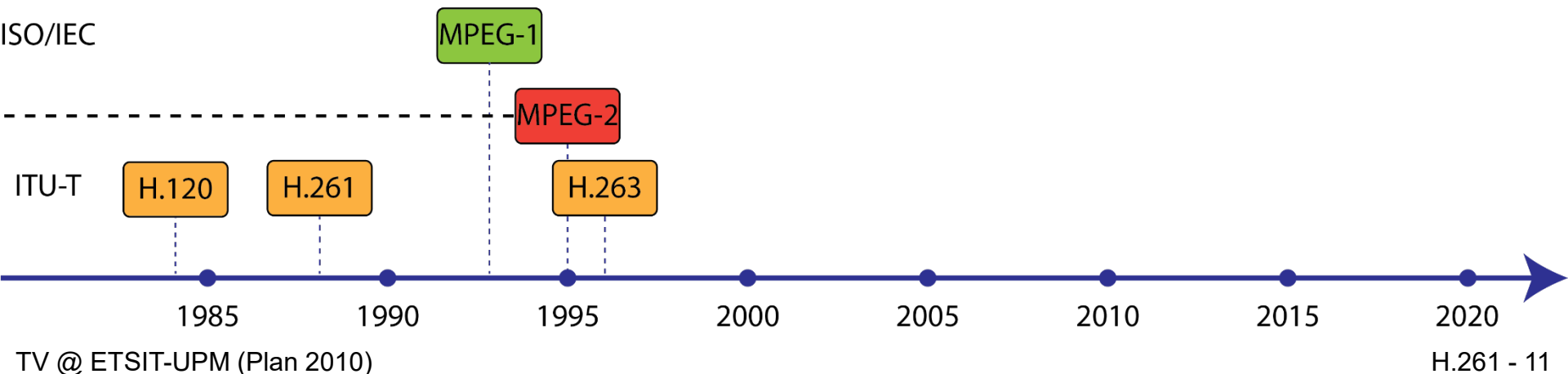
- MPEG: *Moving Picture Experts Group*
 - ISO: *International Organization for Standardization*
 - IEC: *International Electrotechnical Commission*
 - ITU-T: *International Telecommunication Union*
- Utilizado en transmisión (DVB) y almacenamiento (SVCD y DVD).
- Soporte para vídeo entrelazado.



1. Codificador híbrido

H.263 (1996):

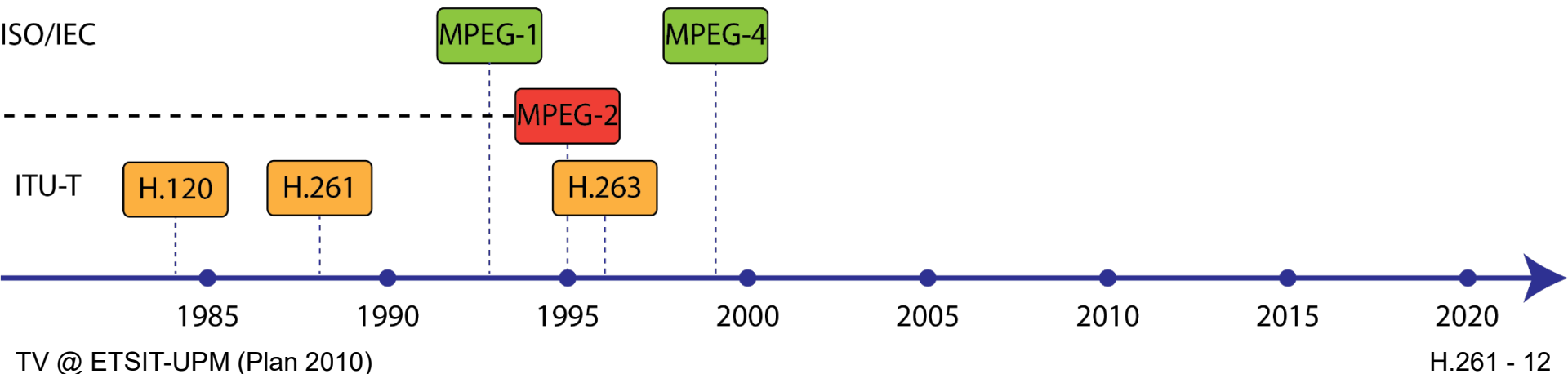
- Publicado por la ITU-T.
- Concebido inicialmente para vídeo-conferencias (tasas bajas de bit y poco movimiento).
- Acabó siendo usado en aplicaciones web como, por ejemplo, flash video.



1. Codificador híbrido

MPEG-4 – parte 2 (1999):

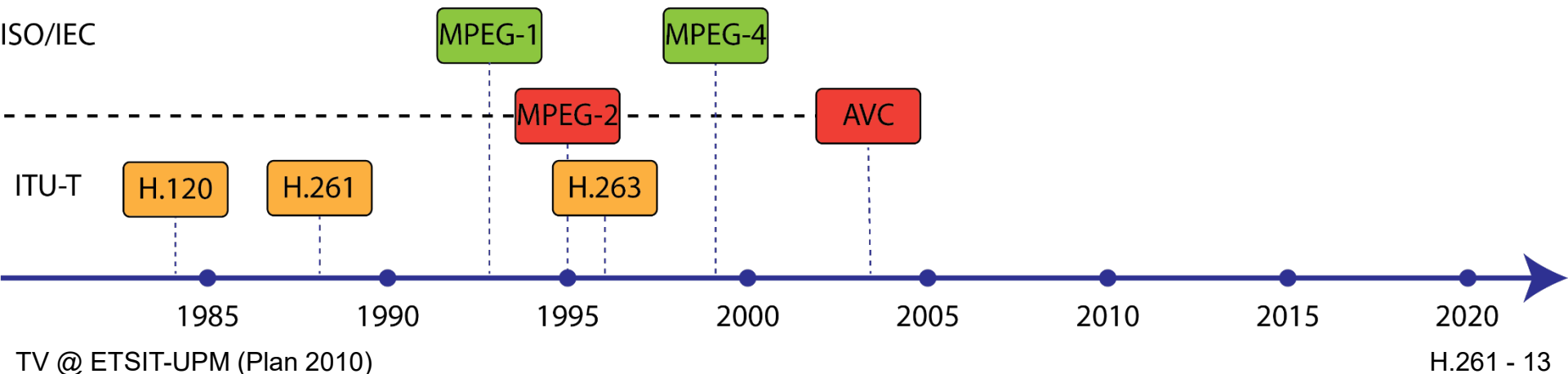
- MPEG: *Moving Picture Experts Group*
 - ISO: *International Organization for Standardization*
 - IEC: *International Electrotechnical Commission*
- También conocido como MPEG-4 Visual.
- Es una evolución de H.263.



1. Codificador híbrido

MPEG-4 – parte 10 / H.264 / AVC (2003):

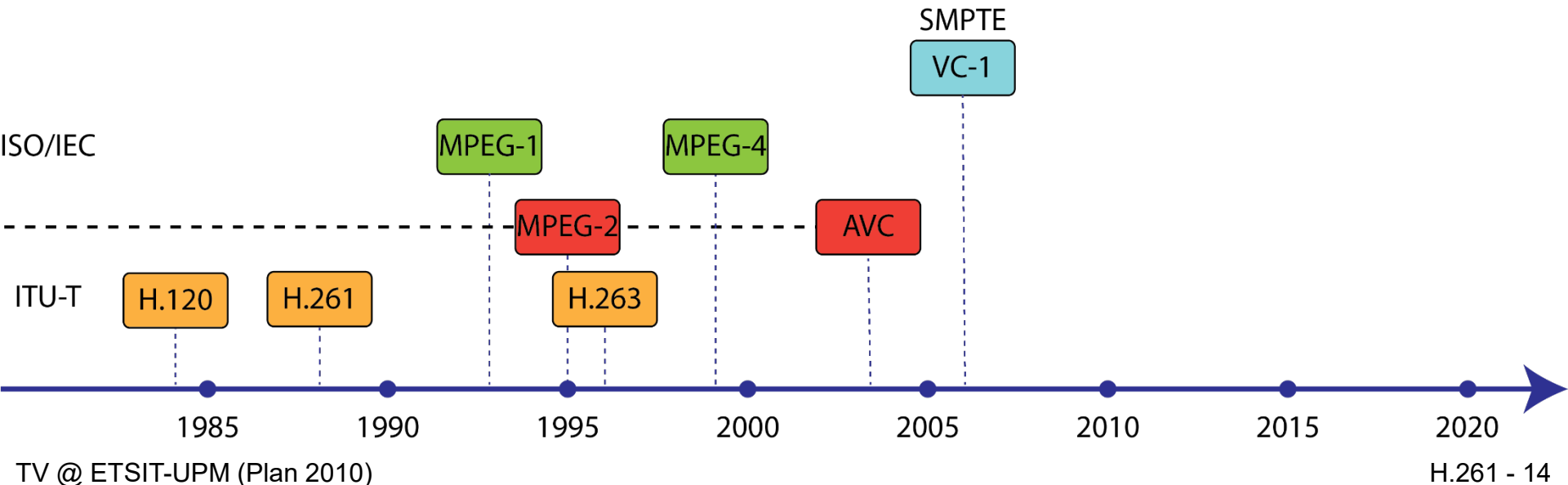
- JCT-VC: *Joint Collaborative Team on Video Coding*
 - MPEG (ISO + IEC)
 - VCEG (*Video Coding Experts Group*) ← ITU-T
- Niveles de compresión muy superiores a sus predecesores.
- Utilizado en: HDTV, Blu-ray, Internet.



1. Codificador híbrido

VC-1 (2006):

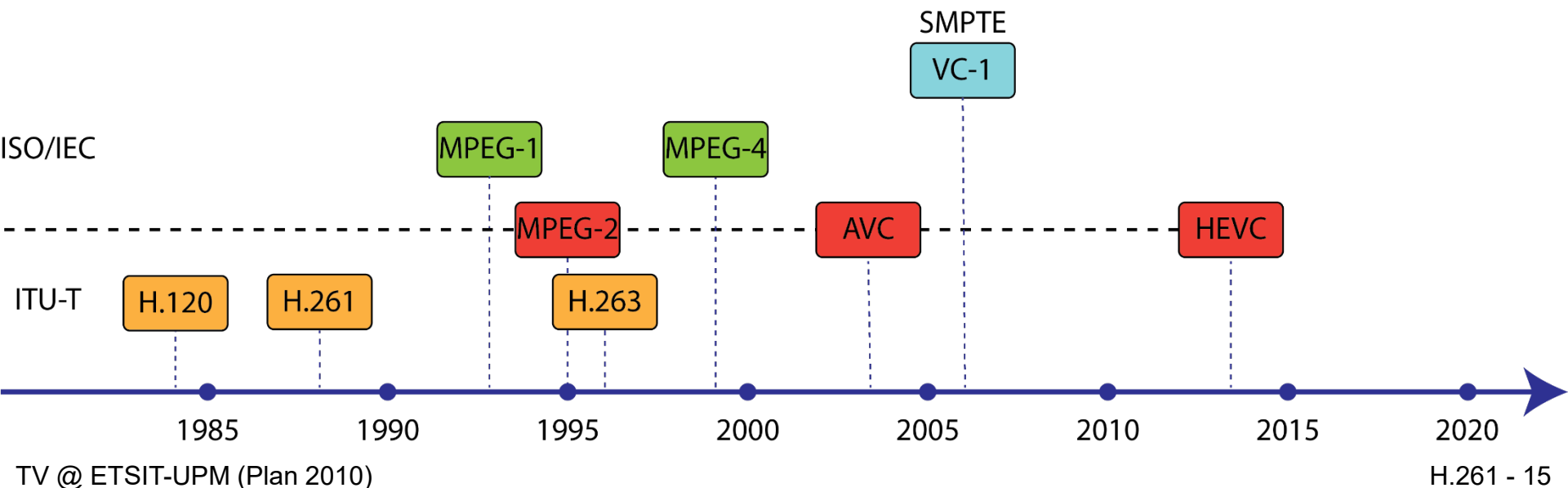
- SMPTE: *Society of Motion Picture and Television Engineers*
- Alianza entre Microsoft y otras tantas compañías.
- Su implementación más popular en Windows Media Video 9.
- Soportado en HD DVB y Blu-ray.
- Códec “oficial” para vídeo en Xbox 360.



