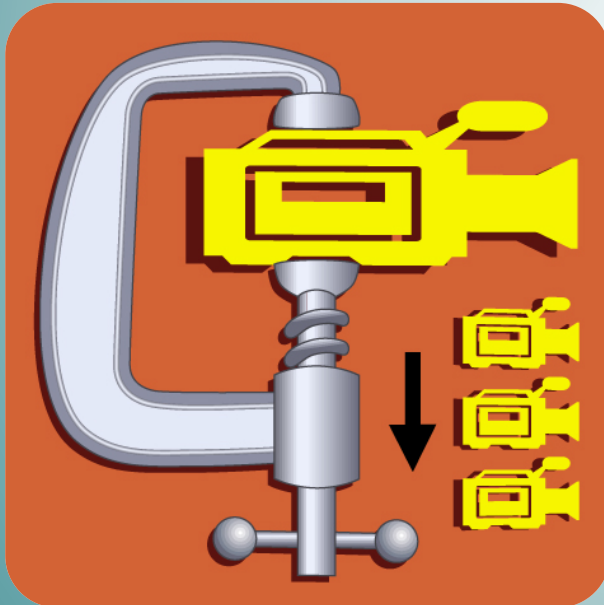


# Compresión de Vídeo

## Tema 2.1. Introducción al H.264



**Juan A. Michell Martín**  
**Gustavo A. Ruiz Robredo**

Departamento de Electrónica y Computadores

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

# NECESIDAD DE COMPRIMIR VIDEO

- Un video sin comprimir (*raw video*) contiene una inmensa cantidad de datos.
- Las capacidades de los sistemas de comunicación y almacenamiento de datos son limitados y caros → Imposible de transmitir video sin comprimir.
- Ejemplo: Video HDTV de 720x1280 pixels/frame a 60 frames/s:

$$\left(720 \times 1280 \frac{\text{pixels}}{\text{frame}}\right) \left(60 \frac{\text{frames}}{\text{sec}}\right) \left(3 \frac{\text{colors}}{\text{pixel}}\right) \left(8 \frac{\text{bits}}{\text{color}}\right) = 1.3 \text{Gb / s}$$

El ancho de banda de un canal HDTV es 20Mb/s → Requiere un factor de compresión de 70 (equivalente a 0.35bits/pixel)

## ➤ Reducción del ancho de banda:

APPLICATION	DATA RATE	
	Uncompressed	Compressed
Video Conference 352x240@15fps	30.4 Mbps	64-768 kbps
CD-ROM Digital Video 352x240@30fps	60.8 Mbps	1.5-4 Mbps
Broadcast Video 720x480@30fps	248.8 Mbps	3-8 Mbps
HDTV 1280x720@60fps	1.33 Gbps	20 Mbps

➤ Video HD de 1080p a 24 fps y RGB :

- 1.5 Hora → 806 Gb
- Bitrate: 1.2 Gbits/s



➤ DVD blu-ray:

- 25 Gb (single layer)
- Bitrate: 36 Mbits/s



➤ TDT (televisión digital) y video streaming:

- 25 Gb (single layer)
- Bitrate: 1 a 20 Mbits/s

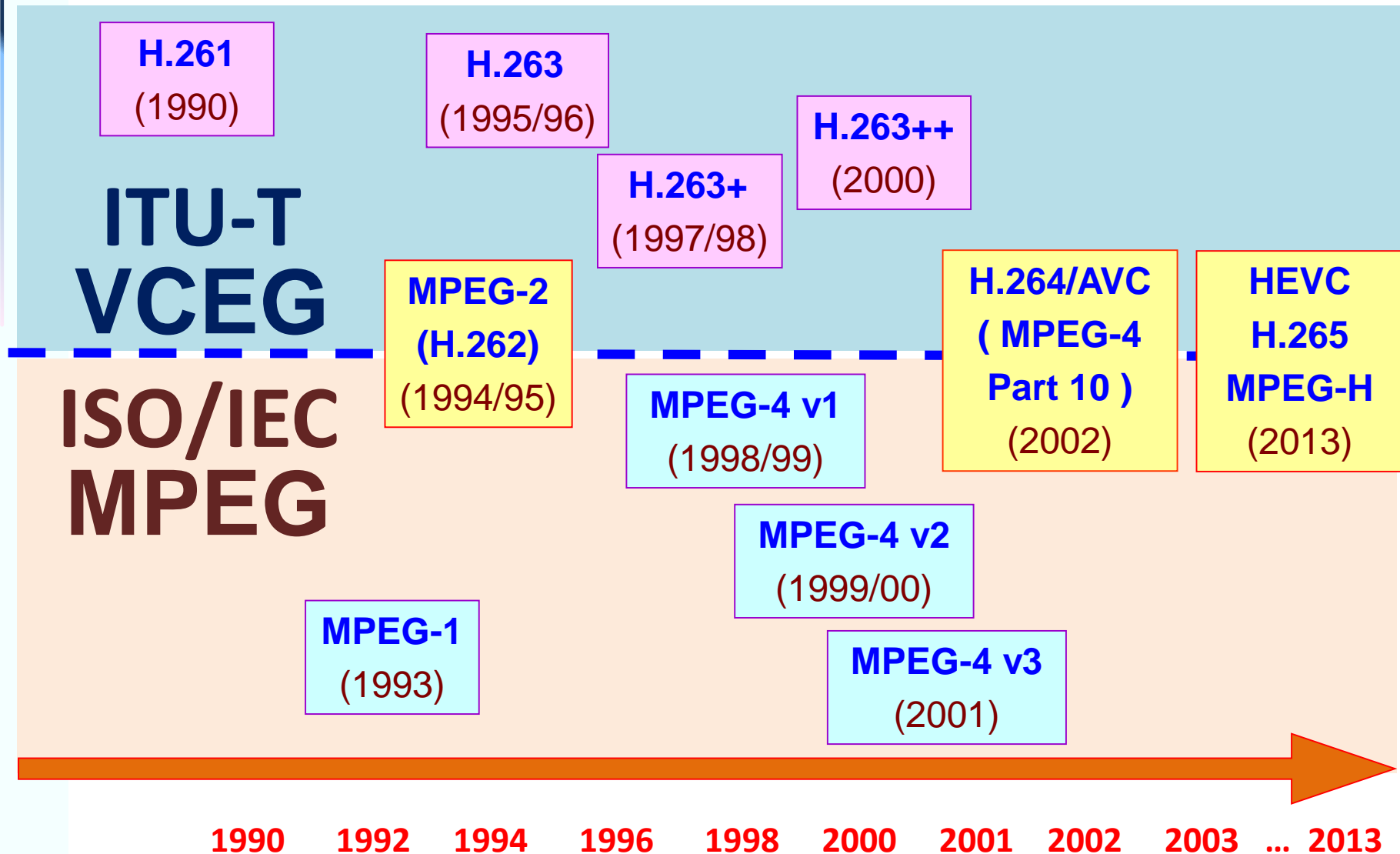


➤ Se necesitan tasas de compresión de 30x a 1300x

# OBJETIVOS DE LOS ESTÁNDARES DE VIDEO

- Asegurar la compatibilidad. Permite la comunicación entre diferentes dispositivos de diferentes fabricantes.
- Reducir costes.
- Reducir la redundancia e información irrelevante (información perceptualmente sin importancia).
- Fuentes de redundancia:
  - **Temporal:** *Frames* adyacentes están altamente relacionadas.
  - **Espacial:** Pixels próximos están relacionados entre sí.
  - **Espacio de color:** Componentes RGB están relacionados entre ellos mismos.

# Cronología de los estándares de video más comunes



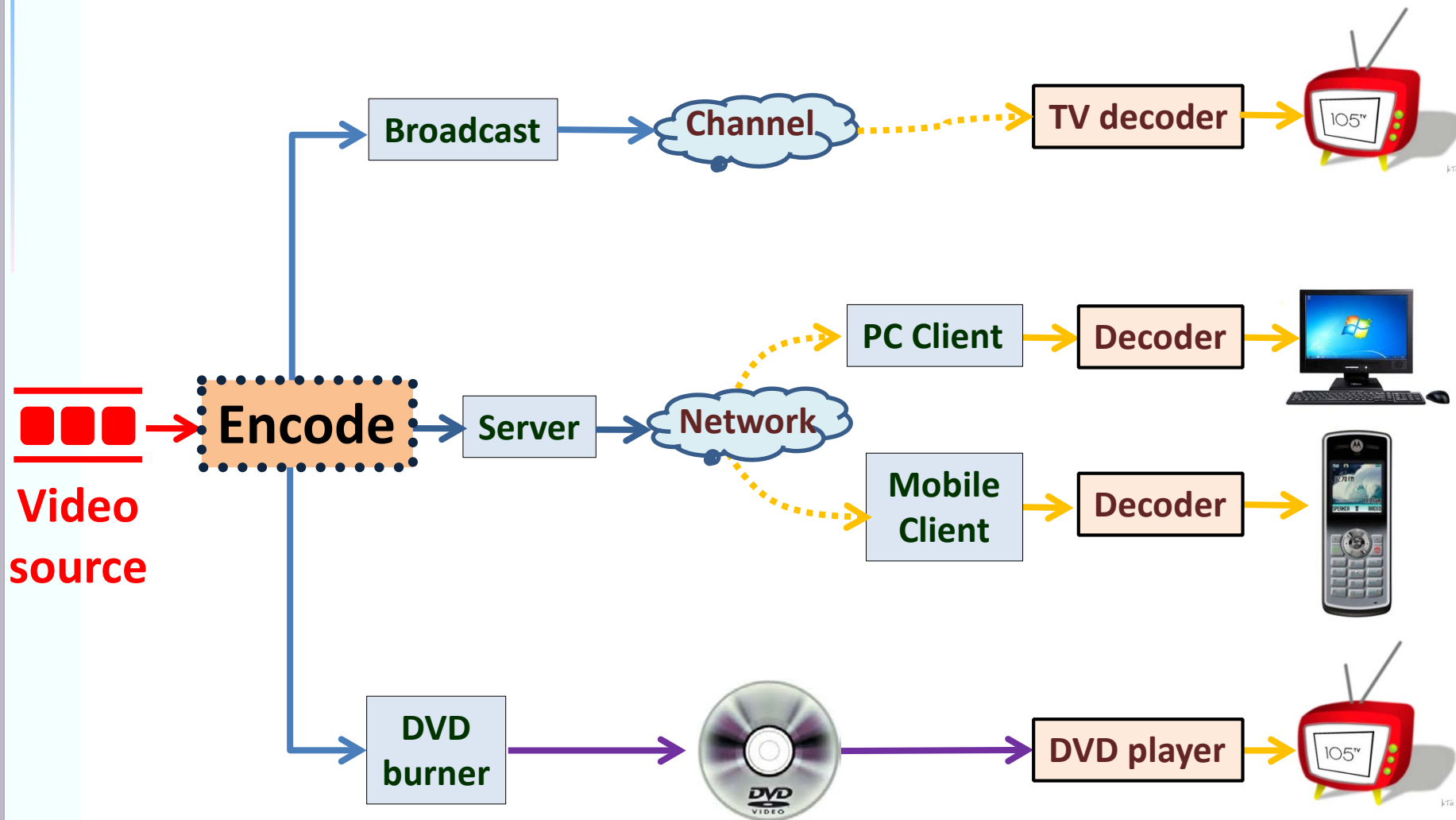
# QUÉ ES EL H.264/AVC O ADVANCED VIDEO CODING?



