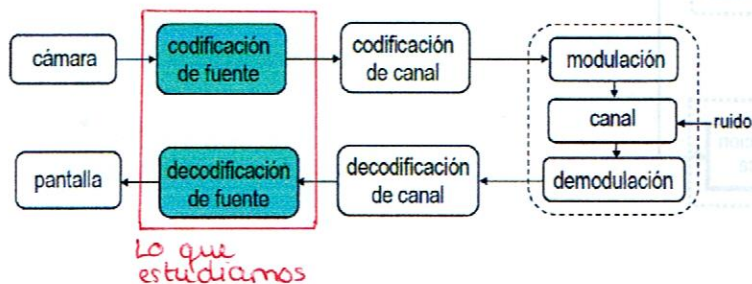


PARTE 3. COMUNICACIONES VISUALES

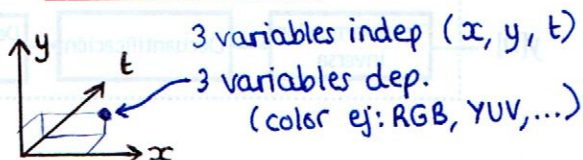
1. INTRODUCCIÓN

Sistema de codificación digital



problema: tasas elevadas
solución: compresión

Señal de vídeo



YUV: Y: luminosidad
U, V: componentes de color

- mucho mejor adecuado al ojo humano que R, G, B

- se emplea en PAL (analógico)

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Digitalización

$(x, y) \rightarrow N_x$ y N_y píxeles

$t \rightarrow f_T$ cuadros/s

color $\rightarrow N_b$ bits/píxel



Tasa binaria total $R = N_x N_y f_T N_b$ [bits/s]

• Norma ITU-R BT.601 $\rightarrow$ Digitalización de vídeo NTSC y PAL

USA/Japón $\rightarrow$ 525/60 : 720 x 480 píxeles/cuadro, 30 cuadros/s
Europa $\rightarrow$ 625/50 : 720 x 576 píxeles/cuadro, 25 cuadros/s $\rightarrow$ misma tasa de píxeles/s

En ambos: $N_b = 8$ bits/píxel

Color (Y, Cb, Cr)

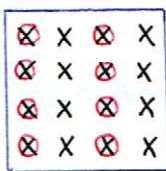
$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 16 \\ 128 \\ 128 \end{pmatrix} \rightarrow Y \in [16, 240] \leftarrow \text{valores límite para sinc}$$

• Muestreo de la crominancia: (X = muestra Y, O = muestra Cb, Cr)

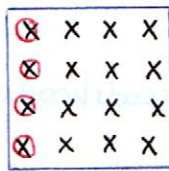
4:4:4



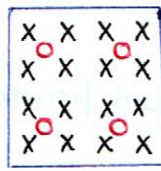
4:2:2



4:1:1



4:2:0



el que mas se usa en aplicaciones habituales
TDT, DivX, MPEG2, ...

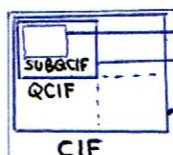
• CIF, QCIF, SubQCIF (Common Intermediate Format)

Formato común para videoconferencia USA/Europa

Y Resolución horizontal: $720 \xrightarrow{\div 2} 360 \rightarrow 352$ (múltiplo de 16)
Resolución vertical: $576 \xrightarrow{\div 2} 288$
Resolución temporal: 30 Hz } los mejores valores de cada uno