1. Codificador híbrido

MPEG-H – Parte 2 / H.265 / HEVC (2013):

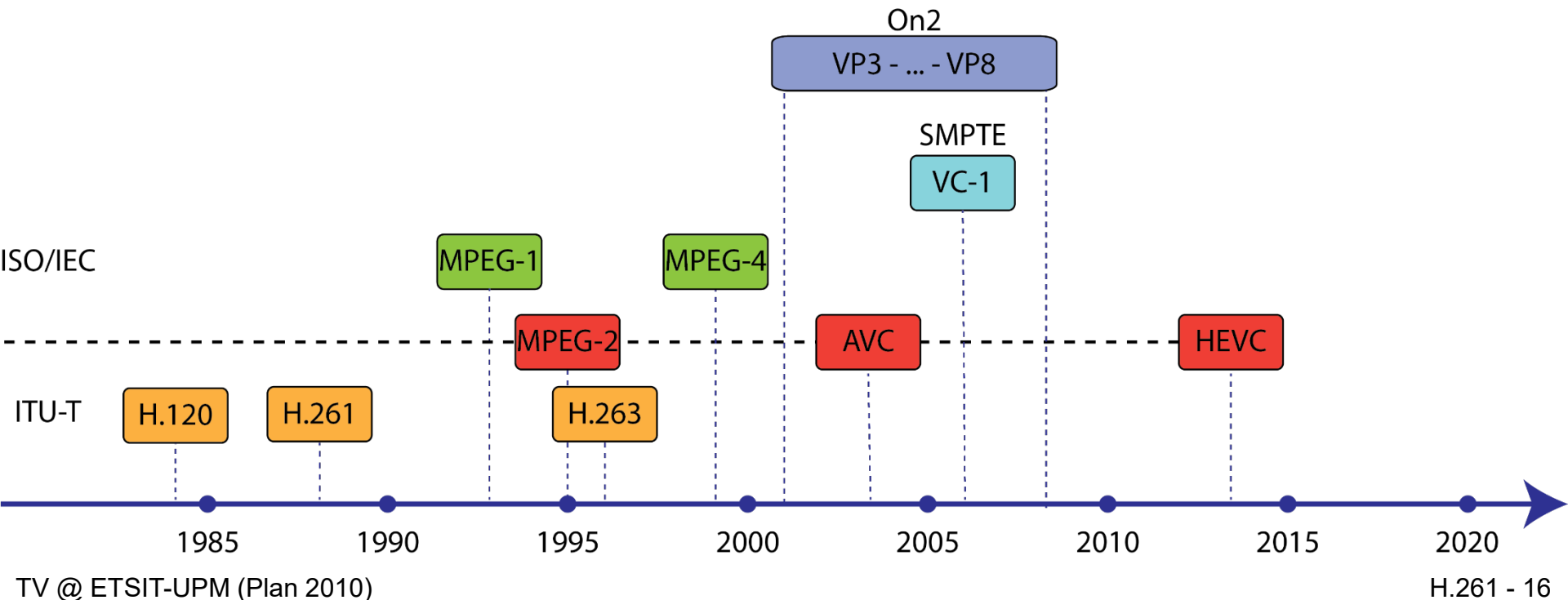
- JCT-VC: *Joint Collaborative Team on Video Coding*
- Concebido para UHDTV y 4k.
- Grandes novedades respecto a los anteriores (e.g., procesamiento en paralelo).



1. Codificador híbrido

VPx (2001 – 2008):

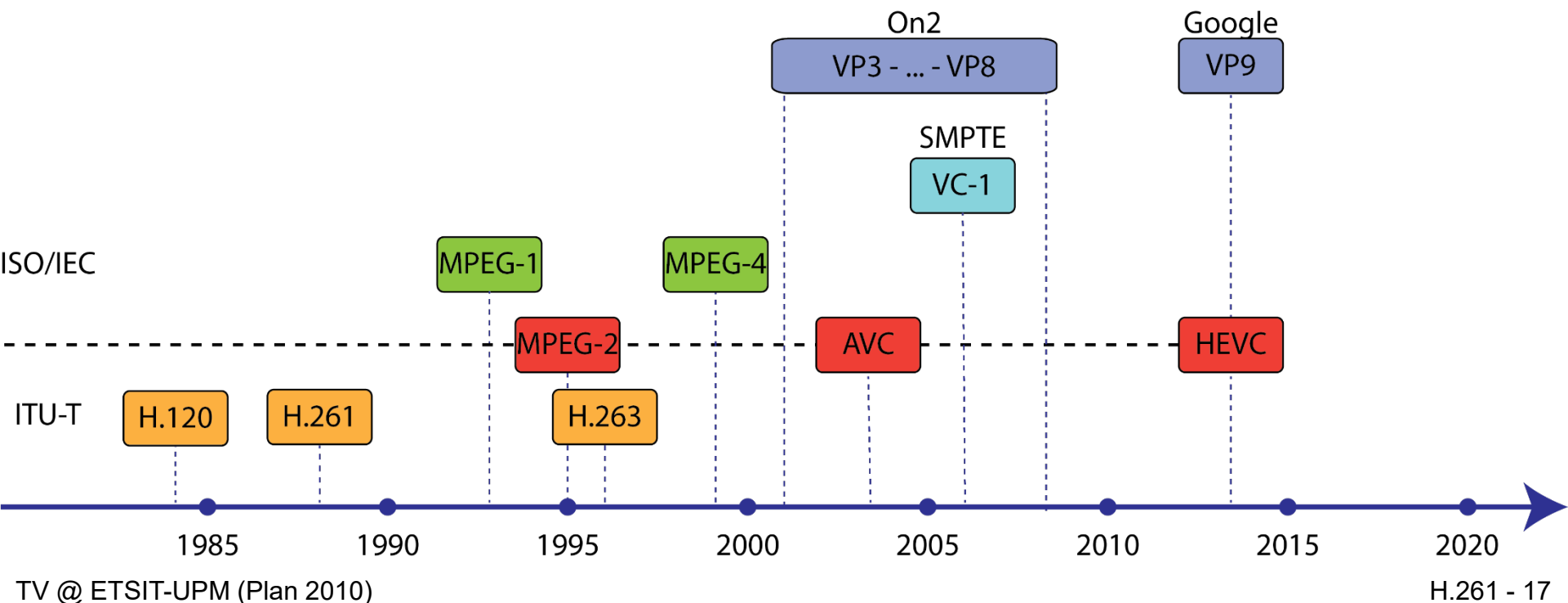
- On2 Technologies
- Buena calidad de vídeo y licencias baratas:
 - Usados por: Adobe, Apple, Skype, ...



1. Codificador híbrido

VP9 (2013):

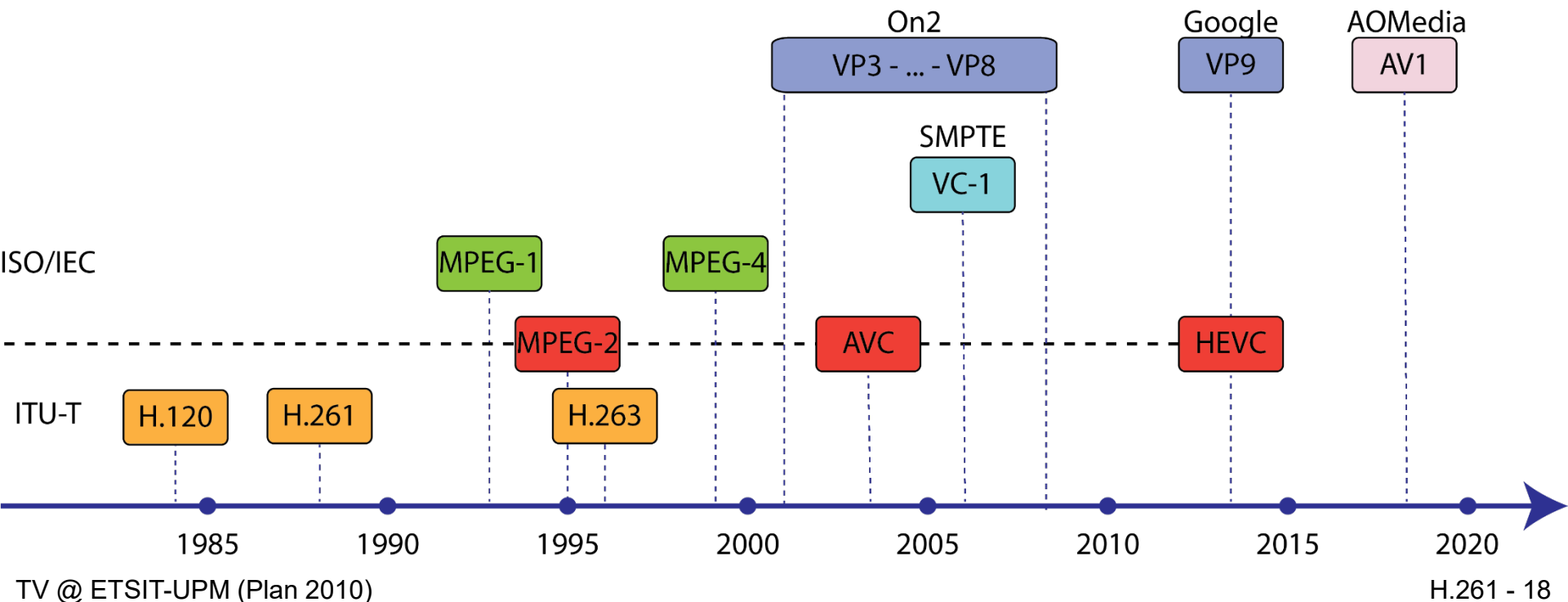
- Google (compró On2 Technologies en 2010).
- Alternativa a HEVC, pero libre de licencia (código abierto).



1. Codificador híbrido

AV1 (2018):

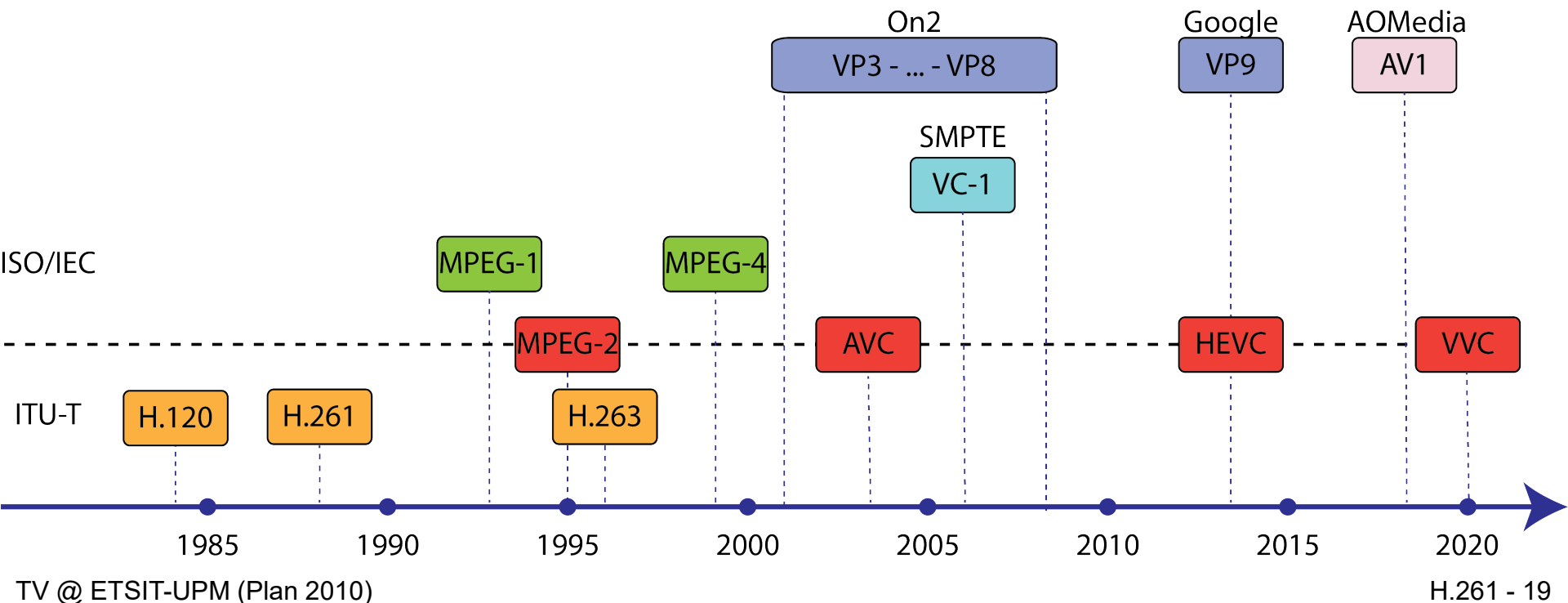
- Desarrollado por la AOMedia: *Alliance for Open Media*.
 - Fundada por: Amazon, Apple, Cisco, Google, IBM, Intel, Microsoft, Mozilla y Netflix.
- Código abierto.