## Un formato de compresión de video

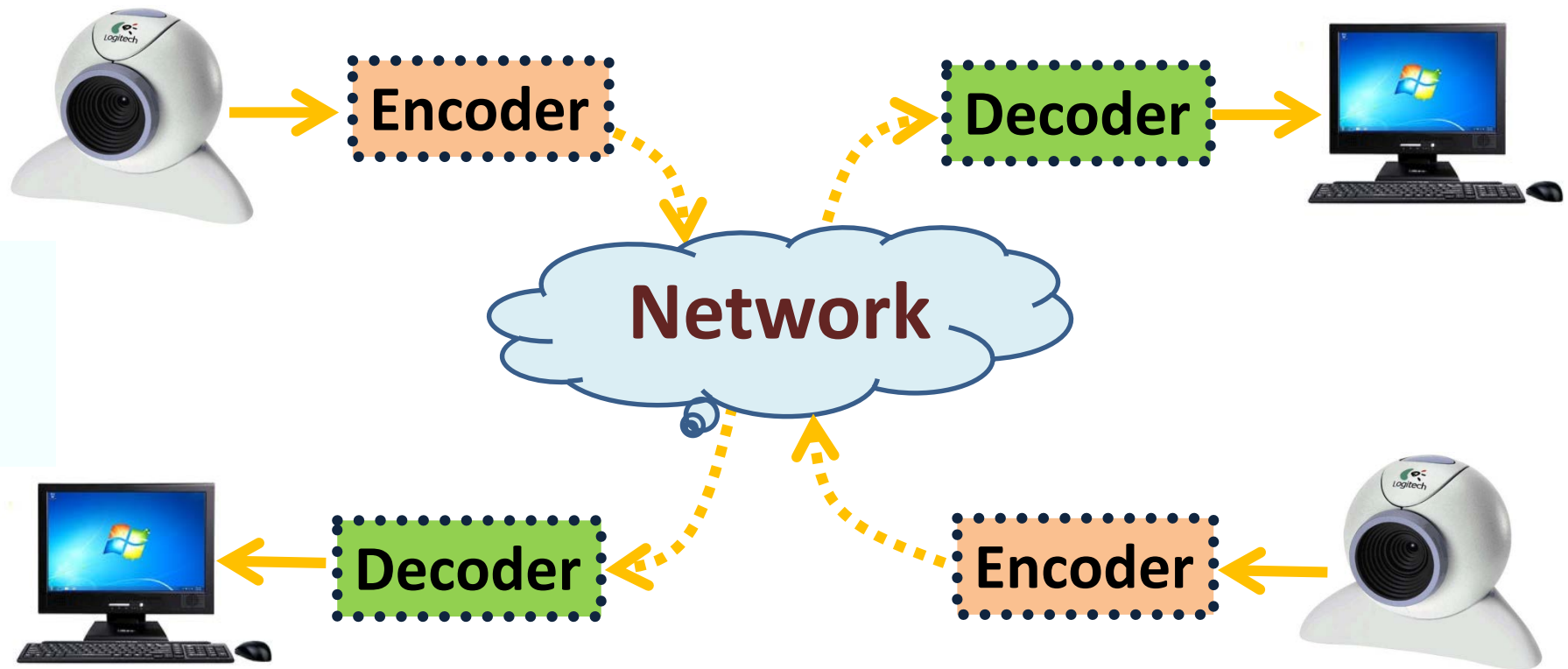
- El H.264/AVC (*Advanced Video Coding*) o MPEG-4 Part 10 es un método y un formato de compresión de video capaz de convertir un video digital en un formato que ocupa menor espacio para ser almacenado y transmitido.

- Típicas aplicaciones *one-way*: televisión digital, DVD/video, TV móvil, video conferencia e internet video streaming.





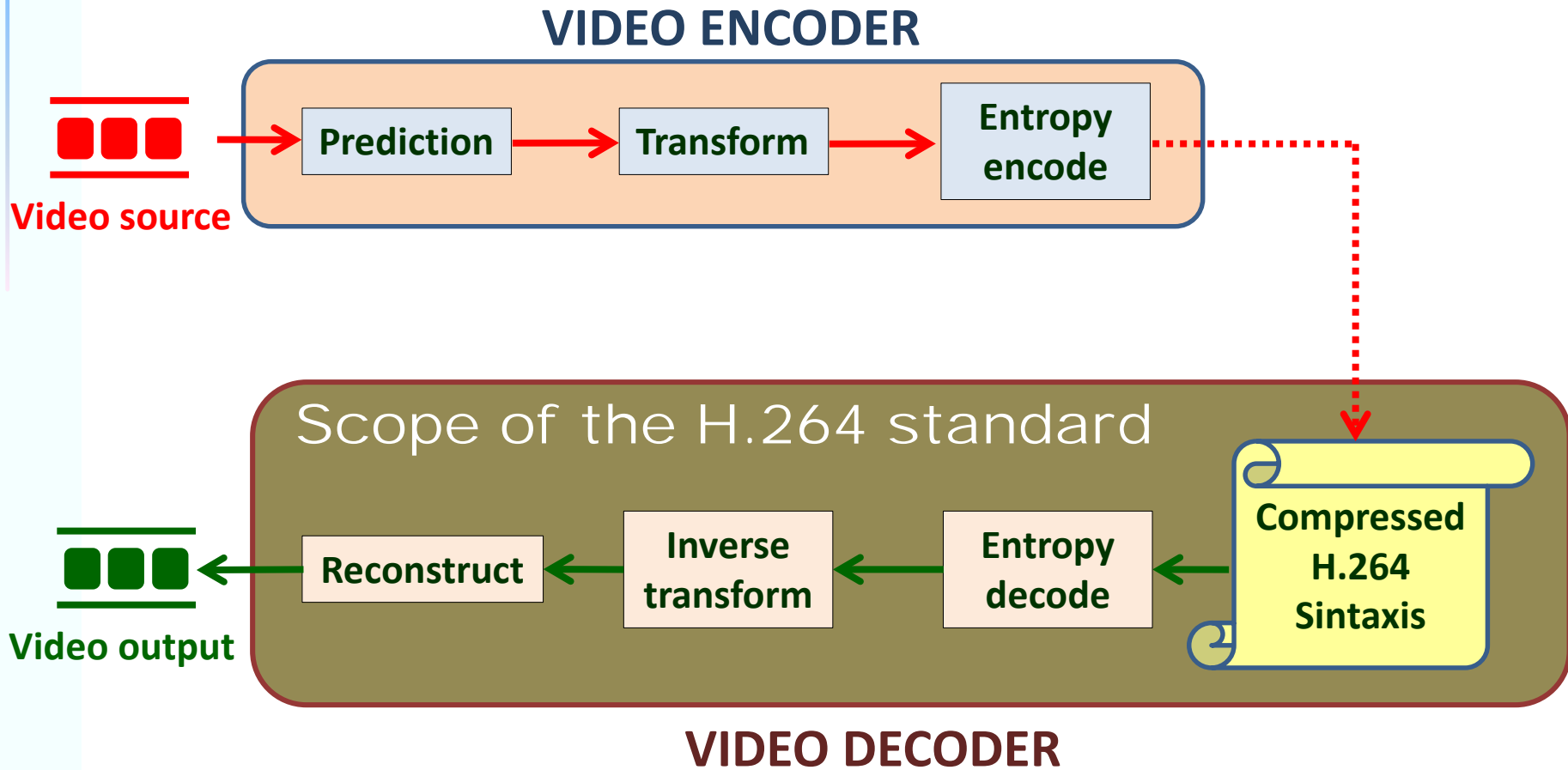
- Típicas aplicaciones *two-way*: video conferencia, Apple iChat, Skype video ... el sistema debe comprimir un video de una cámara local a la vez que descomprime un video transmitido desde una cámara remota.



## Un estándar industrial

- Los grupos expertos en codificación de video ITU-T Study Group 16 (VCEG) e ISO/IEC JTC 1/SC 29/WG 11 (MPEG) bajo el documento **Recommendation H.264: Advanced Video Coding** presentaron dos estándares internacionales en la ITU-T (*International Telecommunication Union*) y en el ISO/IEC (*International Organization for Standardisation/International Electromechanical Commission*).
- Este estándar define un formato y una sintaxis para video comprimido y un método para decodificar esa sintaxis que permite producir una secuencia de video reproducible.
- El estándar **no especifica** como codificar el video sino que deja al diseñador o la compañía la realización del codificador.