Cb
Cr
4:2:0

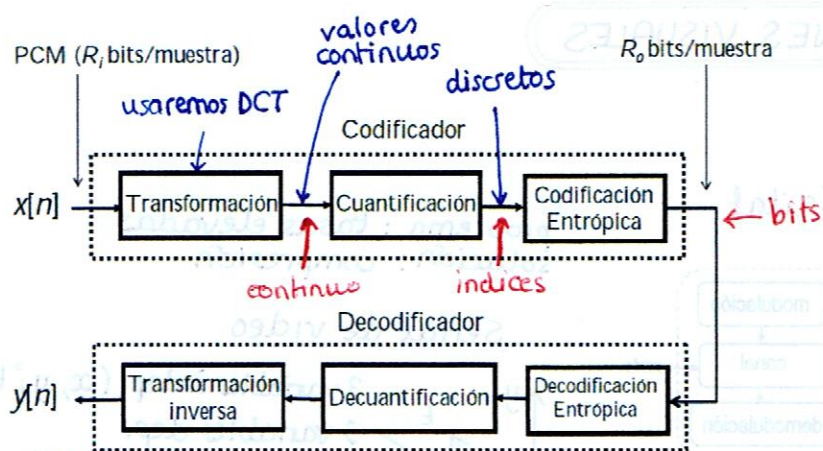
Y



4-CIF

SUBQCIF
QCIF
CIF

Codificación de forma de onda



Parámetros de la compresión

- tasa o factor de compresión
- distorsión/calidad
- retardo o latencia (complejidad) → es importante (ej. videoconferencia) o no (ej. película, TDT, ...) según la aplicación.

Medida de la distorsión

- Medidas objetivas:

$$MSE = \frac{1}{N} \sum (y[n] - x[n])^2$$

$$SNR = 10 \log \frac{P_x}{MSE}$$

← potencia de la imagen

$$PSNR = 10 \log \frac{255^2}{MSE}$$

← buena correlación con MOS excepto en casos especiales (ej. línea blanca de 1 pixel)

↑ peak

- Medidas subjetivas:

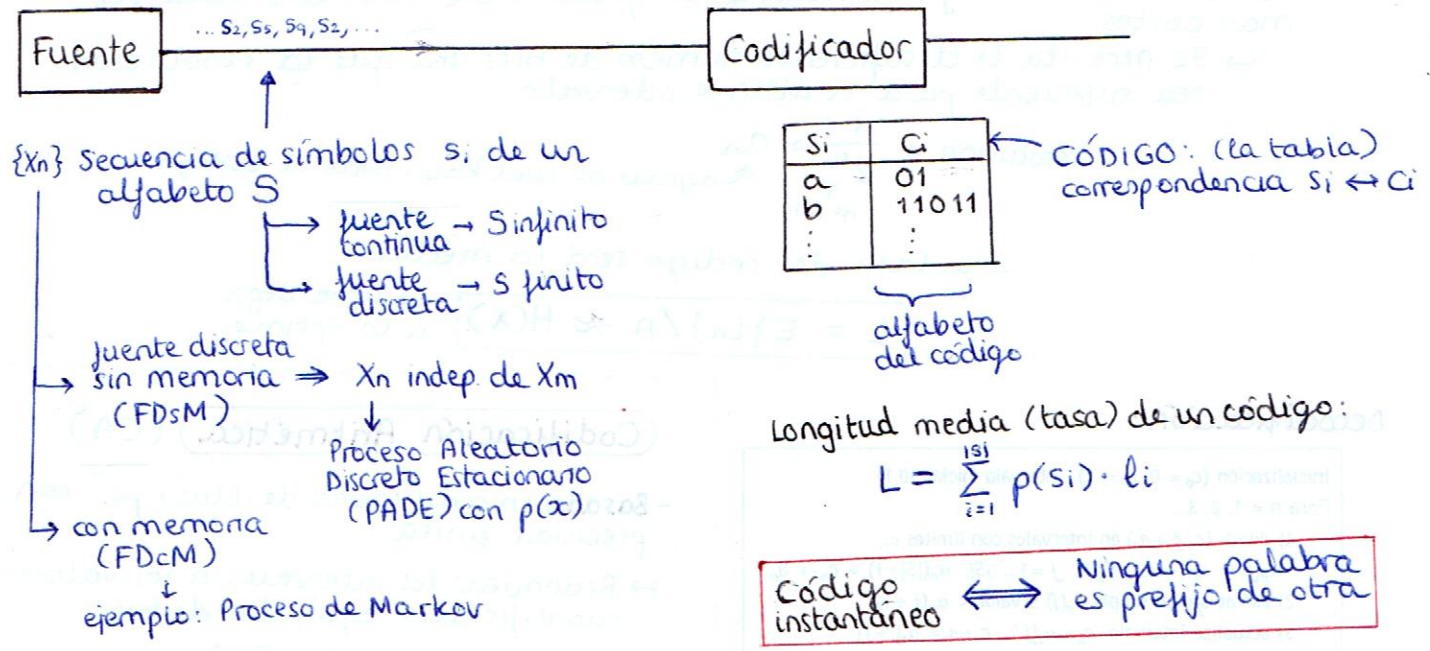
Puntuación de opinión media (MOS)