1. Codificador híbrido

VVC – Versatile Video Coding (2020):

- JVET: *Joint Video Experts Team*.
 - ISO/IEC.
 - ITU-T.
- Sucesor de HEVC.



2 - H.261

1. Introducción
2. Formatos de vídeo
3. Codificador H.261
 1. Predicción
 2. Compensación del movimiento
 3. Representación de los vectores de movimiento
 4. Decisión inter/intra
 5. Actualización forzada
 6. Transformada DCT
 7. Cuantificación
 8. Ordenación de los coeficientes
 9. Direccionamiento de los bloques
 10. Tipos de macrobloques
 11. Filtro de bucle
4. Control del flujo binario
5. Control y ocultación de errores

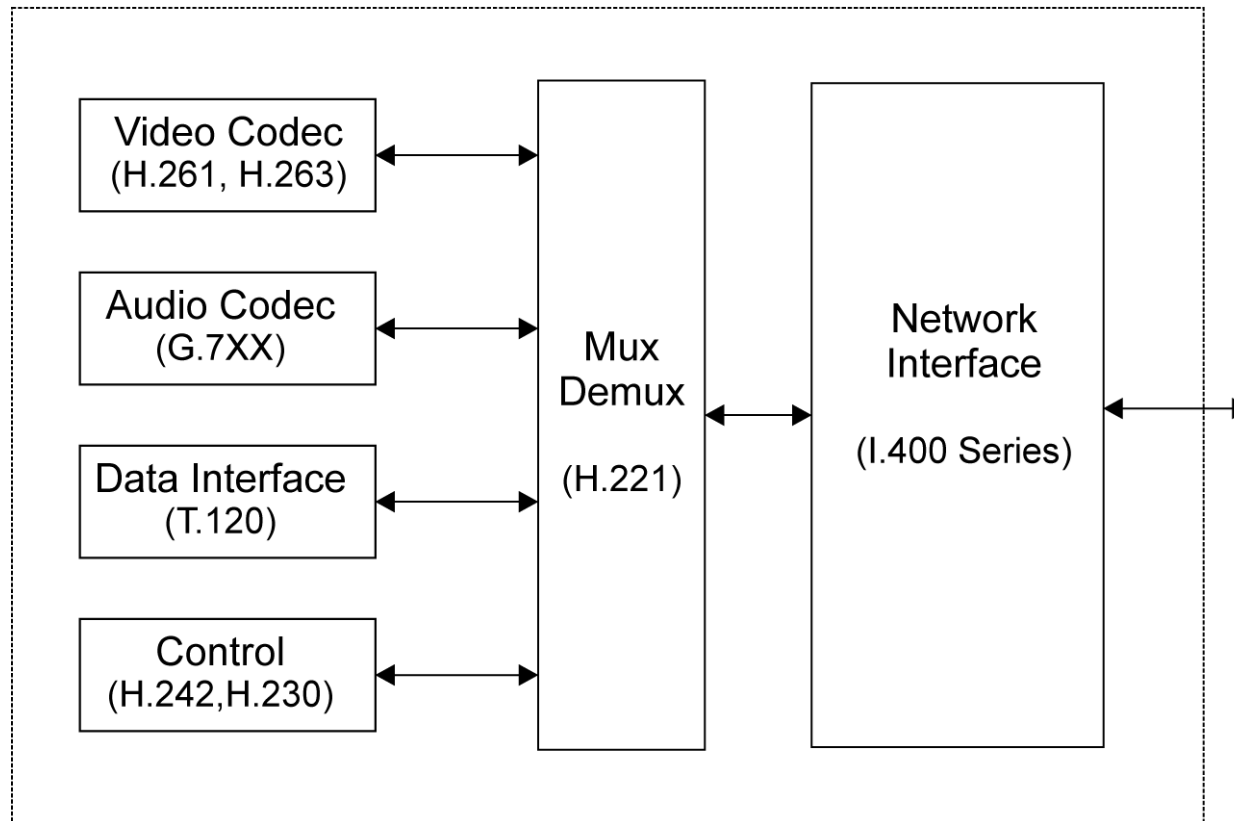
1. Introducción a H.261

- Fue desarrollado entre 1988 y 1993 (su 1ª versión “completa” fue aprobada en diciembre de 1990).
- Es el primero de los estándares de la familia H.26x.
- Define métodos de codificación y decodificación de vídeo para transmisión en redes de tipo RDSI.
- Permite la transmisión con tasas de bits múltiplo de 64 kb/s (hasta un máximo de 1920 kb/s = 30 canales de 64 kb/s).
- Algunas aplicaciones que motivaron la creación de este estándar fueron:
 - Videoconferencia.
 - Vigilancia / monitorización.
 - Telemedicina.

1. Introducción a H.261

- Es parte del grupo de estándares H.320 → Recomendación de la ITU-T para transmisión de vídeo, audio y datos por canales RDSI.

Diagrama de bloques del estándar H.320



1. Introducción a H.261

Etapas principales en el codificador:

- El codificador elabora predicciones que permiten obtener imágenes de error:
 - Así se explota la redundancia temporal de la secuencia de imágenes.
 - Las predicciones se obtienen a partir de la imagen previa reconstruida (también disponible en el decodificador).
 - En el caso más sencillo, el error es la diferencia entre la imagen actual y la anterior.
 - También es posible utilizar compensación de movimiento.
- Las imágenes de error son divididas en bloques sobre los que se aplica la DCT:
 - Así se explota la redundancia espacial de las imágenes.
- Los bloques transformados son cuantificados y codificados mediante códigos de longitud variable (VLC).

1. Introducción a H.261

- **Codificación entre cuadros o inter-frame:**

- Se utilizan predicciones para construir imágenes de error y explotar así la redundancia temporal.

- **Codificación en el cuadro o intra-frame:**

- No se codifica el error de predicción, sino la imagen.
- Sólo se explota la redundancia espacial → DCT.
- Se utiliza, por ejemplo:

?

1. Introducción a H.261

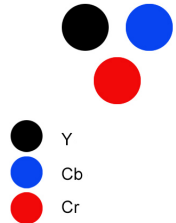
¿Qué impone el estándar?

- Únicamente detalla la estructura del flujo binario a transmitir y la estructura del decodificador.
- No impone ninguna estructura concreta del codificador.
- Se da libertad a los fabricantes para que utilicen codificadores con los módulos que consideren oportunos (por ejemplo, con o sin estimación y compensación de movimiento).

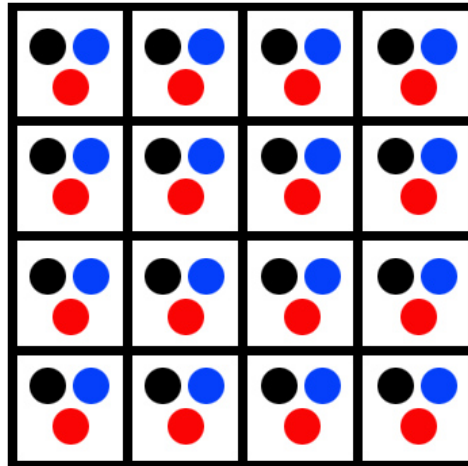
2. Formatos de vídeo

- *Frame rate*:
 - Velocidad mínima $\rightarrow 7,5$ fps
 - Velocidad máxima $\rightarrow 30$ fps
- Soporta dos resoluciones:
 - CIF $\rightarrow 288 \times 352$ píxeles.
 - QCIF $\rightarrow 144 \times 176$ píxeles.
- Componentes:
 - En vez de utilizar el espacio de color RGB, se usa el YCrCb para poder tratar de distinta forma la información de luminancia y crominancia.
- Muestreo (4:2:0):
 - Muestreo completo de Y.
 - Submuestreo en horizontal y vertical (2:1) de Cr y Cb.