## ➤ Proceso de codificación y decodificación del H.264



# Aplicaciones

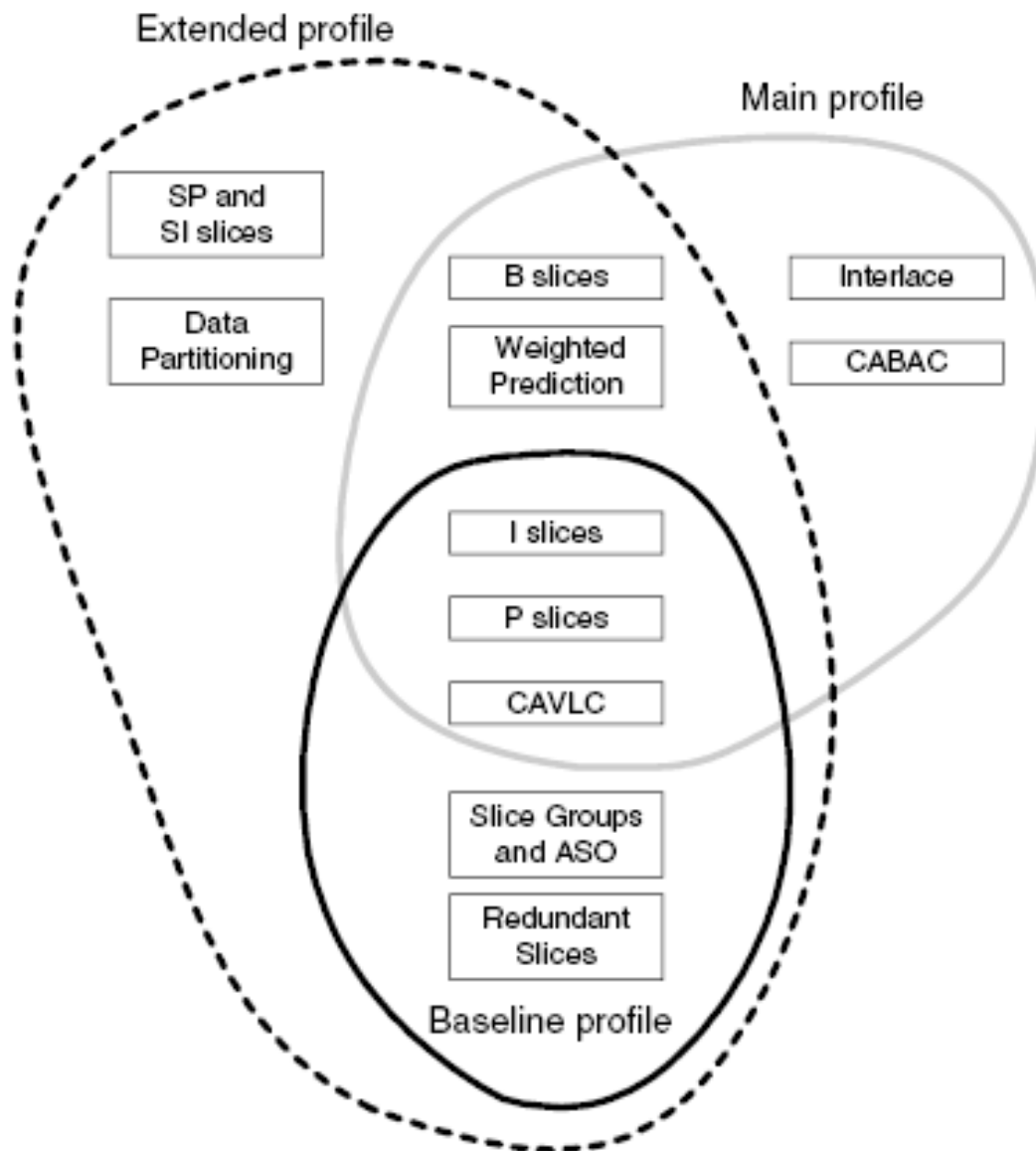
- **Streaming internet:**  
*Vimeo, YouTube y iTunes Store*
- **Software Web**  
*Adobe Flash Player y Microsoft Silverlight*
- **HDTV**  
*ATSC, ISDB-T, DVB-T, DVB-T2, DVB-C, DVB-S and DVB-S2*
- **Blu-ray Discs**
- **Formato de grabación**  
*“AVC/DC - Advanced Video Coding High Definition”  
desarrollado por Sony y Panasonic y usado por  
Canon y Nikon.*

## Un herramienta para compresión video

- EL H.264/AVC describe un conjunto de herramientas para la compresión del video. EL estándar especifica cómo el video codificado con estas herramientas debe ser representado y decodificado.
- EL decodificador del H.264 debe ser capaz de usar un conjunto de esas herramientas conocido como **profile**.

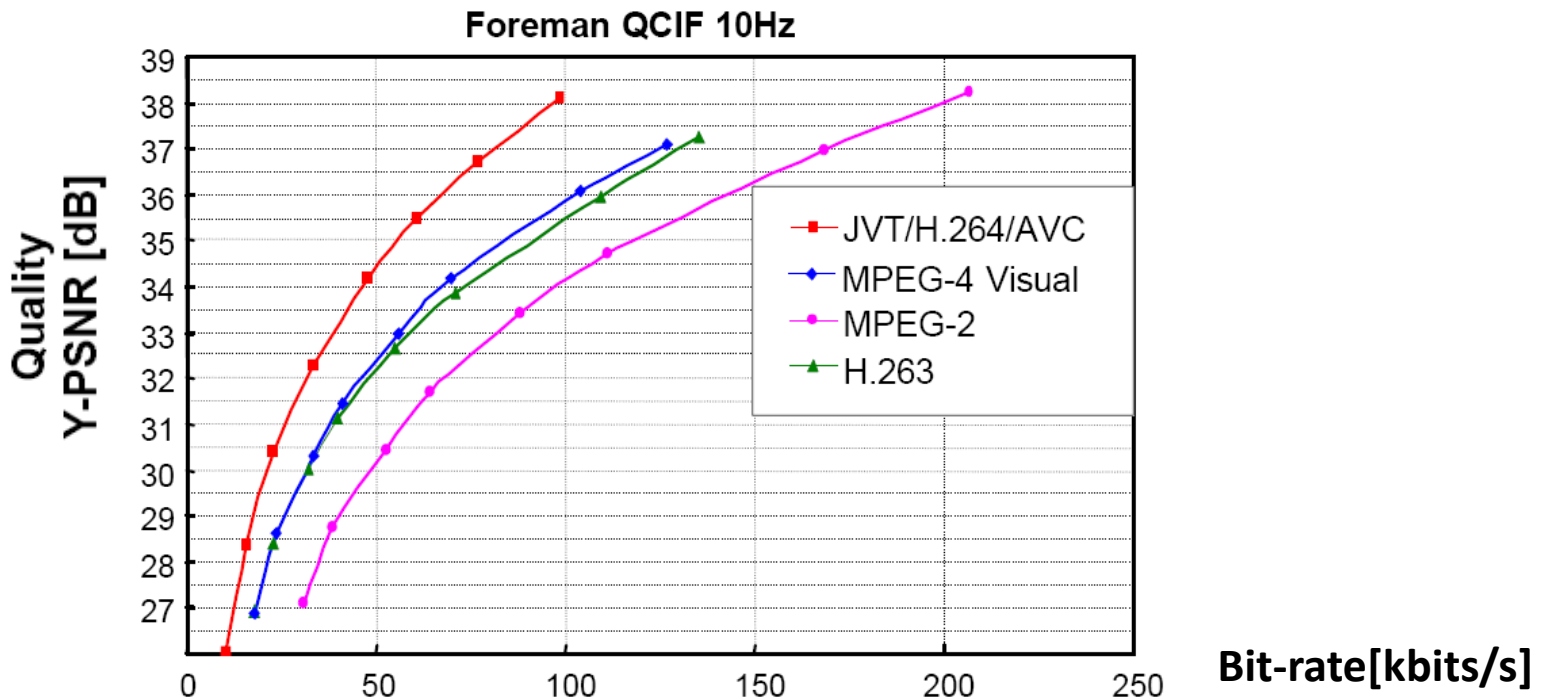
Profile	Description
Baseline Profile (BP)	Primarily for lower-cost applications with limited computing resources, this profile is used widely in videoconferencing and mobile applications.
Main Profile (MP)	Originally intended as the mainstream consumer profile for broadcast and storage applications, the importance of this profile faded when the High profile was developed for those applications.
Extended Profile (XP):	Intended as the streaming video profile, this profile has relatively high compression capability and some extra tricks for robustness to data losses and server stream switching.

Application	Requirements	H.264 Profiles
Broadcast television	Coding efficiency, reliability (over a controlled distribution channel), interlace, low-complexity decoder	Main
Streaming video	Coding efficiency, reliability (over a uncontrolled packet-based network channel), scalability	Extended
Video storage and playback	Coding efficiency, interlace, low-complexity encoder and decoder	Main
Video conferencia	Coding efficiency, reliability, low latency, low-complexity encoder and decoder	Baseline
Mobile video	Coding efficiency, reliability, low latency, low-complexity encoder and decoder, low power consumption	Baseline
Studio distribution	Lossless or near-lossless, interlace, efficient transcoding	Main High



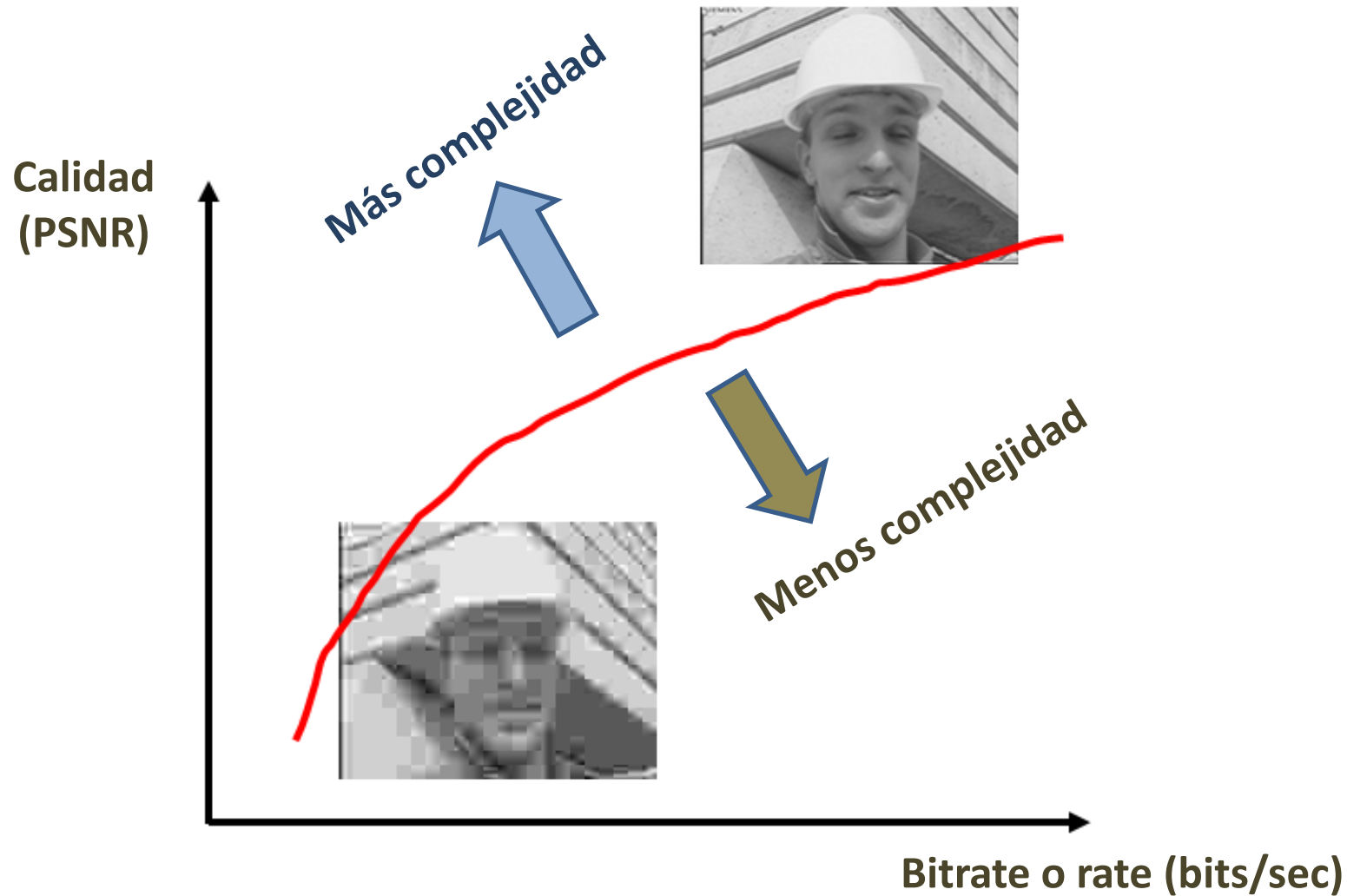
## Mejor compresor de video actual

- Comparado con otros estándares de compresión de video, el H.264/AVC proporciona mejores parámetros de compresión y calidad de imagen.
- Proporciona mejor calidad de imagen a la misma tasa de *bitrate* o proporciona un menor *bitrate* de video comprimido para la misma calidad de imagen.