Complejidad

Simetría codificador/decodificador con/sin pérdidas

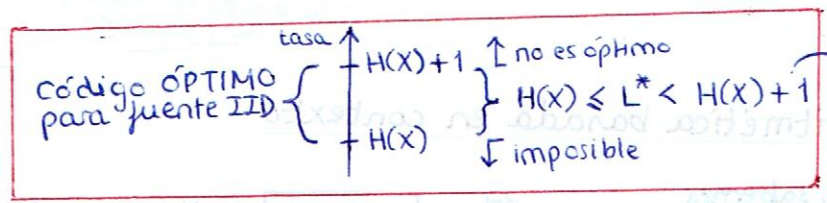
2. CODIFICACIÓN ENTRÓPICA

Transformar índices k en bits para minimizar la tasa



Códigos óptimos

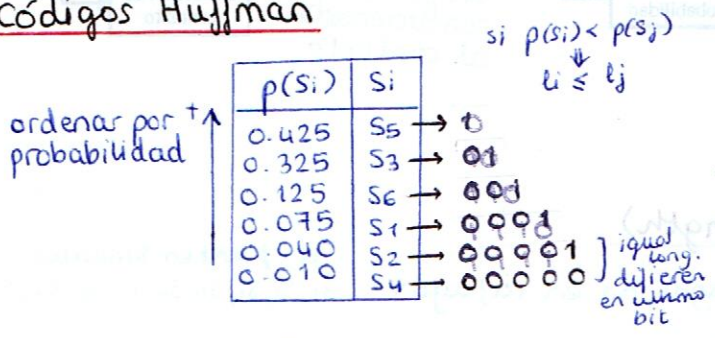
No se puede codificar X sin error a una tasa $R < H(X)$ (entropía de la fuente [bit/simb])



Es ineficaz si $H(X) \ll 1$
 Solución: extensión de fuente orden M
 $H(X) \leq L^* < H(X) + 1/M$

si $M \rightarrow \infty$
 $L^* \rightarrow H(X)$
 Problema: 10^{10} palabras

Códigos Huffman



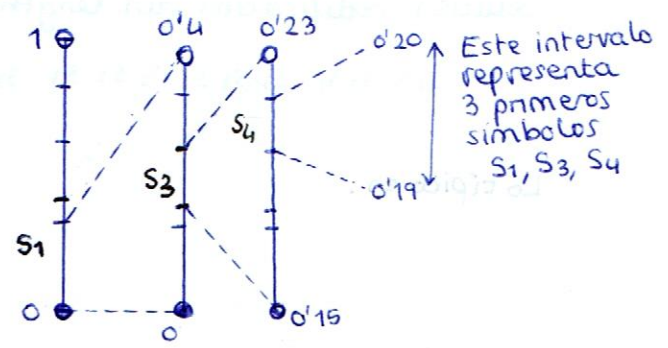
Lo típico es:

- codificador simple → usar Huffman
- codificador puede permitirse complejidad → usar codificación aritmética