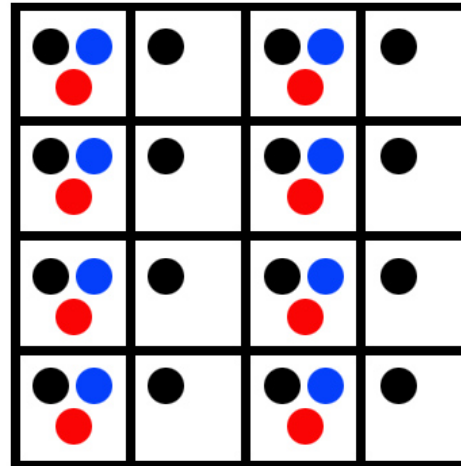
2. Formatos de vídeo



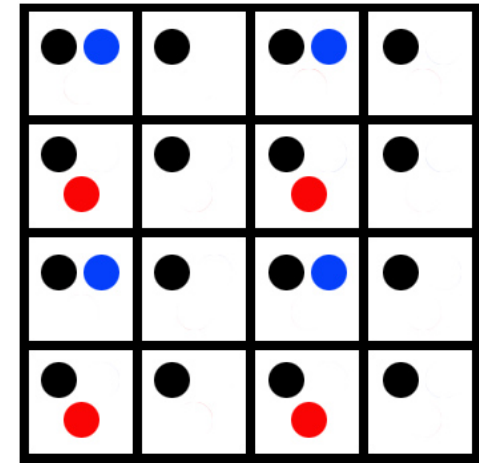
4:4:4



4:2:2



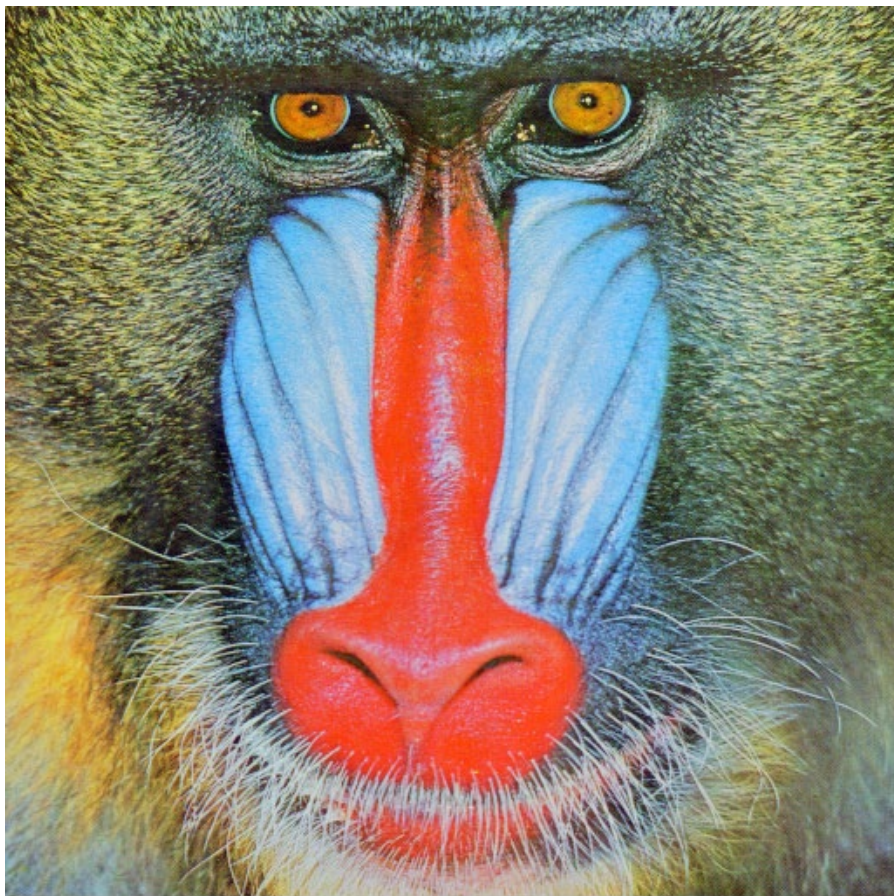
4:2:0



2. Formatos de vídeo

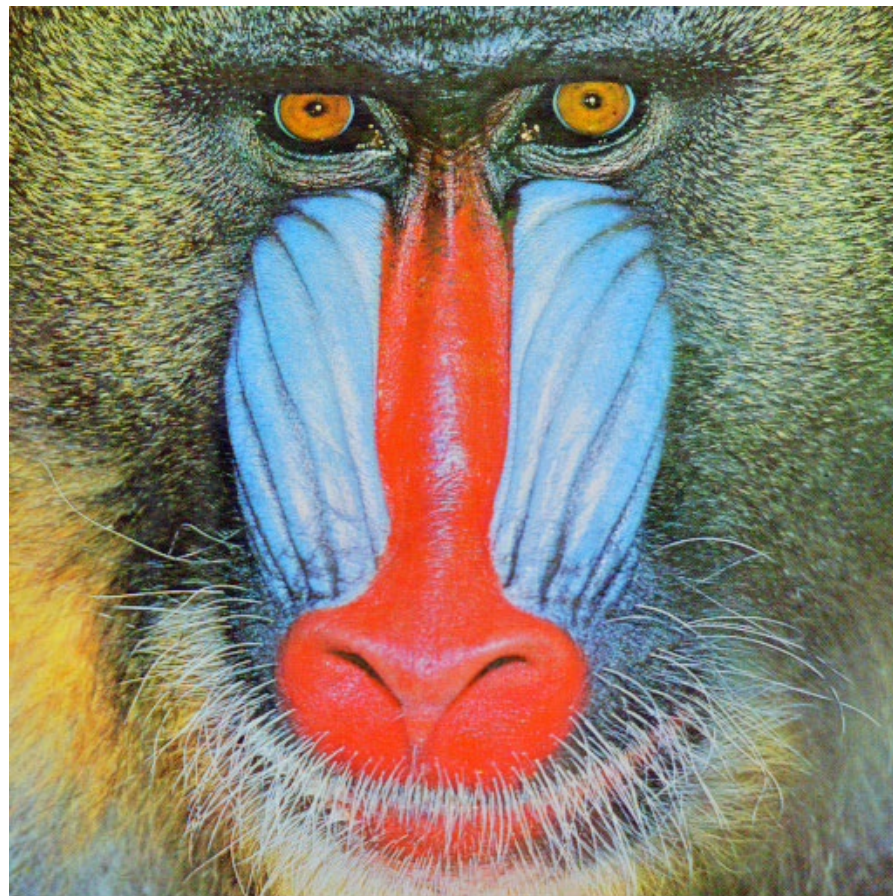
4:4:4

**YCrCb
24 bpp
6.29 Mb**



4:2:0

**YCrCb
12 bpp
3.15 Mb**



2. Formatos de vídeo

- **Estructura jerárquica de 4 capas:**

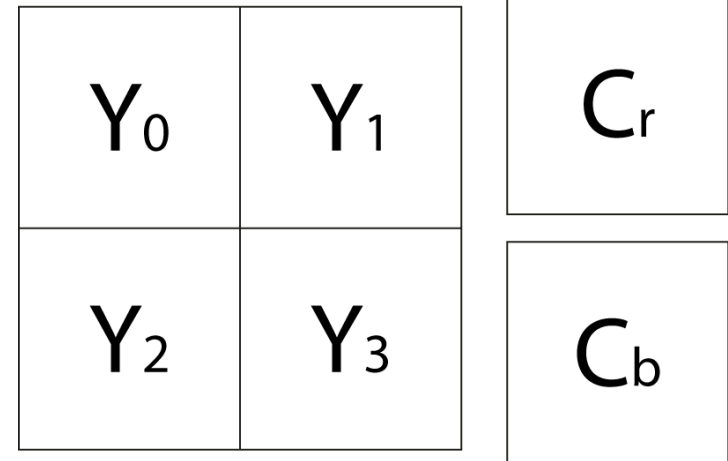
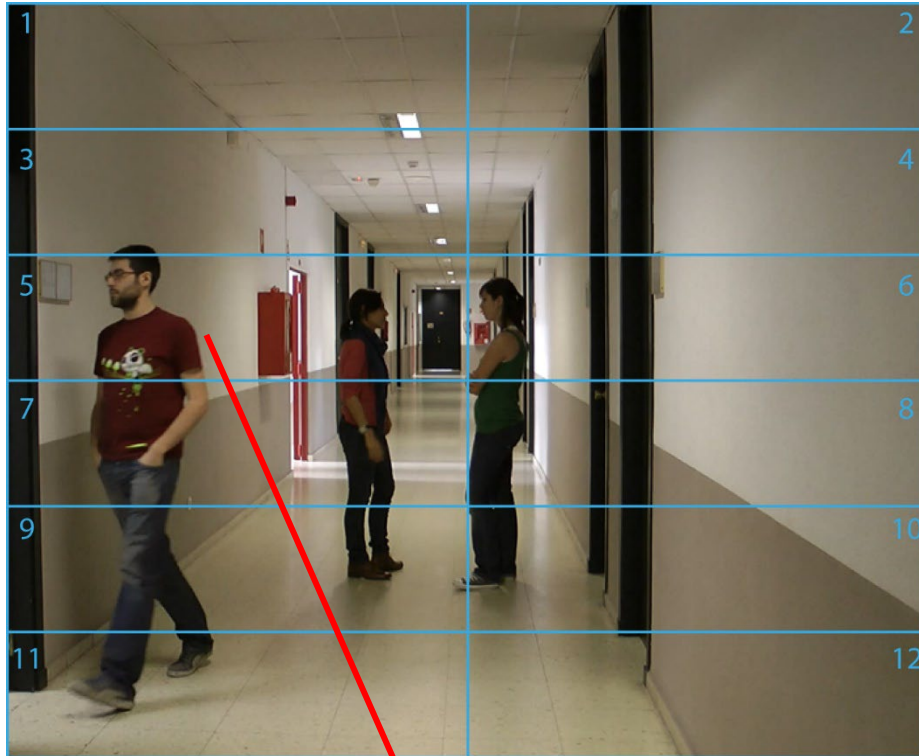
- CIF:

1. Imagen.
2. Grupo de bloques (GOB) → Una imagen se compone de 12 GOBs (6 filas y 2 columnas).
3. Macrobloque (MB) → Cada GOB contiene 33 MBs dispuestos en 3 filas y 11 columnas.
 - Cada MB consta de 16×16 píxeles.
4. Bloque → Cada MB se compone de 6 bloques.
 - 4 de 8×8 muestras de Y → 16×16 muestras contiguas.
 - 1 de 8×8 muestras de Cr.
 - 1 de 8×8 muestras de Cb.

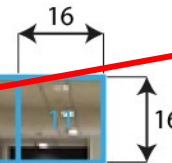
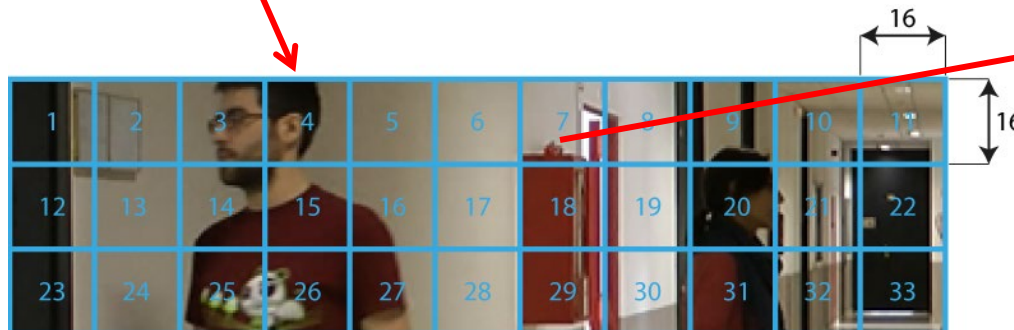
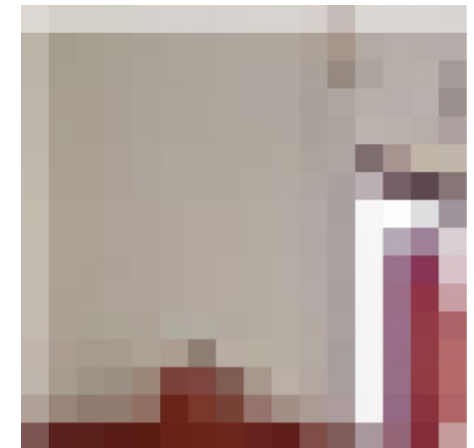
- QCIF:

1. Imagen.
2. Grupo de bloques (GOB) → Una imagen se compone de 3 GOBs (3 filas y 1 columna).
3. Macrobloque (MB) → Igual que CIF.
4. Bloque → Igual que CIF.

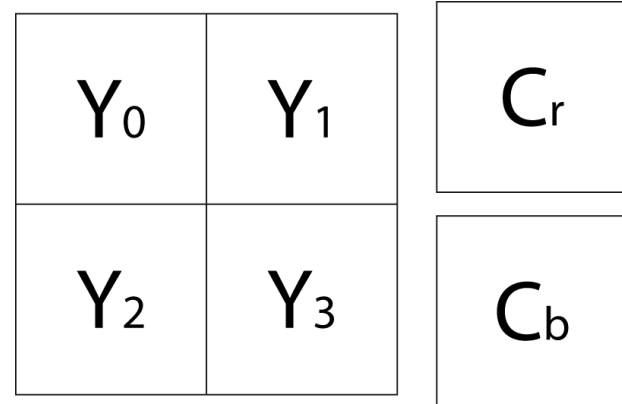
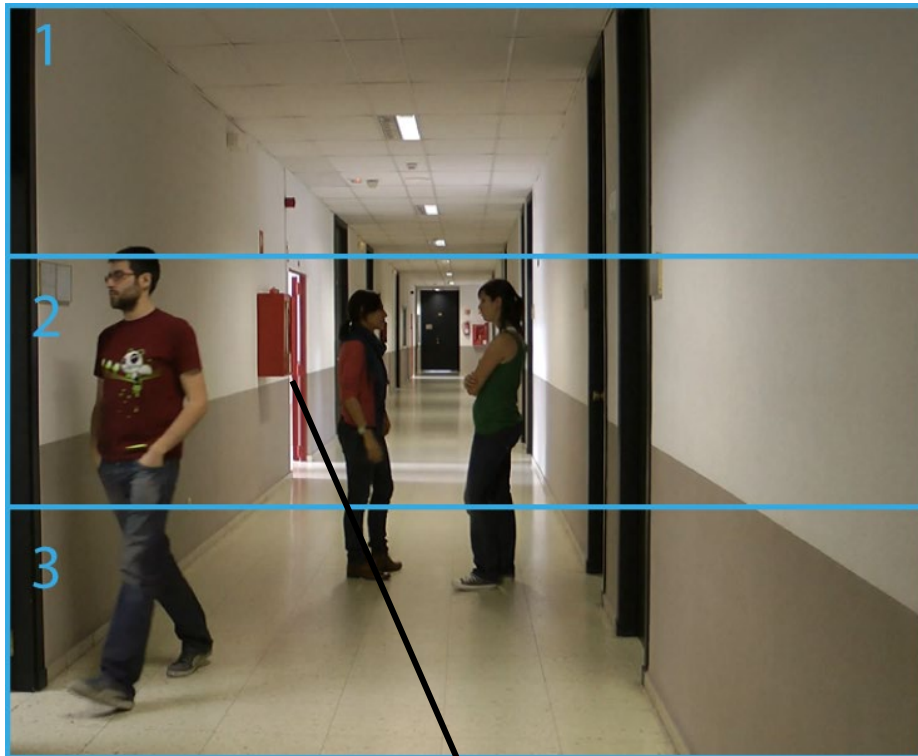
2. Formatos de vídeo