# CURVAS RATE-DISTORSIÓN



## PNSR o Peak Signal to Noise Ratio

- El **PSNR** es una medida en escala logarítmica que indica el error entre dos imágenes  $I_A$  e  $I_B$  de dimensiones  $M \times N$ , de forma que

$$\text{PSNR} = 10 \cdot \log_{10} \left( \frac{2^n - 1}{\text{MSE}} \right)$$

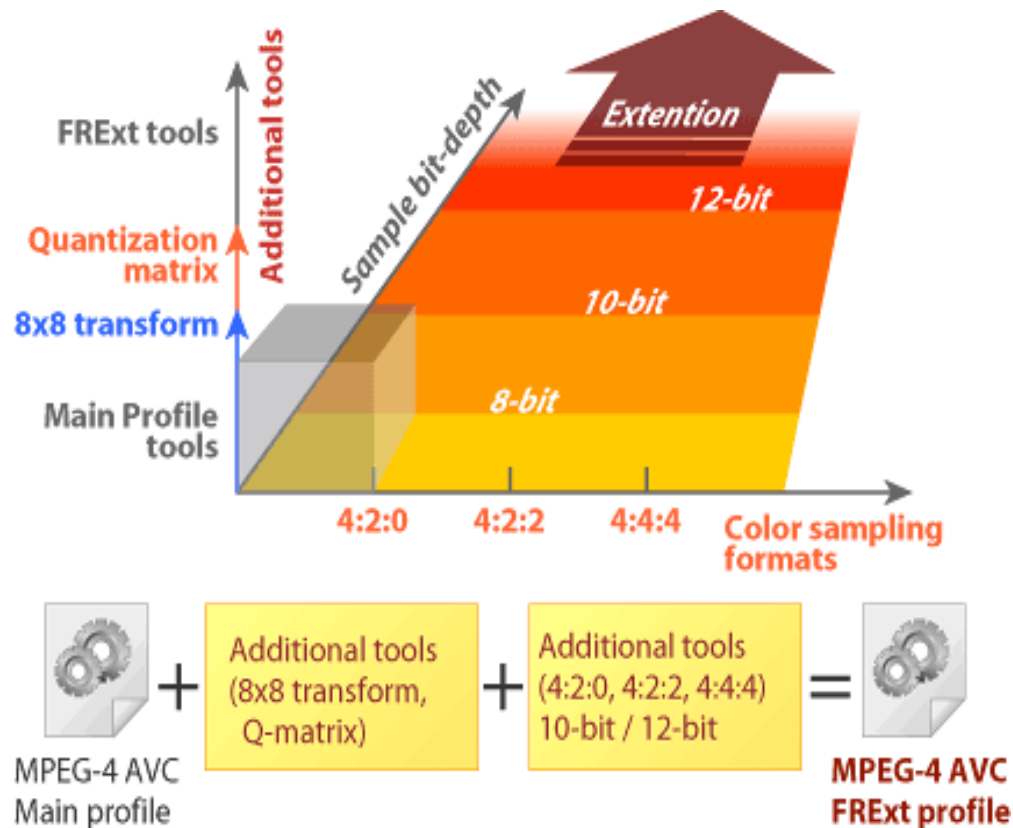
donde **MSE** o *Mean Squared Error* es el error cuadrático medio definido

$$\text{MSE} = \frac{1}{M \cdot N} \sum_{i=1}^N \sum_{j=1}^M [I_A(i, j) - I_B(i, j)]^2$$

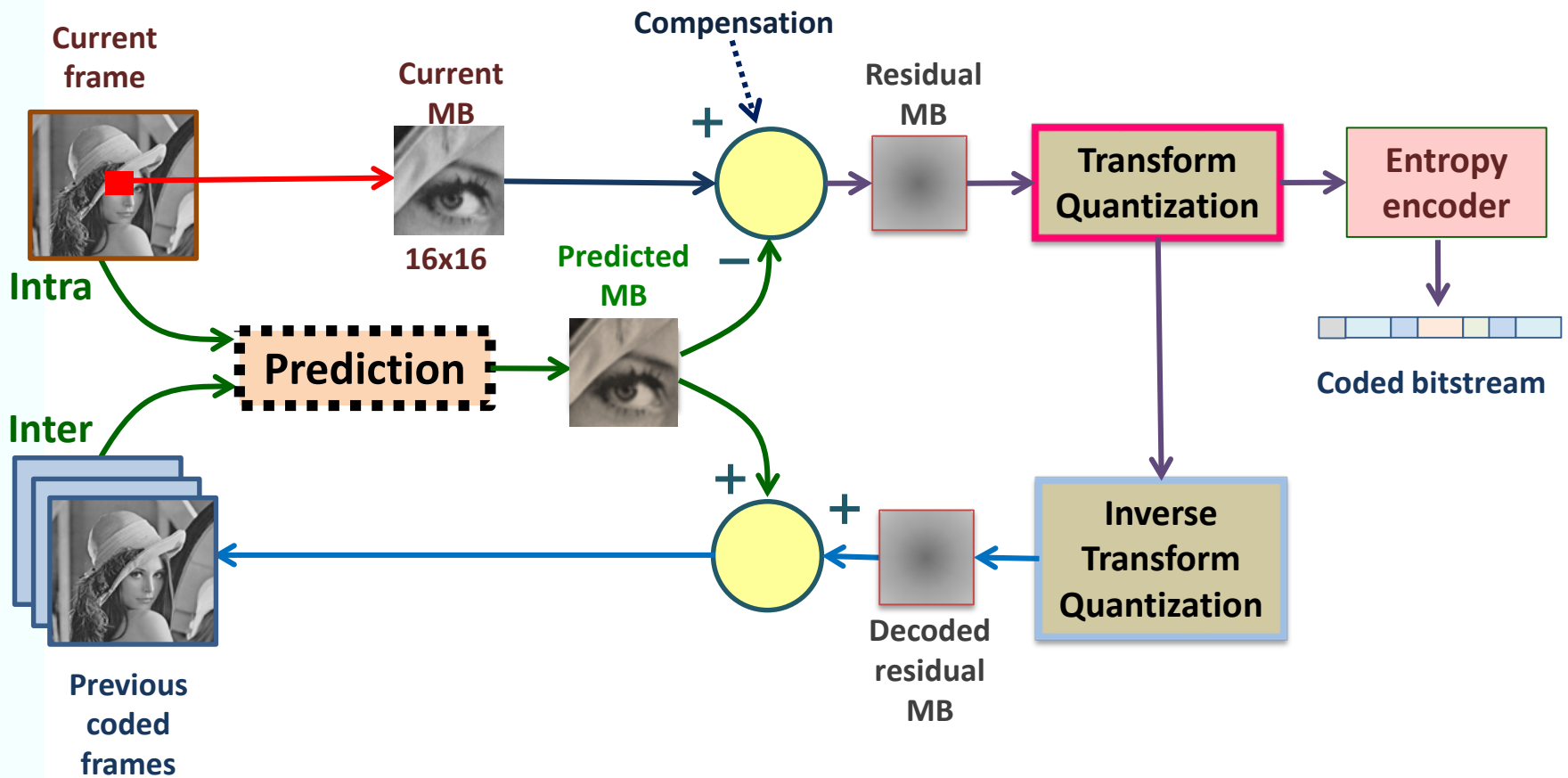
y **n** es el número de bits usado para representar cada pixel.

# H.264 FREXT (*FIDELITY RANGE EXTENSIONS*) AMENDMENT

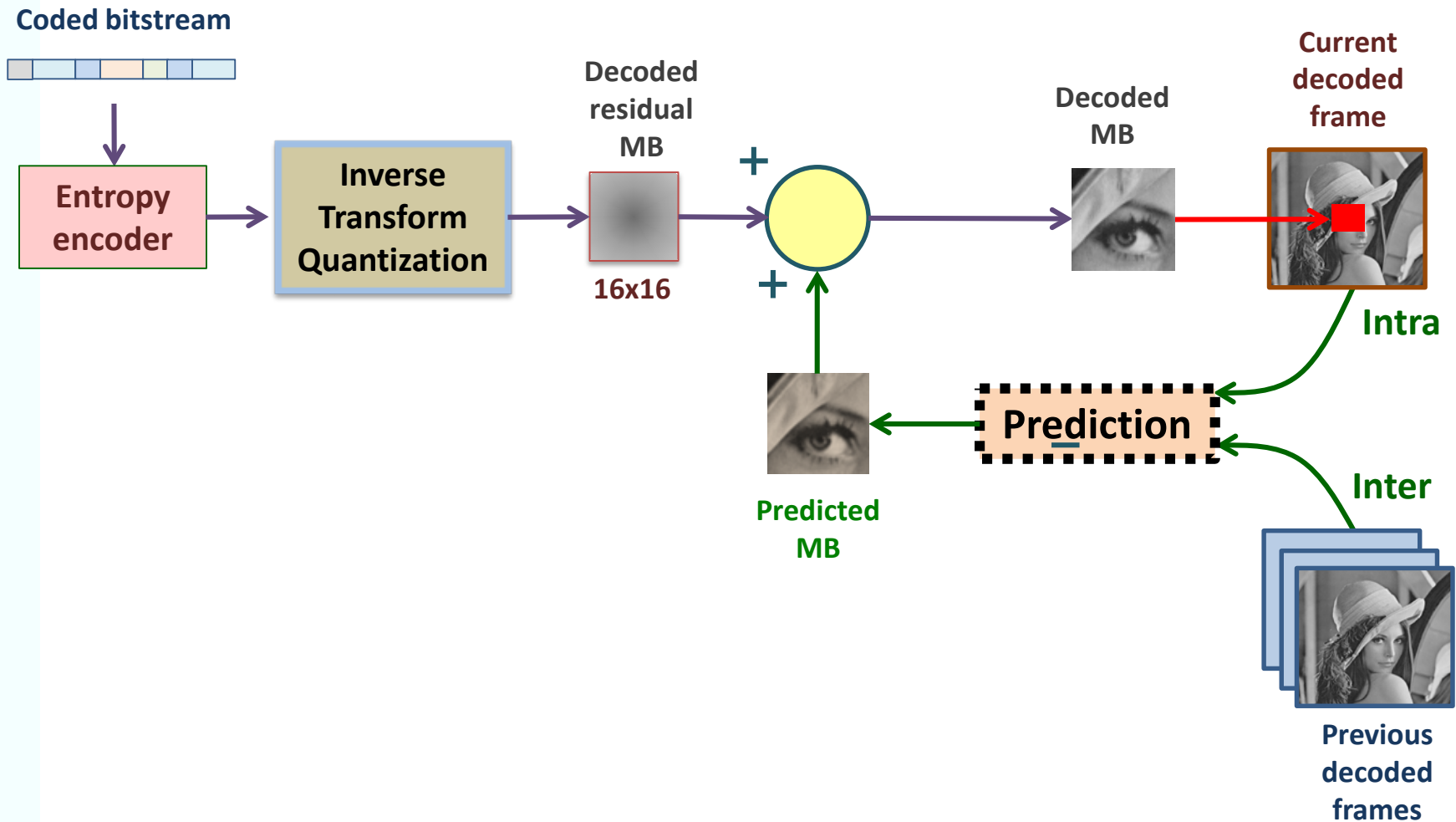
- Finalizado en Julio 2004.
- Propuesto para codificar video de alta fidelidad para aplicaciones profesionales y alta definición.