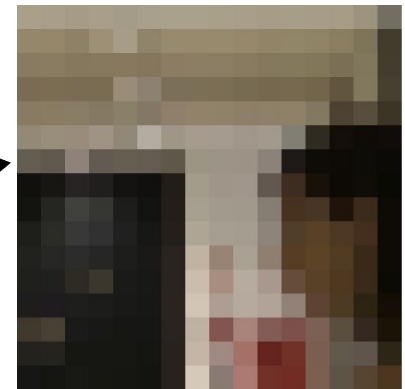
CIF



2. Formatos de vídeo



QCIF



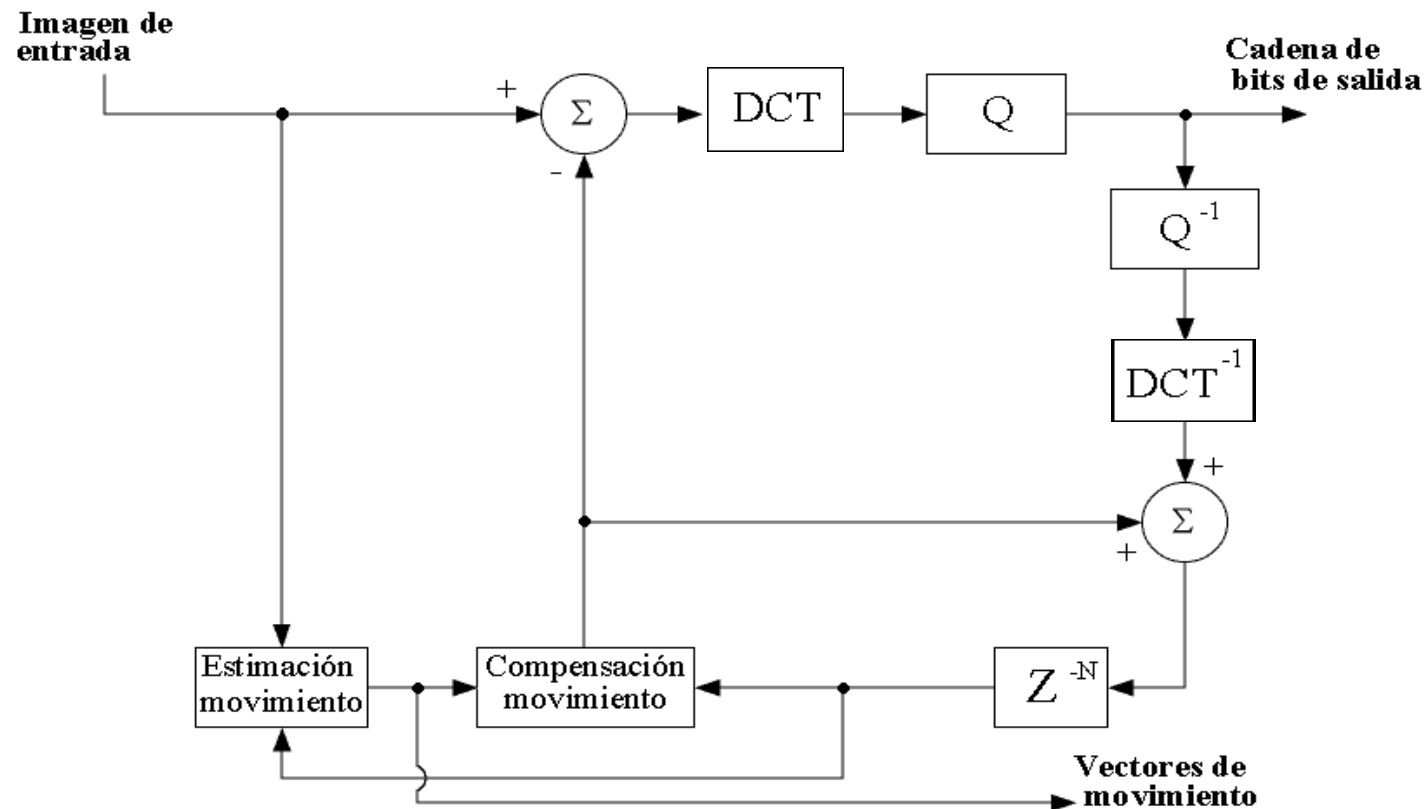
2. Formatos de vídeo

- **¿Por qué esta estructuración en MB y GOB?**

- Para utilizar el mismo tipo de codificación (ínter o intra) en bloques de luminancia y crominancia de la misma área de la imagen.
- Para utilizar un único vector de movimiento por MB (el mismo para luminancia y crominancia).
- Para facilitar la recuperación de la sincronización → Se introduce información de sincronización al comienzo de cada GOB, por lo que en caso de error sólo una pequeña parte de la imagen es irrecuperable.
- Para evitar un exceso de información adicional a transmitir (información de cuantificación, vectores de movimiento, referencias temporales...).

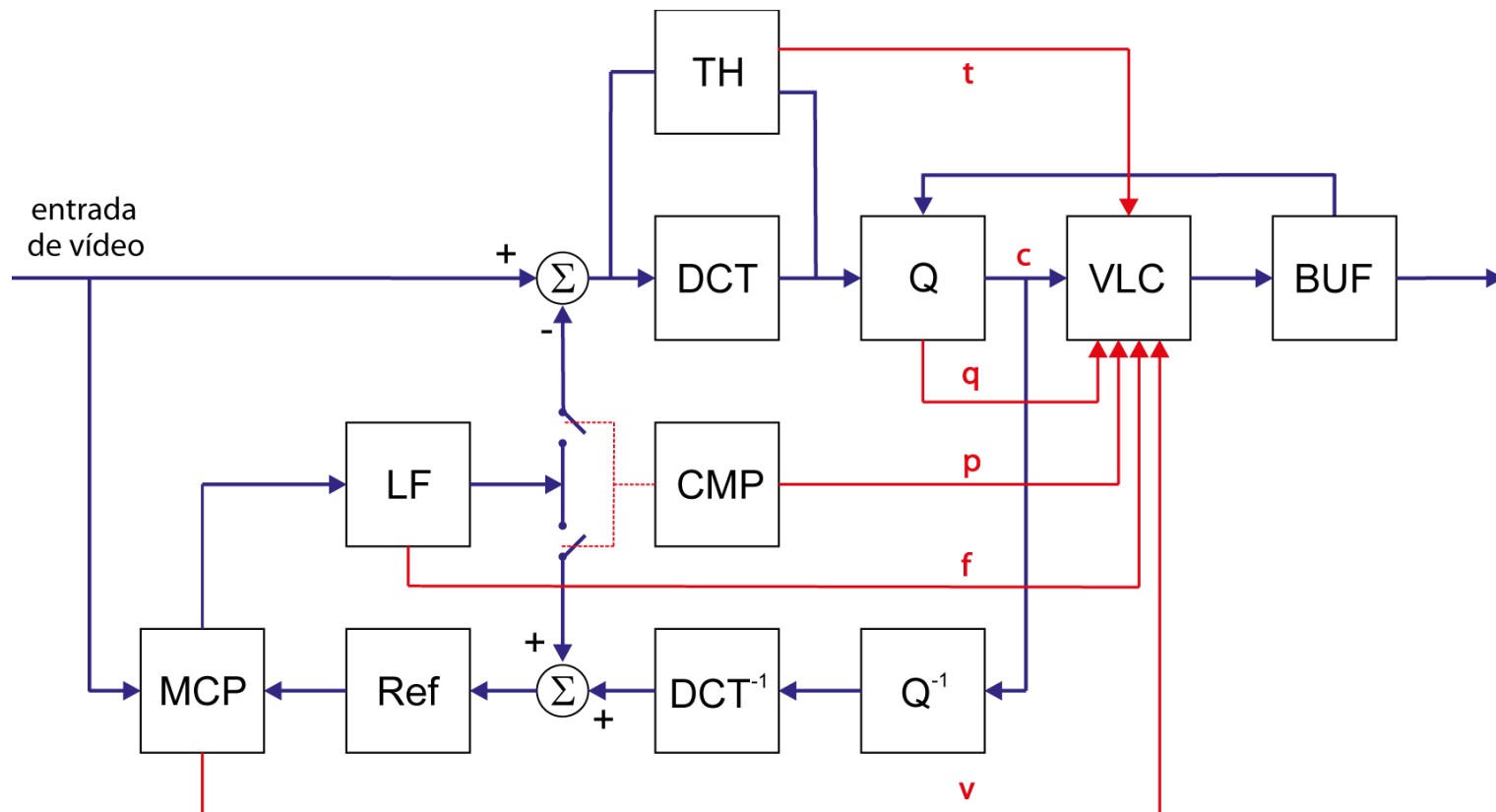
3. Codificador H.261

- Codificador híbrido predictivo-transformacional:



3. Codificador H.261

- Esquema simplificado del codificador H.261:
 - El elemento básico sobre el que se opera es el MB.



3. Codificador H.261

- Esquema simplificado del codificador H.261:
 - q → Escalón de cuantificación.
 - c → Coeficientes cuantificados.
 - CMP → Comparador (para decidir si se usa codificación ínter o intra).
 - p → Indicador de codificación ínter o intra.
 - Ref → Imagen de referencia.
 - MCP → Compensación de movimiento.
 - v → Vectores de movimiento.
 - LF → Filtro en el bucle.
 - f → Activación/desactivación del filtro.
 - TH → Extensión del umbral.
 - t → Transmitido / no transmitido.
 - VLC → Codificación de longitud variable.
 - BUF → *Buffer*.

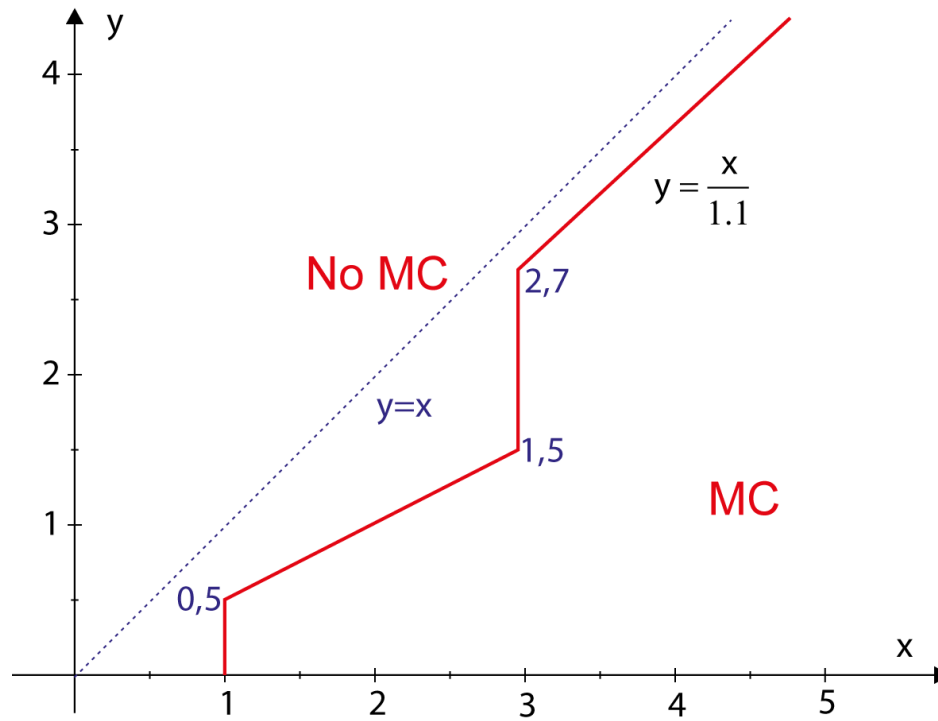
3.1. Predicción

- Cada MB de la imagen de entrada se compara con una predicción obtenida del cuadro anterior → predicción inter-cuadro:

- De esta forma se explota la redundancia temporal de la secuencia.
- Sin estimación y compensación → Diferencia con el MB en la misma posición.
- Con estimación y compensación:
 - Es necesario transmitir un vector asociado a cada MB.
 - Cada una de las componentes (x , y) del vector está limitada a ± 15 píxeles.
 - Se comparan únicamente muestras de luminancia.
 - En el caso de la crominancia, se utiliza el mismo vector dividido por 2 (y truncado hacia 0).
 - Los vectores de movimiento se transmiten diferencialmente → diferencias (por componentes) entre cada vector y el del MB anterior.
 - Las diferencias son codificadas utilizando VLC.

3.2. Compensación del movimiento

- Sólo se aplica MC (*Motion Compensation*) si se reduce notablemente el error de predicción.
- Para decidir si se usa o no, se compara la magnitud del error con y sin MC.



Eje x (sin MC): $x = \frac{|BD|}{256}$

Eje y (con MC): $y = \frac{|DBD|}{256}$

$[D]BD = [Displaced] Block Difference$

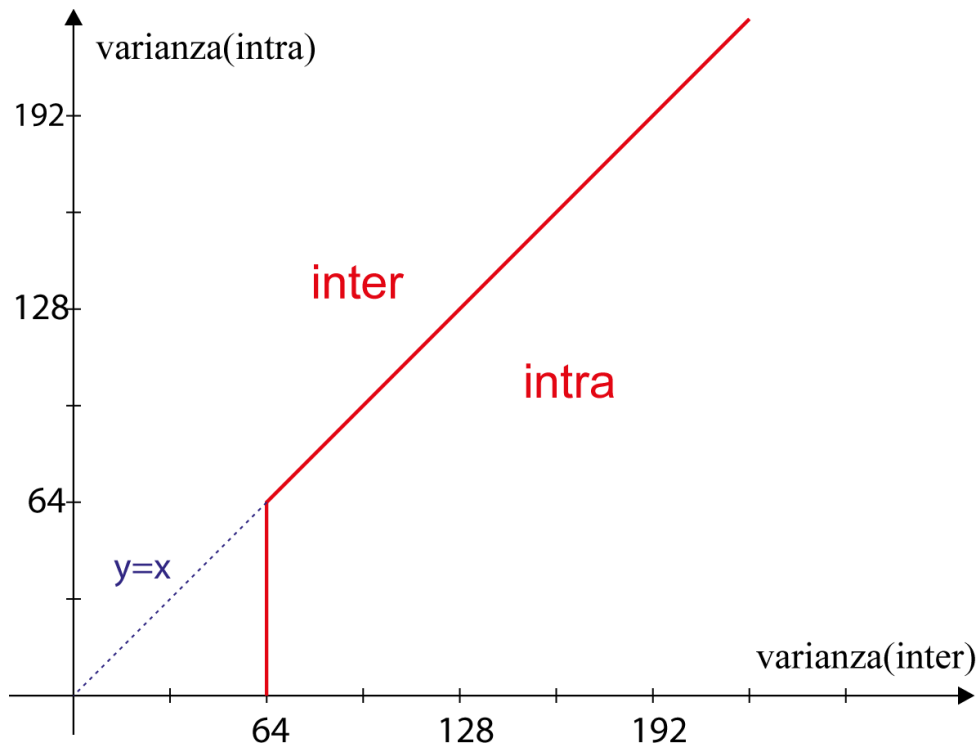
Si la ganancia con MC no es significativa (en general, $y < x/1,1$), no se aplica → Así se evita el envío ineficiente de vectores de movimiento.

3.3. Representación de los vectores

- Los vectores de MBs adyacentes suelen ser parecidos → Es más eficiente codificarlos diferencialmente.
- Casos en los que no se usa predicción entre vectores (se codifica directamente el vector):
 - Primer MB dentro de un GOB.
 - El MB previo no fue codificado (luego veremos cuándo sucede esto).
 - El MB previo se codificó como intra (id.).
 - El MB previo tiene asociado un vector de movimiento nulo.
- La información de los vectores se codifica utilizando un VLC.
- Pros y contras de la codificación diferencial:
 - ✓ En las secuencias de videoconferencia los objetos móviles (busto parlante) se mueven de forma muy rígida → La codificación diferencial mejora mucho la eficiencia.
 - ✗ Los errores en transmisión se propagan, aunque sólo hasta que empiece el siguiente GOB.

3.4. Decisión íter/intra

- Se utiliza el tipo de codificación que dé lugar a menor varianza.



- Para valores grandes de varianza, no se establece ninguna preferencia entre los dos tipos de codificación.
- Para valores bajos de varianza se prefiere el modo inter:
 - En el modo intra, los coeficientes de continua (DC) se codifican con mayor precisión (8 bits) → Esto aumenta la tasa binaria.

3.5. Actualización forzada

- La codificación intra es más robusta frente a errores de transmisión que la ínter.
 - Si la decisión intra se toma con una frecuencia demasiado baja, la calidad puede reducirse notablemente.



- La norma recomienda que cada MB se actualice con una codificación tipo intra al menos una vez cada 132 imágenes:
 - En resolución CIF (396 MB) → En término medio, 3 MB por imagen utilizan codificación intra.
 - Con una velocidad de 30 imágenes por segundo → El tiempo máximo durante el que se puede visualizar un error es de:

$$\frac{132}{30} = 4,4 \text{ segundos}$$

- Cuanto menor es el *framerate*, mayor es este tiempo máximo.

3.6. Transformada DCT (1)

- Se aplica sobre cada uno de los 6 bloques de 8 x 8 muestras asociados a cada MB.
- Su objetivo es explotar la redundancia espacial de los datos.
- Es una transformada sin pérdidas inherentes (con la IDCT se obtienen los valores iniciales de los bloques) pero...
 - La posterior cuantificación introduce un error que hace que la IDCT no obtenga exactamente los valores iniciales.

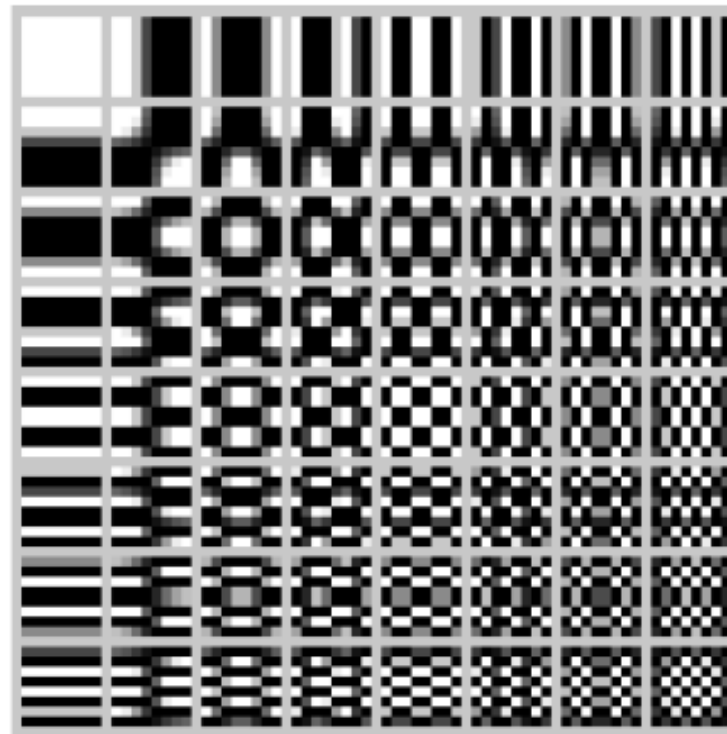
$$z(i, j) = \frac{1}{4} C_i C_j \sum_{n=0}^7 \sum_{m=0}^7 x(n, m) \cos \frac{(2n+1)i\pi}{16} \cos \frac{(2m+1)j\pi}{16} \quad i, j = 0, 1, \dots, 7$$

$$x(n, m) = \frac{1}{4} \sum_{i=0}^7 \sum_{j=0}^7 C_i C_j z(i, j) \cos \frac{(2n+1)i\pi}{16} \cos \frac{(2m+1)j\pi}{16} \quad n, m = 0, 1, \dots, 7$$

$$C_k = \begin{cases} \frac{1}{\sqrt{2}} & \text{si } k = 0 \\ 1 & \text{si } k \neq 0 \end{cases}$$

3.6. Transformada DCT (2)

- La DCT descompone cada bloque de entrada en 64 funciones base (ondas) con distintas frecuencias espaciales.



Con la transformada DCT se buscan los pesos (coeficientes) de cada una de estas 64 funciones base, para que sumadas y escaladas por dichos pesos reconstituyan el bloque original.

3.6. Transformada DCT (3)

- Ejemplo de transformación:

Bloque original →

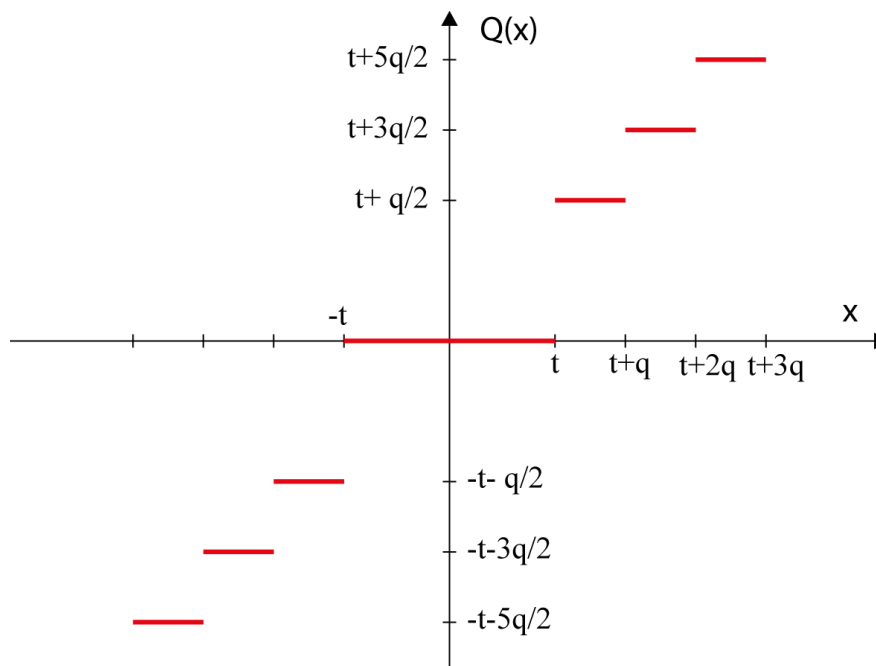
$$x = \begin{bmatrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 159 & 160 & 161 & 162 & 162 & 155 & 155 & 155 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 163 & 162 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 \end{bmatrix}$$

Bloque transformado →

$$z = \begin{bmatrix} 1260 & -1 & -12 & -5 & 2 & -2 & -3 & 1 \\ -23 & -17 & -6 & -3 & -3 & 0 & 0 & -1 \\ -11 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -7 & -2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 2 & 0 & -1 & 1 & 1 \\ 2 & 0 & 2 & 0 & -1 & 1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 2 & 1 & -1 \\ -3 & 2 & -4 & -2 & 2 & 1 & -1 & 0 \end{bmatrix}$$

3.7. Cuantificación (1)

- Se aplica sobre bloques de 8×8 muestras resultantes de la transformación DCT.
- Se aplica distinta cuantificación dependiendo de:
 - El tipo de codificación aplicada $\rightarrow$ ínter o intra.
 - El orden del coeficiente dentro del bloque.
- El cuantificador utilizado se denomina UTQ (*Uniform Threshold Quantizer*).