# ESTRUCTURA DEL CODIFICADOR H.264



# ESTRUCTURA DEL DECODIFICADOR H.264



# PREDICCIÓN ESPACIAL Y TEMPORAL

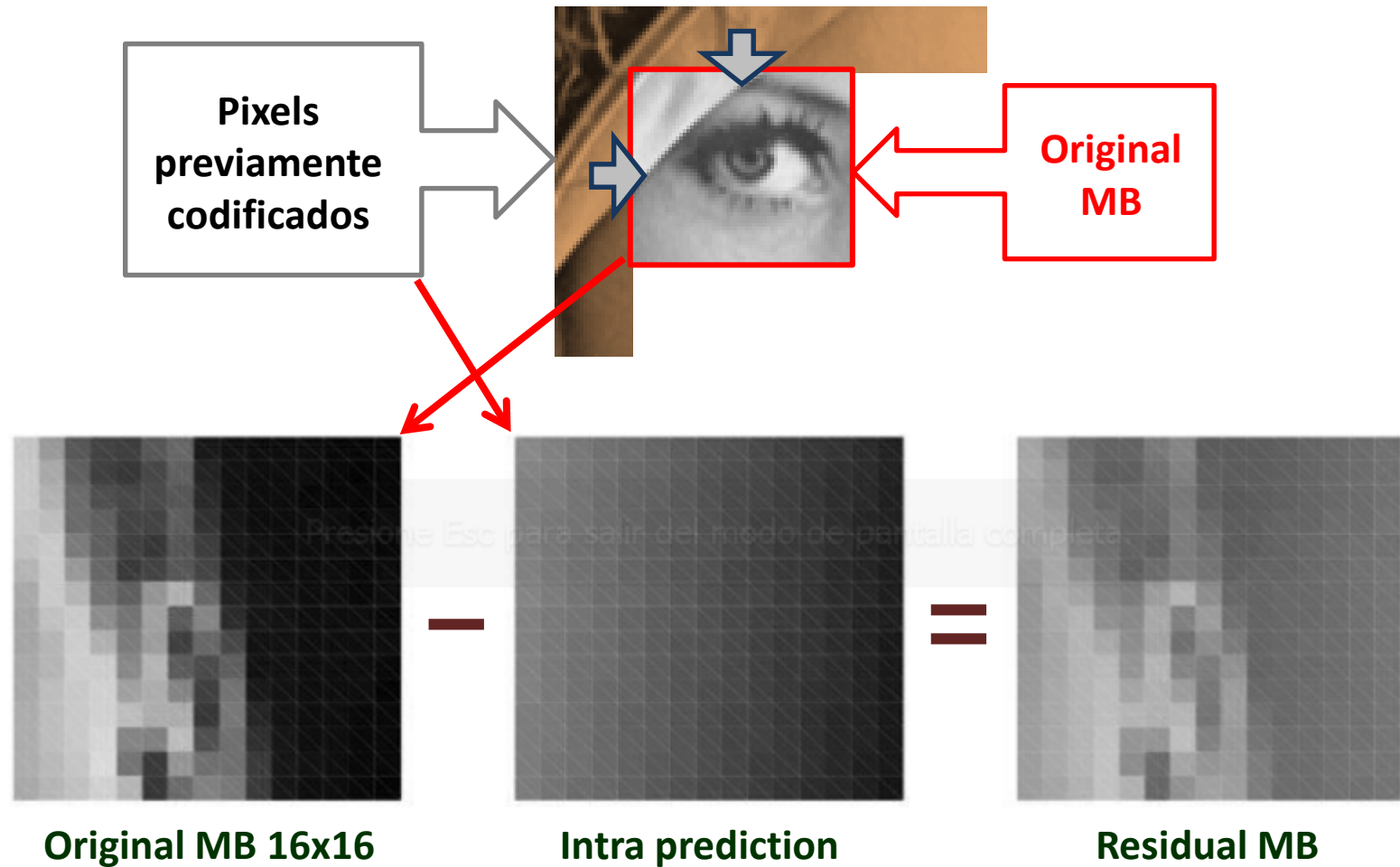
- El codificador construyen la predicción del MB actual basado en datos codificados previamente, utilizando dos tipos de predicción:
  - Intrapredicción o predicción *intra* a partir de datos de la propia *frame*. Predicción *espacial*.
  - Interpretación o predicción *inter* a partir de datos de *frames* previamente codificadas. Predicción *temporal*.



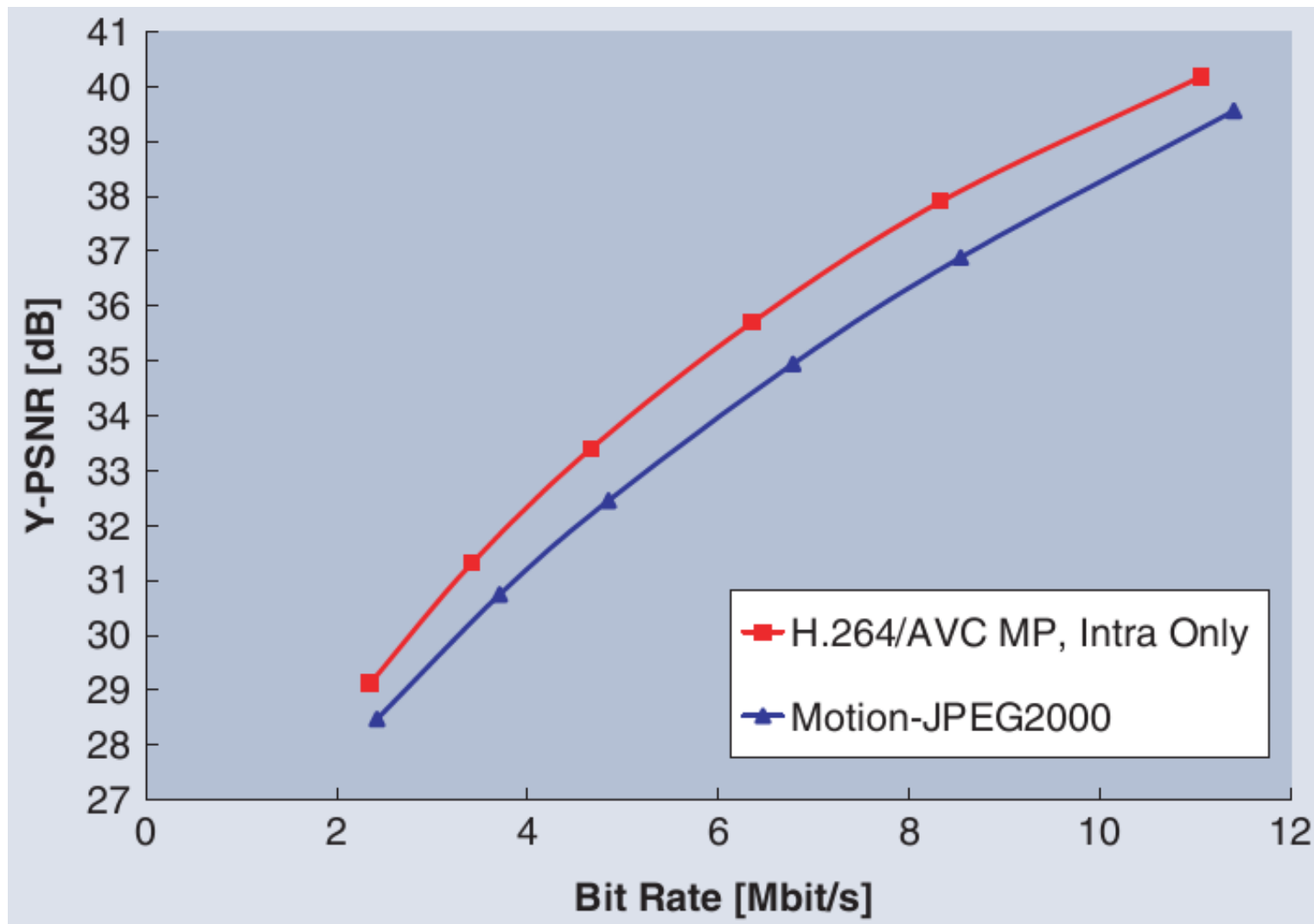
1. Se comprime  y  en la *frame* 1 (*Intra*).

2. Se comprime el movimiento de  en el resto de las frames (*inter*).

## Intrapredicción o predicción *intra*

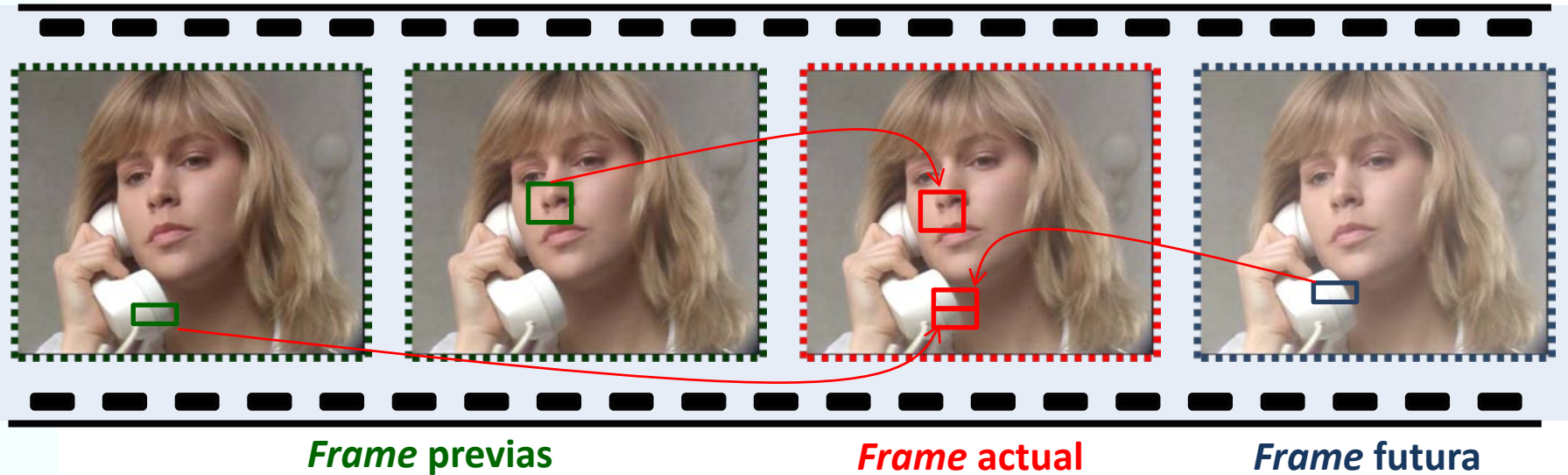


- Comparación entre la codificación de un video solo en modo predicción Intra en el H.264 con respecto a la codificación JPEG2000.