- Se define mediante 2 parámetros:
 - El umbral t (que puede ser cero) $\rightarrow$ Define lo que se conoce como “zona muerta” (zona en la que el valor de reconstrucción es 0).
 - El escalón de cuantificación q $\rightarrow$ El valor de reconstrucción dentro de cada intervalo se sitúa en el centro del intervalo.

3.7. Cuantificación (2)

- Cuando no hay zona muerta ($t = 0$) el cuantificador se denomina UTQ (*Uniform Threshold Quantizer*).
 - Se utiliza para los coeficientes DC codificados en modo intra.
 - Es equivalente a utilizar un cuantificador uniforme.
 - En este caso, $q = 8 \rightarrow 8$ bits.
- Cuando hay zona muerta ($t > 0$) se habla de UTQ-DZ (*Uniform Threshold Quantizer-Dead Zone*).
 - Se utiliza para:
 - Los coeficientes de alterna de los bloques codificados en modo intra.
 - Todos los coeficientes de los bloques codificados en modo inter.
 - Cuanto mayor es t , mayor número de coeficientes se codifican con un 0.
 - En este caso, $q \in \{2, 4, \dots, 62\}$.
- Los valores de t y q determinan el grado de ocupación del buffer.
 - Cuanto mayor es $t \rightarrow$ Más coeficientes cuantificados con un 0.
 - Cuanto mayor es $q \rightarrow$ Menos bits necesarios para cuantificar los coeficientes más significativos.

3.7. Cuantificación (3)

- Ejemplo:

- Bloque codificado en **modo inter.**
- $t = 8$ y $q = 16$.

102	-25	55	-2	21	-9	0	1
39	-6	-7	-35	4	5	1	-1
15	13	16	3	0	0	-5	2
27	7	-2	-2	1	5	0	0
-23	0	-1	-1	3	1	0	0
2	-1	-1	2	2	3	3	-2
6	-2	1	-4	-2	1	1	0
0	0	-5	-6	5	3	-1	1

Coeficientes

Coeficientes cuantificados

3.7. Cuantificación (4)

- Ejemplo:

- Los coeficientes comprendidos dentro de la zona muerta se cuantifican con un cero ($t = 8$).

102	-25	55	-2	21	-9	0	1
39	-6	-7	-35	4	5	1	-1
15	13	16	3	0	0	-5	2
27	7	-2	-2	1	5	0	0
-23	0	-1	-1	3	1	0	0
2	-1	-1	2	2	3	3	-2
6	-2	1	-4	-2	1	1	0
0	0	-5	-6	5	3	-1	1

Coeficientes

			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	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Coeficientes cuantificados

3.7. Cuantificación (5)

• Ejemplo:

○ El rango del resto de intervalos de cuantificación (de la parte positiva) será:

- IC1: Rango $\rightarrow (8, 24]$; Etiqueta $\rightarrow 1$; Valor de reconstrucción $\rightarrow 16$
- IC2: Rango $\rightarrow (24, 40]$; Etiqueta $\rightarrow 2$; Valor de reconstrucción $\rightarrow 32$
- ...

102	-25	55		21	-9		
39			-35				
15	13	16					
27							
-23							

Coeficientes

			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	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Coeficientes cuantificados

3.7. Cuantificación (6)

- Ejemplo:

- Para el coeficiente de valor 55 → ¿IC?

$$\left\lceil \frac{55 - 8}{16} \right\rceil = 3$$

102	-25	55		21	-9		
39			-35				
15	13	16					
27							
-23							

Coeficientes

		3	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	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Coeficientes cuantificados

3.7. Cuantificación (7)

- Ejemplo:

- Para el coeficiente de continua (102) → ¿IC?

$$\left\lceil \frac{102 - 8}{16} \right\rceil = 6$$

102	-25			21	-9		
39			-35				
15	13	16					
27							
-23							

Coeficientes

6		3	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	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Coeficientes cuantificados

3.7. Cuantificación (8)

- Ejemplo:

- Para el resto de los coeficientes se procede de igual modo.

	-25			21	-9		
39			-35				
15	13	16					
27							
-23							

Coeficientes

6	-2	3	0	1	-1	0	0
2	0	0	-2	0	0	0	0
1	1	1	0	0	0	0	0
2	0	0	0	0	0	0	0
-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	0

Coeficientes cuantificados

3.7. Cuantificación (9)

- El proceso completo se puede resumir en la aplicación de la siguiente expresión:

$$i(x) = \begin{cases} 0 & |x| \leq t \\ \text{sign}(x) \left\lceil \frac{|x| - t}{q} \right\rceil & |x| > t \end{cases}$$

102	-25	55	-2	21	-9	0	1
39	-6	-7	-35	4	5	1	-1
15	13	16	3	0	0	-5	2
27	7	-2	-2	1	5	0	0
-23	0	-1	-1	3	1	0	0
2	-1	-1	2	2	3	3	-2
6	-2	1	-4	-2	1	1	0
0	0	-5	-6	5	3	-1	1

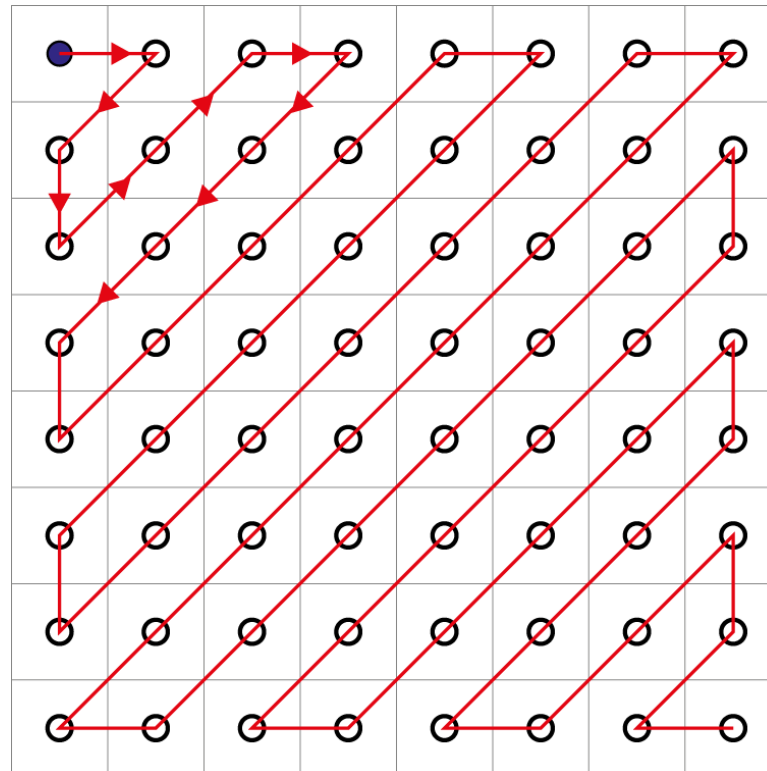
Coeficientes

6	-2	3	0	1	-1	0	0
2	0	0	-2	0	0	0	0
1	1	1	0	0	0	0	0
2	0	0	0	0	0	0	0
-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	0

Coeficientes cuantificados

3.8. Ordenación de los coeficientes (1)

- Normalmente, la mayor parte de la energía de los bloques de muestras se concentra en los coeficientes de baja frecuencia.
 - Para aprovechar esta característica, los coeficientes de los bloques se leen en zig-zag (hasta llegar al último no nulo).

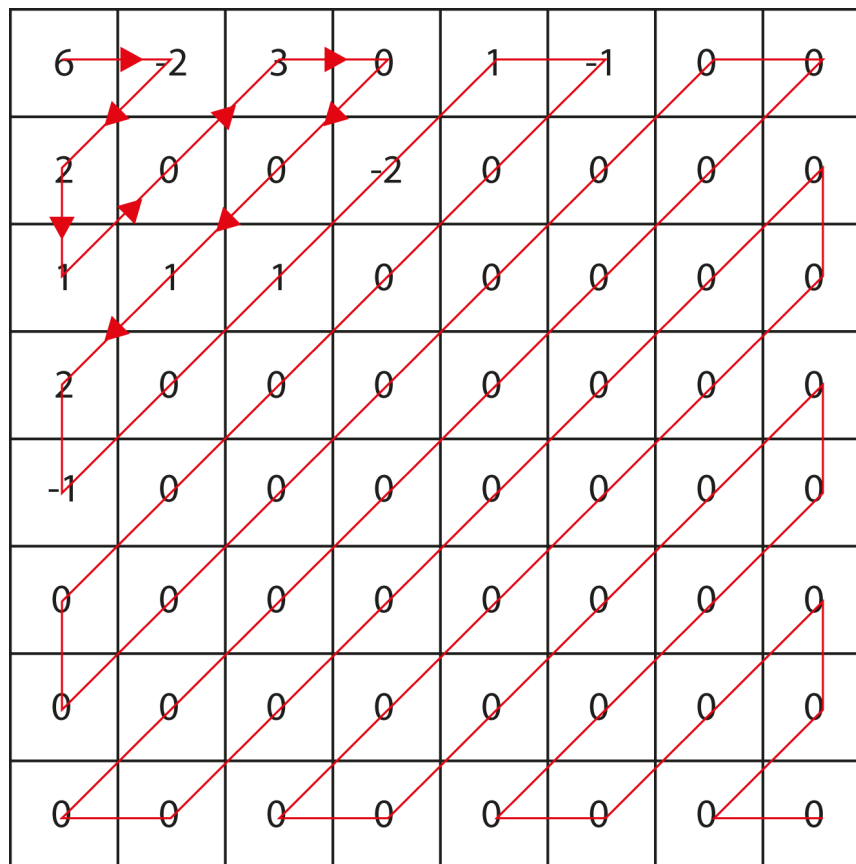


3.8. Ordenación de los coeficientes (2)

- Para mejorar la eficiencia de la codificación se utiliza un código VLC-2D.
 - Fase 1: Según se van recorriendo los coeficientes y, cada vez que se encuentra uno distinto de cero, se forma una pareja (nz, q) .
 - $nz \rightarrow$ Número de ceros que preceden al coeficiente (hasta el último coeficiente codificado).
 - $q \rightarrow$ Coeficiente cuantificado.
 - Cuando se han recorrido todos los coeficientes no nulos se genera el símbolo EOB (End Of Block) para indicarlo.
 - Fase 2: Las parejas (nz, q) se codifican utilizando un código Huffman modificado.
 - El código Huffman original no es práctico debido al elevado número de eventos posibles.
 - Ejemplo: Con 255 IC y teniendo en cuenta que el nz toma valores entre 0 y 63, se obtendría un total de $255 \times 64 + 1 = 16321$ eventos (la mayoría muy improbables).
 - En el código Huffman modificado se agrupan todos los eventos de menor probabilidad en un solo símbolo (ESC).
 - Los eventos menos probables se codifican con ESC + nz y q codificados con un código de longitud fija.
 - Coeficiente DC: Se utiliza un cuantificador sin zona muerta y se codifica con 8 bits.

3.8. Ordenación de los coeficientes (3)

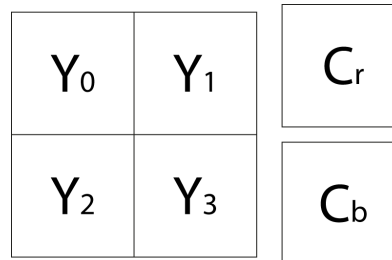
- Ejemplo:



(0,6) (0,-2) (0,2) (0,1) (1,3) (2,1) (0,2) (0,-1) (1,1) (0,-2) (0,1) (0,-1) EOB

3.9. Direccionamiento de los bloques

- Sabemos que un MB se compone de 4 bloques de luminancia y dos de crominancia.



- Tras cuantificarse los 6 bloques de un MB, se genera una cadena de 6 bits (uno por bloque) que sigue el patrón $Y_0, Y_1, Y_2, Y_3, C_r, C_b$.
 - Un '1' indica que el bloque tiene coeficientes no nulos y un '0' que todos los coeficientes son nulos.
 - De las 64 posibles combinaciones sólo se utilizan 63 → Cuando los 6 bloques son nulos el MB se trata de forma distinta (lo veremos después).
 - El código generado se codifica utilizando VLC.
 - En el caso de MB tipo intra se supone que todos los bloques contendrán información → Todos los bloques serán codificados siempre → No es necesario transmitir el patrón.

3.10. Tipos de macrobloques. Direccionamiento

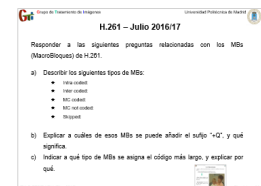
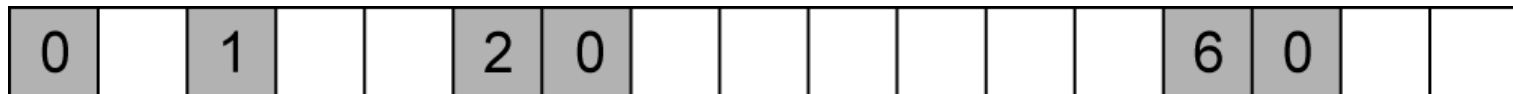
- Existen 8 tipos de MB:
 1. Inter coded: Predicción temporal sin utilizar compensación de movimiento.
 2. MC coded: Predicción temporal con compensación de movimiento, con un error de predicción lo suficientemente grande como para requerir su codificación con la DCT.
 3. MC not coded: Predicción temporal con compensación de movimiento, con error de predicción lo suficientemente pequeño como para no requerir ser codificado ($Y_0Y_1Y_2Y_3C_rC_b = 000000$).
 - Sólo hay que transmitir el vector de movimiento.
 4. Intra coded: Codificación en el cuadro (sin predicción).
 5. Skipped: Con vector de movimiento nulo y con $Y_0Y_1Y_2Y_3C_rC_b = 000000$.
 - No es necesario transmitir nada. El decodificador simplemente copia el MB de la imagen anterior.

Para optimizar el uso del *buffer*, al comienzo de cada GOB se pueden cambiar los parámetros del cuantificador, por lo que dichos parámetros deben ser transmitidos → Aparecen 3 nuevos tipos de MB:

6. Inter coded + Q
7. MC coded + Q
8. Intra coded + Q

3.10. Tipos de macrobloques. Direccionamiento

- La información relativa al tipo de MB se transmite utilizando VLC.
 - Si el MB no se transmite (tipos 3 y 5), no se envía ningún código.
 - El código más largo se le asigna al tipo Intra + Q → Por ser el menos probable (debido al tipo de aplicaciones en las que normalmente se usa H.261).
- Es muy probable que un número muy elevado de MBs no se transmita:
 - Hay que indicar la posición en el GOB de los que se reciben.
 - Se indica el número de MBs entre el actual y el anterior MB recibido, o entre el actual y el comienzo del GOB (cuando es el primero del GOB en recibirse).
 - Una vez más, esta longitud se codifica mediante VLC.



3.11. Filtro de bucle

- Cuando se utilizan tasas binarias bajas, el escalón de cuantificación (q) suele ser grande:
 - Muchos coeficientes de la DCT son cuantificados con un cero.
 - Sólo se transmite el coeficiente de continua y algunos de alterna.



- Algunos bloques empiezan a ser visibles (zonas con contenido de alta frecuencia).
- Ruido de mosquito → Si la posición de estos bloques cambia de una imagen a otra.



3.11. Filtro de bucle

- Se añade un filtrado paso bajo tras la compensación de movimiento con el fin de mejorar la predicción:

- ✓ Se reduce el efecto de bloques.
- ✗ Aparece cierto emborronamiento (*blurring*).



Sin filtrado



Con filtrado

3.11. Filtro de bucle

- Para evitar un emborronamiento innecesario, sólo se aplica sobre los bloques (8×8) asociados a MB con movimiento.

- Su respuesta al impulso es:

$$h(x, y) = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

9	3							
3	1							
			1	2	1			
			2	4	2			
			1	2	1			
			1	2	1			
			3	6	3			

- En las esquinas de los bloques se pueden usar filtros distintos.
- Sólo se activa para velocidades inferiores a $6 \times 64 \text{ kb/s} = 386 \text{ kb/s}$.
 - A mayor velocidad no se aprecia ninguna mejora subjetiva tras su aplicación.
 - Por este motivo, en codificadores más modernos (p.ej. MPEG-1), donde la velocidad es muy superior, no se incluye este filtrado.

4. Control del flujo binario

- La tasa binaria resultante de la codificación varía mucho dependiendo de distintos factores:
 - Movimiento.
 - Amplitud de los datos.
- Sin embargo, la capacidad de transmisión del canal es constante y hay que tratar de explotarla al máximo:
 - Para lograrlo se sitúa un *buffer* a la salida del codificador.
 - Hay que controlar la cantidad de información que llega al *buffer*, para evitar que se llene (*overflow*) o que se vacíe (*underflow*).
 - Cuanto más grande es el *buffer*:
 - ✓ Menor probabilidad de desbordamiento.
 - ✗ Mayor retardo.

4. Control del flujo binario

- Control del buffer:

- Se ajusta el paso del cuantificador (q) en función de la ocupación del *buffer*.
- Por ejemplo, si la velocidad de transmisión objetivo es $p \times 64$ kb/s, el paso del cuantificador suele establecerse como:

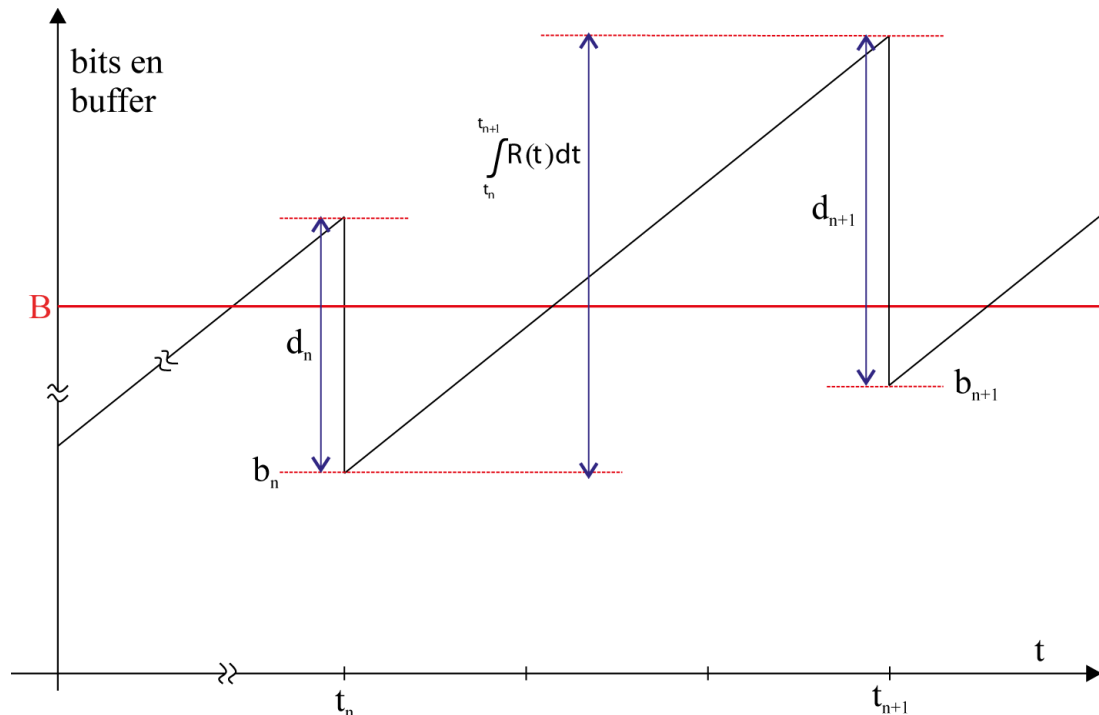
$$q = 2 \left\lfloor \frac{\text{bits en el buffer}}{200p} \right\rfloor$$

- Si la ocupación del *buffer* alcanza un nivel crítico, sólo se transmite información correspondiente al GOB o a la imagen, pero no de los MB y bloques.
- Si el *buffer* se está quedando vacío, se transmiten bits de relleno para que el decodificador no pierda la sincronización.
- La norma define un modelo de referencia para el *buffer* al que deben ajustarse todos los decodificadores.

4. Control del flujo binario

• Modelo de referencia para el *buffer*:

- Inicialmente está vacío y se examina cada Δt segundos $\rightarrow \Delta t = \frac{1}{29,97} \approx 33\text{ms}$
- Si en el *buffer* hay datos correspondientes a una imagen completa:
 - Se retiran los datos correspondientes a la imagen más antigua.
 - Una vez retirados los datos, la norma exige una ocupación máxima del *buffer* $\rightarrow B = \frac{4R_{\max}}{29,97}$



4. Control del flujo binario

- El mayor defecto del mecanismo de control del *buffer* es que los bits disponibles pueden repartirse de forma poco justa en la imagen:
 - Las partes más activas de las imágenes (por ejemplo, la parte central en una videoconferencia) pueden dar lugar a una gran ocupación del *buffer* → Se incrementará el paso del cuantificador por igual para todos los bloques de la imagen:
 - La imagen entera será cuantificada de forma más grosera.
 - Lo ideal sería cuantificar más groseramente sólo algunas zonas (las que están haciendo que se llene el *buffer*).

El procedimiento de adaptación de la cuantificación en función de la ocupación del *buffer* es uno de los aspectos más relevantes en el diseño de un codec → Hace que un codec sea mejor o peor que otro.

5. Control y ocultación de errores

- La información transmitida puede verse alterada durante la transmisión.
- Puede haber errores aleatorios en los bits y errores de borrado.
- Errores aleatorios en los bits:
 - Si se utilizan códigos de longitud fija → Un error en un bit afectará sólo a una palabra.
 - Si se utilizan VLCs → Los errores suelen ocasionar una pérdida del sincronismo y los datos recibidos son inteligibles hasta que se recupera.
- Errores de borrado:
 - Se producen cuando se pierde parte de la información. Ejemplos: pérdida de un paquete en una red de transmisión de paquetes, o fallo en el sistema de pequeña duración.

5. Control y ocultación de errores

- **Protección frente a errores de transmisión:**

- Para detectar y eliminar errores se introduce cierta redundancia en los datos codificados.
- Se introduce información de sincronización (al comienzo de las imágenes, al empezar un GOB...).
- El decodificador realiza las siguientes operaciones:
 1. Determinar si se ha producido algún error, en base a los datos recibidos y a la sintaxis esperada.
 2. En caso de error, resincronizar el decodificador.
 3. Marcar los bloques erróneos.
 4. Minimizar el efecto de los errores → Interpolando desde regiones vecinas, usando imágenes previas...



Ejercicios

H.261 – Julio 2016/17

Responder a las siguientes preguntas relacionadas con los MBs (MacroBloques) de H.261.

a) Describir los siguientes tipos de MBs:

- ★ Intra coded:
- ★ Inter coded:
- ★ MC coded:
- ★ MC not coded:
- ★ Skipped:

b) Explicar a cuáles de esos MBs se puede añadir el sufijo “+Q”, y qué significa.

c) Indicar a qué tipo de MBs se asigna el código más largo, y explicar por qué.