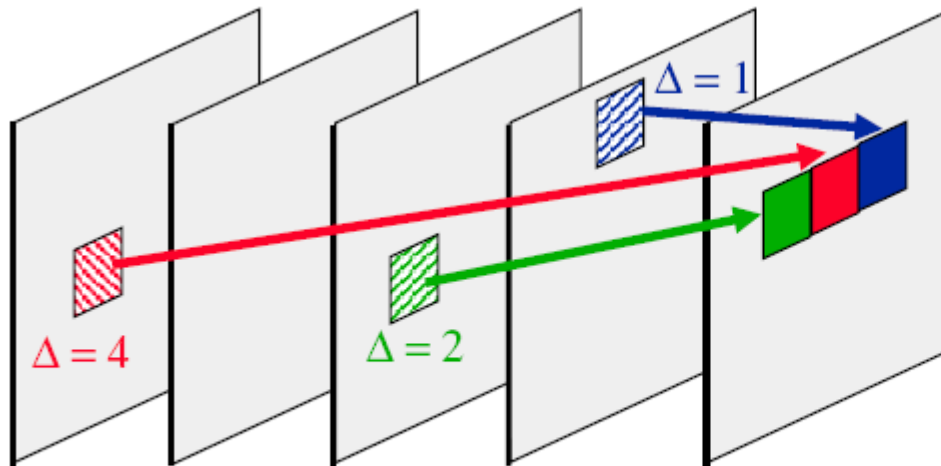
## Interpredicción o predicción inter



$$\begin{array}{ccc} \text{Original MB } 16 \times 16 & - & \text{Interprediction} \\ \hline & = & \text{Residual MB} \end{array}$$

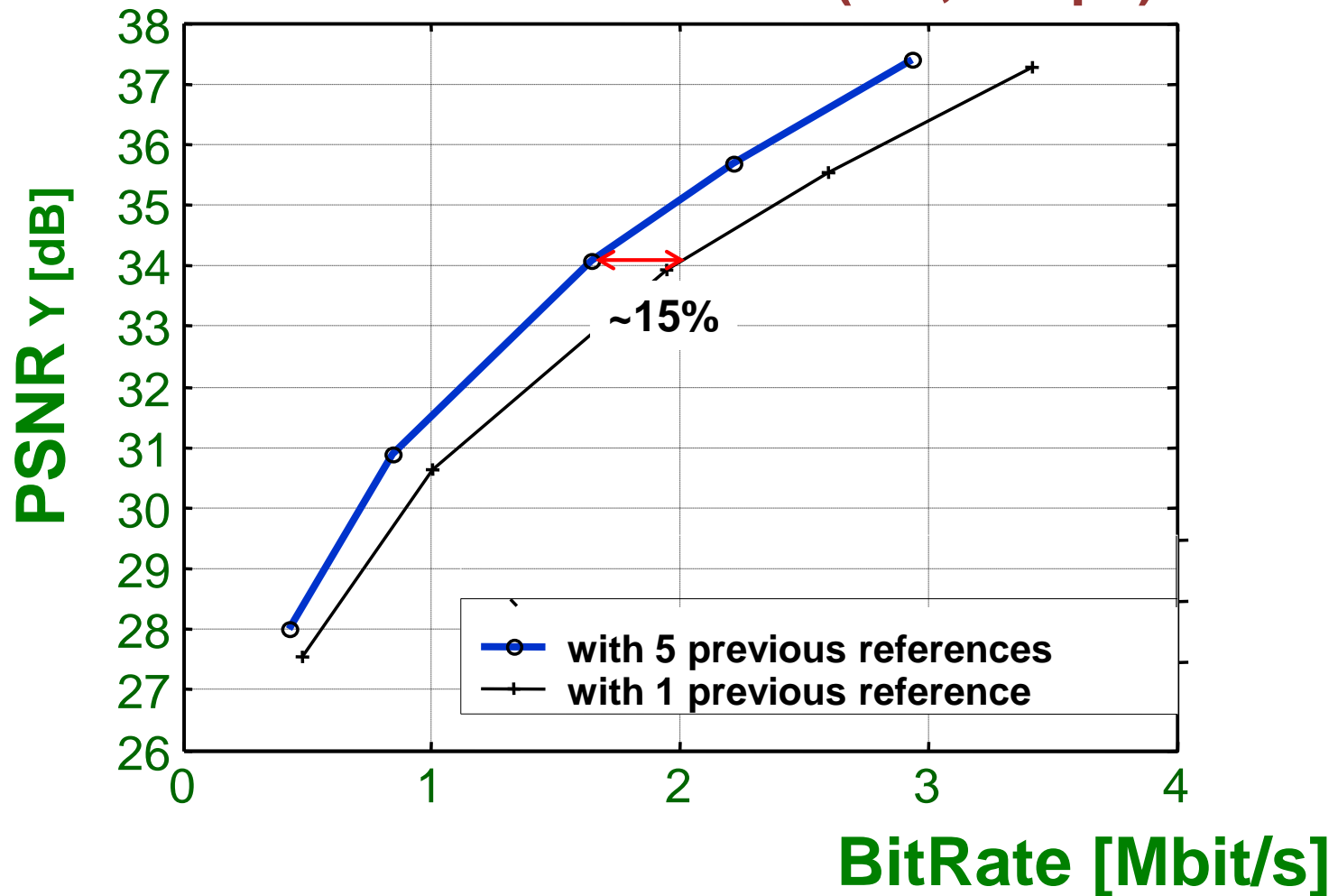
## Frames de referencia múltiple

- Las técnicas de estimación de movimiento basados en múltiples referencias proporcionan mayor predicción en la inter-predicción y mejora la robustez en la pérdida de datos.
- El mayor inconveniente es la necesidad de almacenar en el decodificador las imágenes de referencia, con el consiguiente coste en hardware.



- Utilizar un número mayor de *frames* de referencia mejora la codificación del video.

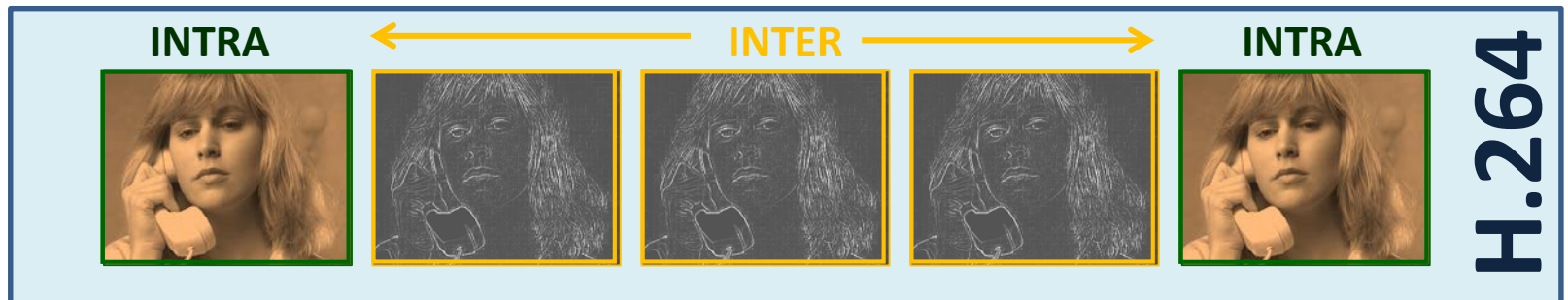
## Mobile & Calendar (CIF, 30 fps)



- H.264 maneja una secuencia de predicción combinada de *inter* e *intra*



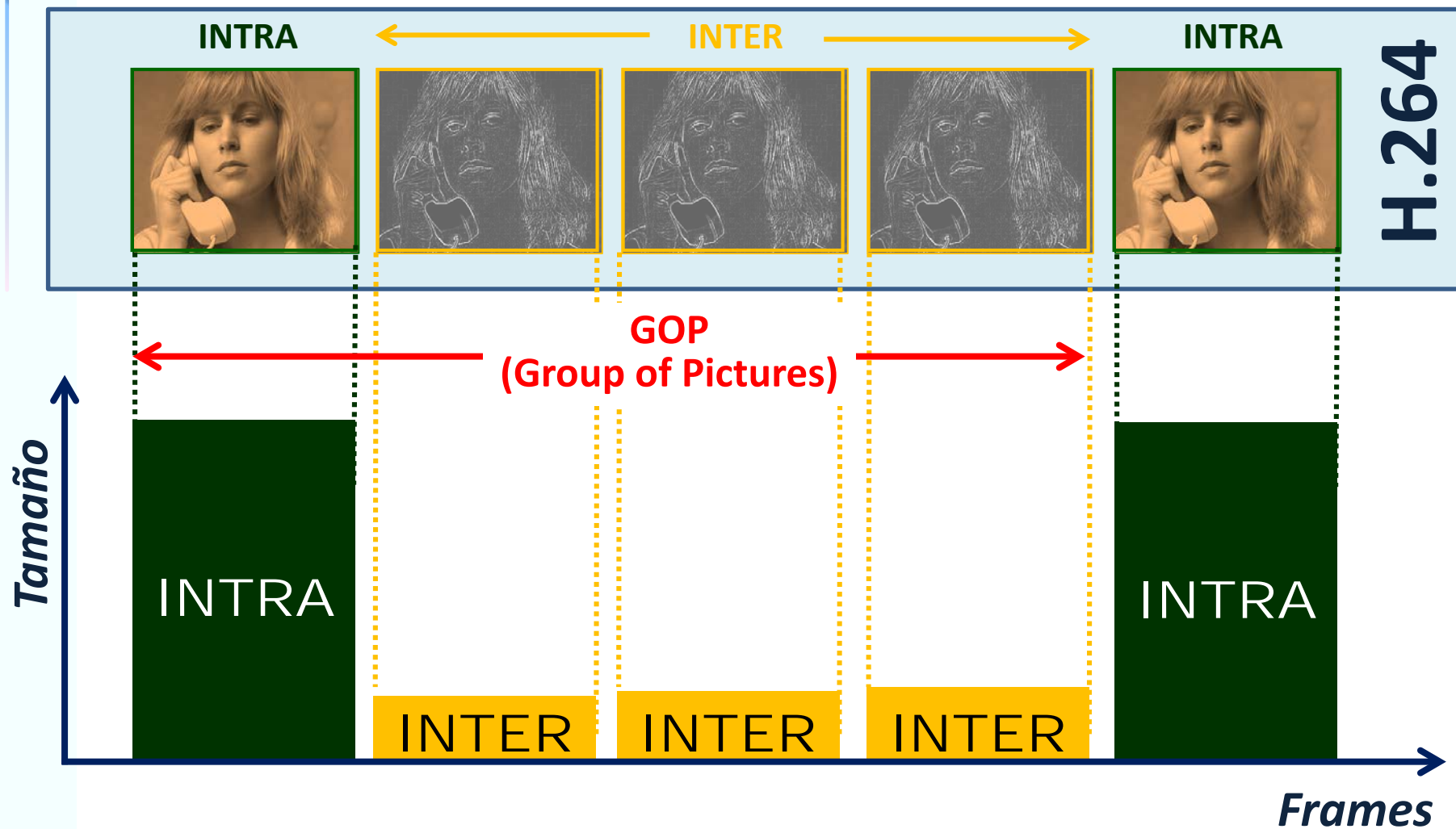
↓ **ENCODING**



↓ **DECODING**

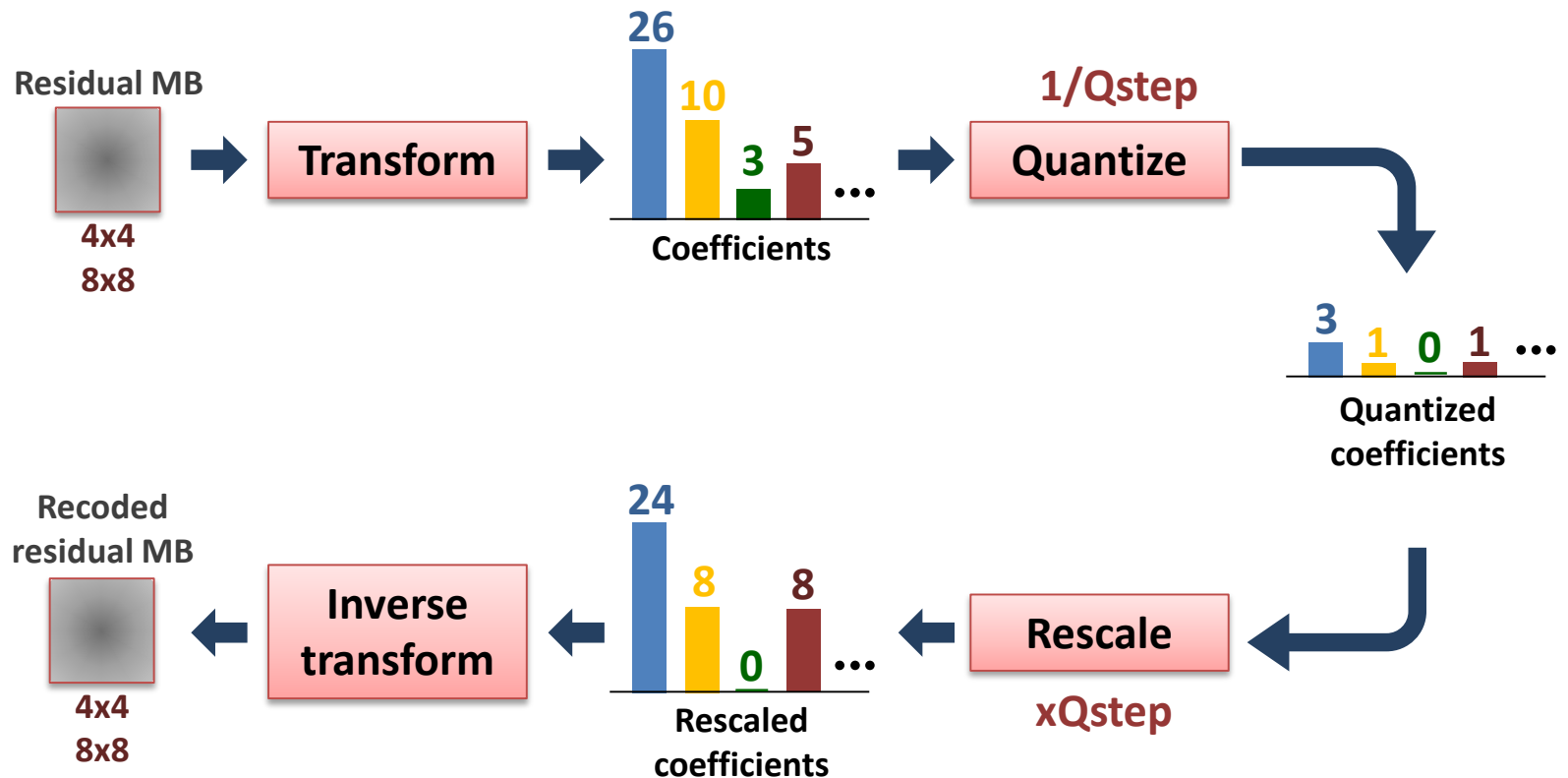


- Tamaño de ocupación de las *frames* comprimidas en modo *intra* e *inter*.



# TRANSFORMADA Y CUANTIZACIÓN

- El H.264 aplica una transformada basada en la DCT de dimensiones 4x4 y 8x8 de números enteros a los MB residuales.



## ➤ Ejemplo numérico de cuantificación y reescalado

$$\begin{bmatrix} 73 & 87 & 64 & 13 \\ 40 & 63 & 23 & 2 \\ 36 & 24 & 68 & 26 \\ 29 & 98 & 67 & 12 \end{bmatrix}$$

Original block 4x4

DC

$$\begin{bmatrix} 181 & 47 & -66 & 4 \\ 6 & 30 & 16 & -2 \\ 40 & 18 & -29 & -14 \\ 13 & -20 & 14 & 34 \end{bmatrix}$$

Forward transform coefficients

$$\begin{bmatrix} 74 & 89 & 65 & 11 \\ 42 & 64 & 23 & 4 \\ 30 & 29 & 64 & 24 \\ 27 & 99 & 67 & 8 \end{bmatrix}$$

Inverse Transform Qstep=10

$$\begin{bmatrix} 180 & 50 & -70 & 0 \\ 10 & 30 & 20 & 0 \\ 40 & 20 & -30 & -10 \\ 10 & -20 & 10 & 30 \end{bmatrix}$$

Rescaled coefficients Qstep=10

$$\begin{bmatrix} 18 & 5 & -7 & 0 \\ 1 & 3 & 2 & 0 \\ 4 & 2 & -3 & -1 \\ 1 & -2 & 1 & 3 \end{bmatrix}$$

Quantified coefficients Qstep=10

$$\begin{bmatrix} 71 & 78 & 59 & 24 \\ 29 & 61 & 23 & 2 \\ 40 & 23 & 73 & 29 \\ 30 & 103 & 60 & 15 \end{bmatrix}$$

Inverse Transform Qstep=20

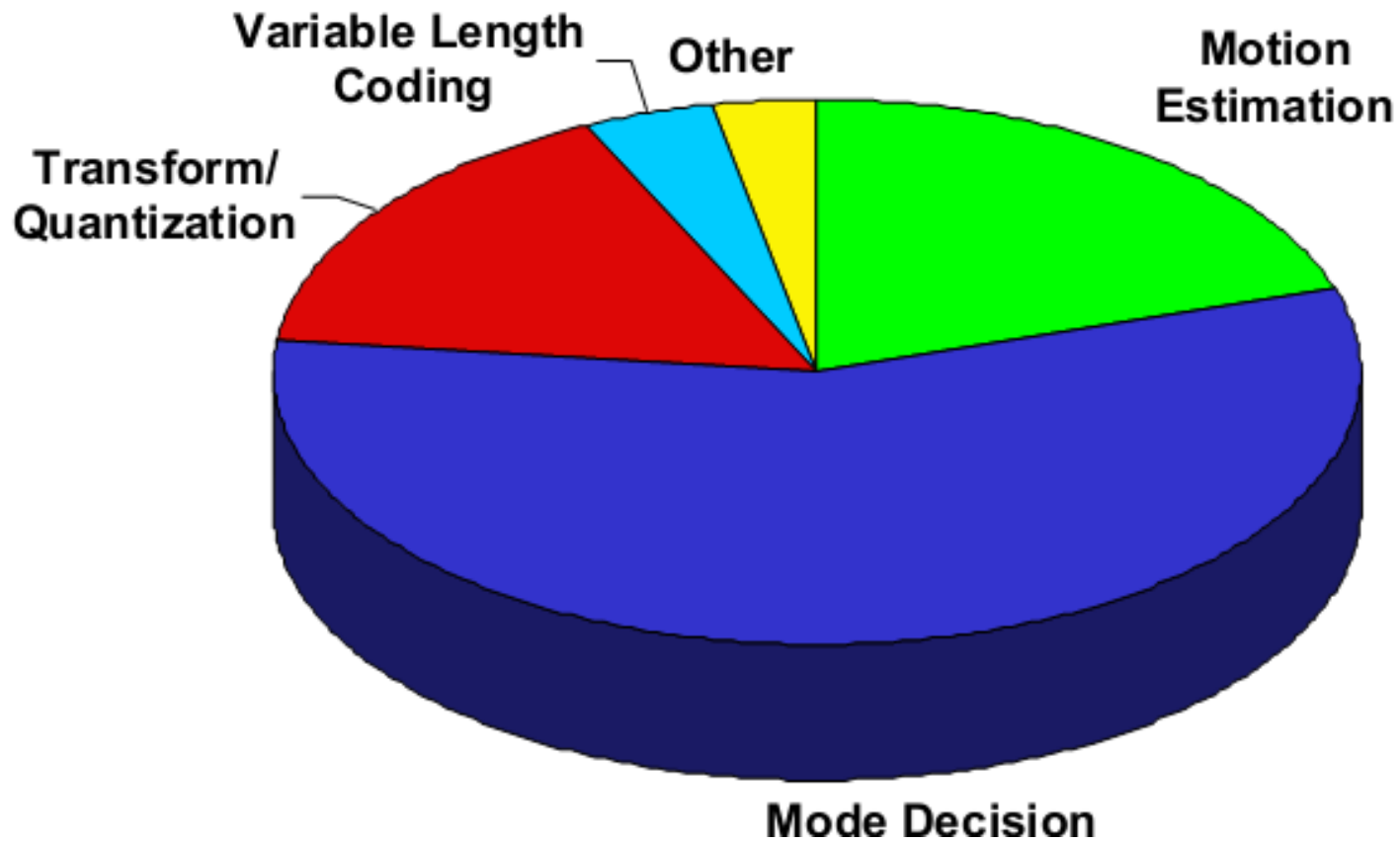
$$\begin{bmatrix} 180 & 40 & -60 & 0 \\ 0 & 20 & 20 & 0 \\ 40 & 20 & -20 & -20 \\ 20 & -20 & 20 & 40 \end{bmatrix}$$

Rescaled coefficients Qstep=20

$$\begin{bmatrix} 9 & 2 & -3 & 0 \\ 0 & 1 & 1 & 0 \\ 2 & 1 & -1 & -1 \\ 1 & -1 & 1 & 2 \end{bmatrix}$$

Quantified coefficients Qstep=20

# COSTE COMPUTACIONAL





# IMPLEMENTACIÓN DEL H.264

## ❑ Grupos relacionados:

- MPEG website <http://www.mpeg.org>
- JVT website: <ftp://standards.polycom.com>
- [www.mpegif.org](http://www.mpegif.org)

## ❑ Software

- <http://iphone.hhi.de/suehring/tml/download>

## ❑ Productos e implementaciones

- Multitud de equipos y Chips:

[http://en.wikipedia.org/wiki/H.264/MPEG-4\\_AVC\\_products\\_and\\_implementations](http://en.wikipedia.org/wiki/H.264/MPEG-4_AVC_products_and_implementations)

## ❑ Secuencias de test

- [ftp.tnt.uni-hannover.de/pub/jvt/sequences/](ftp://ftp.tnt.uni-hannover.de/pub/jvt/sequences/)
- <http://trace.eas.asu.edu/yuv/>
- [ftp://ftp.ldv.ei.tum.de/videolab/public/SVT\\_Test\\_Set/](ftp://ftp.ldv.ei.tum.de/videolab/public/SVT_Test_Set/)
- <http://videocoders.com/yuv.html>

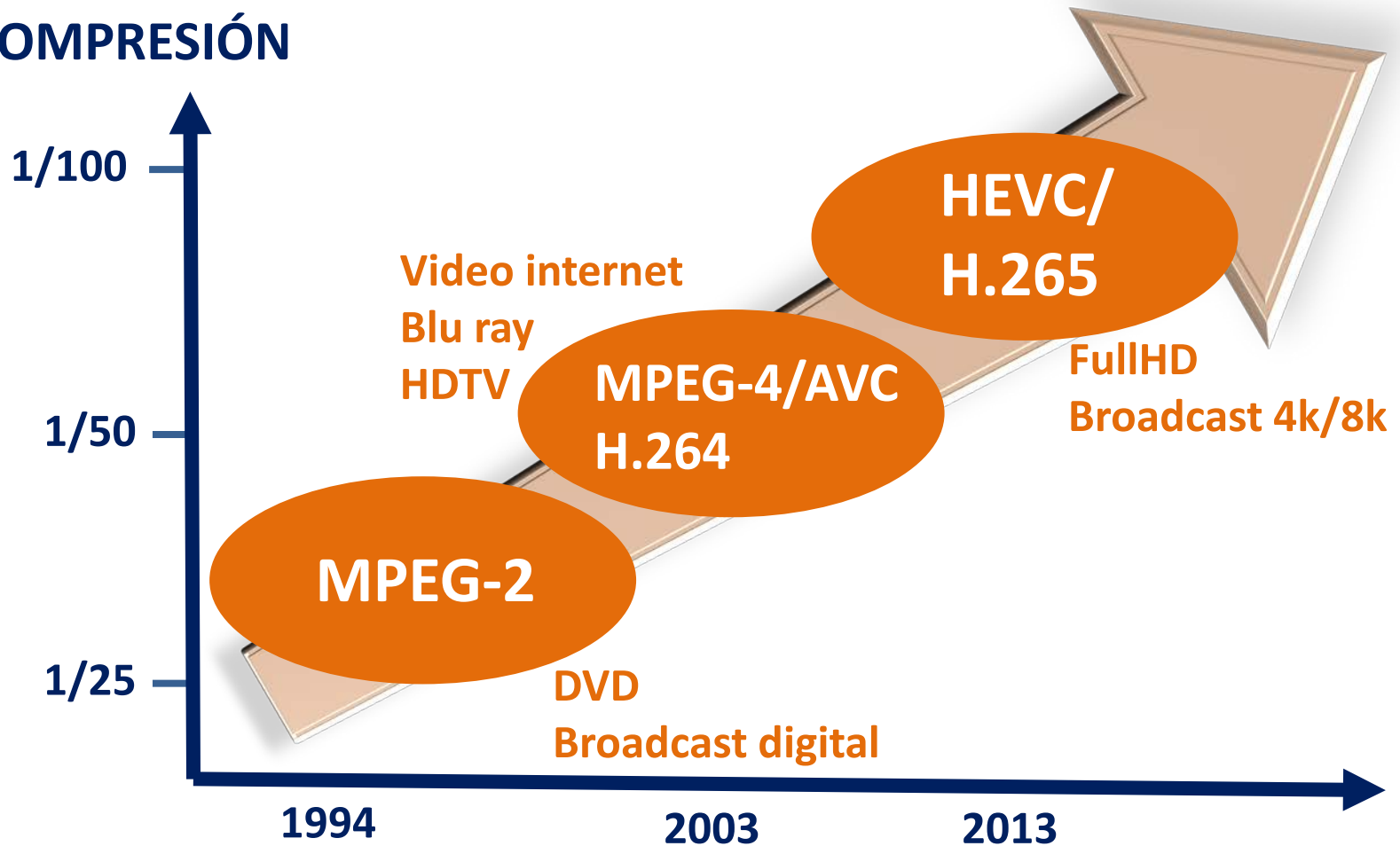
## ❑ Links interesantes

- <https://sites.google.com/site/wwwgmsu/usefullink>
- [http://nsl.cs.sfu.ca/wiki/index.php/Video\\_Library\\_and\\_Tools](http://nsl.cs.sfu.ca/wiki/index.php/Video_Library_and_Tools)

# HEVC O H.265 COMO FUTURO ESTÁNDAR

- El **H.265** o **MPEG-H Part 2** o **HEVC** (*High Efficiency Video Coding*) se presenta como una evolución del H.264/AVC.
- El H.265 mejora en un 40% la tasa de compresión del H.264/AVC pero sus necesidades de cómputo son hasta 4 veces superiores, apenas hay chips para las tecnologías actuales.
- Es capaz de manejar *frames* con resoluciones de 320×240 a **7680×4320** pixels.
- El primer **Draft** se presentó a principios del 2013.
- Más información en <http://www.h265.net>

## RATIO DE COMPRESIÓN



# Aplicación: MSU Video Quality Measurement

MSU Video Quality Measurement Tool 2.7.3

**MSU Video Quality Measurement Tool**  
Version 2.7.3 / Easy way of codecs comparison  
**Not for usage in companies!**

Step 1: File selection

Original file (avi, avs, yuv, bmp):

☐ Open with AVISynth

Processed (compressed):

☐ Comparative analysis

☐ Open with AVISynth

Second processed (another codec):

☐ Open with AVISynth

☐ Use mask file:

☐ Use black mask ☐ Open with AVISynth

Step 2: Metric Selection

Color component


☒ Y-YUV ☐ U-YUV ☐ V-YUV ☐ L-LUV ☐ R-RGB ☐ G-RGB ☐ B-RGB

Step 3: Output Selection

☐ Save CSV file

☐ Save metric visualization video / image

☐ Save "bad frames"

  
GRAPHICS & MEDIA LAB  
VIDEO GROUP

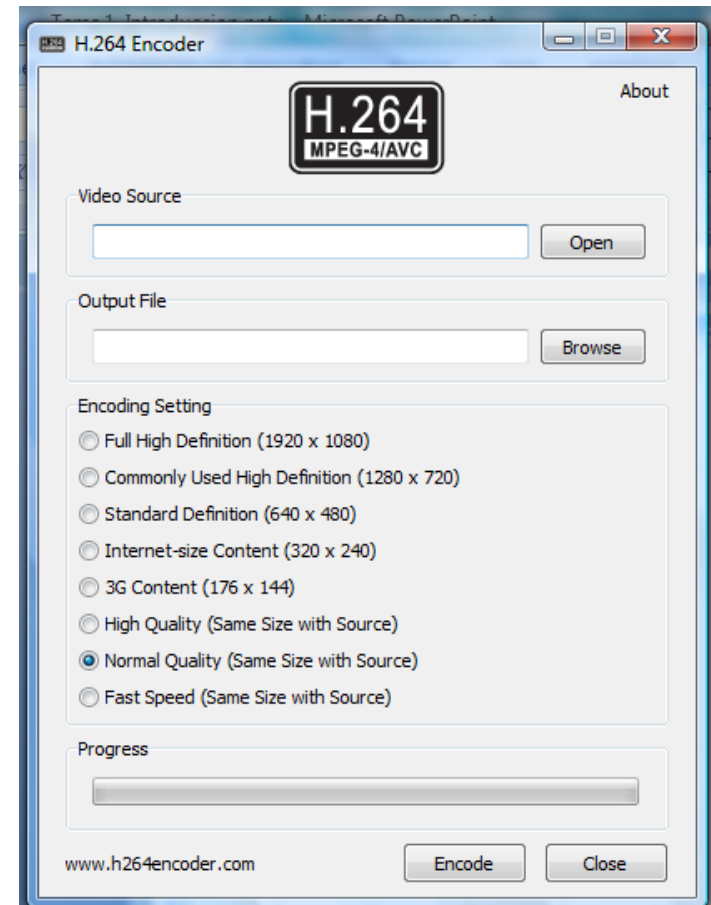
Ready ☒ Show results visualization

# SOFTWARE H.264

➤ Software básico H264 encoder <http://www.h264encoder.com>

➤ Formatos de entrada:

- \*.avi
- \*.wmv
- \*.mp4
- \*.flv
- \*.mov
- \*.3gp
- \*.mpg



## Software FastVDO

- Añade un CODEC a la aplicación Windows Media Player para visualizar los ficheros codificados con el H.264.
- Identifica los ficheros H.264 con la etiqueta \*.264
- Página de descarga: <http://www.fastvdo.com/H.264.html>. Es una versión demo limitada a 5 mins.



Windows Media Player

# SOFTWARE DE REFERENCIA JM

- **Software de referencia JM** (<http://iphone.hhi.de/suehring/tml/download>).
- Desarrollado en C y C++ implementa el estandar H.264.
- Incluye codificador y decodificador.
- La versión compilada para Windows y Linux se encuentra en:  
`..\software\encoderDecoderH.264\H264exe`
- Codificador: **EncodeH264MSWin.exe**
- Decodificador: **DecodeH264MSWin.exe**



Fichero de  
configuración



Video de entrada

**EncodeH264MSWin.exe**  
Software JM

Fichero binario  
H.264



Video reconstruido.  
Idéntico al video generado  
por el  
decodificador

## cmd) EncodeH264MSWin.exe -f configuracion.cfg

**InputFile** = "..\..\..\videos\mobile@300@352x288.yuv" # Input sequence

...

**StartFrame** = 0 # Start frame for encoding. (0-N)

**FramesToBeEncoded** = 300 # Number of frames to be coded

**FrameRate** = 20.0 # Frame Rate per second (0.1-100.0)

**SourceWidth** = 352 # Frame width

**SourceHeight** = 288 # Frame height

...

**ReconFile** = "mobile@300@352x288QP15.yuv"

**OutputFile** = "mobile@300@352x288QP15.h264"

...

**QPISlice** = 15 # Quant. param for I Slices (0-51)

**QPPSlice** = 15 # Quant. param for P Slices (0-51)

.....

**QPBslice** = 30 # Quant. param for B slices (0-51)