Codificación de Elias

- irrealizable en la práctica
- En el paso i se tiene el intervalo $[C_i, C_i + a_i]$
- El símbolo S_i determina que subintervalo escoger para el paso $i+1$
- se divide en subintervalos proporcionales a la probabilidad de cada símbolo.

ejemplo:



Propiedades Codificación de Elias: - Los intervalos son disjuntos

- Cualquier número real en $[c_n, c_n + a_n[$ es un código de $X_0:n$

- Problema: un mensaje de mucha longitud tiene intervalos cada vez mas cortos

→ Se necesita tx el suficiente número de bits tal que la resolución sea suficiente para el último intervalo

$$\text{resolución} = \frac{1}{2^{L_n}} \approx a_n$$

↑
nº bits

← depende de cual haya sido el código

- la tasa del código será la media:

$$L = E\{L_n\}/n \approx H(X)$$

se logra llegar a lo óptimo

Decodificación:

Inicialización ($c_0 = 0, a_0 = 1$). Intervalo inicial: $[0, 1)$

Para $n = 1, 2, 3, \dots$

1) divide $[c_n, c_n + a_n)$ en intervalos con límites α_n :

$$\alpha_n(j) = c_n + a_n \cdot F(j), \quad j = 1, \dots, |S|; \quad \alpha_n(|S|+1) = c_n + a_n$$

2) extrae s_j que cumple $\alpha_n(j) \leq \text{valor} < \alpha_n(j+1)$

3) actualiza intervalo: $c_n = \alpha_n(j), \quad c_n + a_n = \alpha_n(j+1)$

¿cómo sabemos cuando acabar?

- número fijo de símbolos
- código de terminación

Codificación Aritmética (CA)

- Basado en codificación de Elias pero con precisión finita

→ Redondear los intervalos a los valores cuantificados según el nº de bits

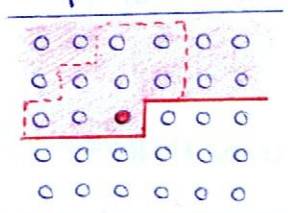
$$\text{i.e. } 0 < a_{n+1} \leq \frac{a_n \cdot p(x_n)}{2}$$

tomar valor redondeado ej 0.5 ej: 0.501213

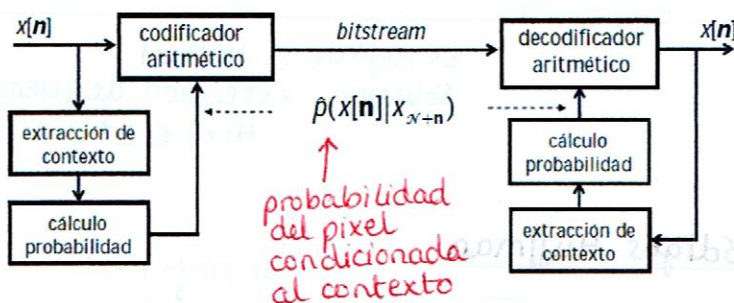
- Los intervalos siguen siendo disjuntos
- se pierde eficacia pero la pérdida es despreciable

Codificación aritmética basada en contexto

Al codificar un pixel, sabemos cuales son los anteriores, lo cual permite refinar la probabilidad del valor de cada pixel



se puede modelar con proceso de Markov



Codificación de ráfagas (run-length)

- En algunas fuentes los símbolos se agrupan en ráfagas
- CA requeriría contexto muy grande

ej: fuentes binarias

$S_1, S_1, S_1, S_0, S_0, S_0, S_0, S_1, S_1$

solución: codificación run length: parejas (k, r_k)

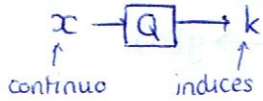
↑ símbolo ↑ nº de apariciones

$S_1 S_1 S_1 S_1 \quad S_3 S_3 S_3 S_3 S_3 S_3 \quad S_2 S_2 S_2 \rightarrow (1, 4) (3, 6) (2, 3)$

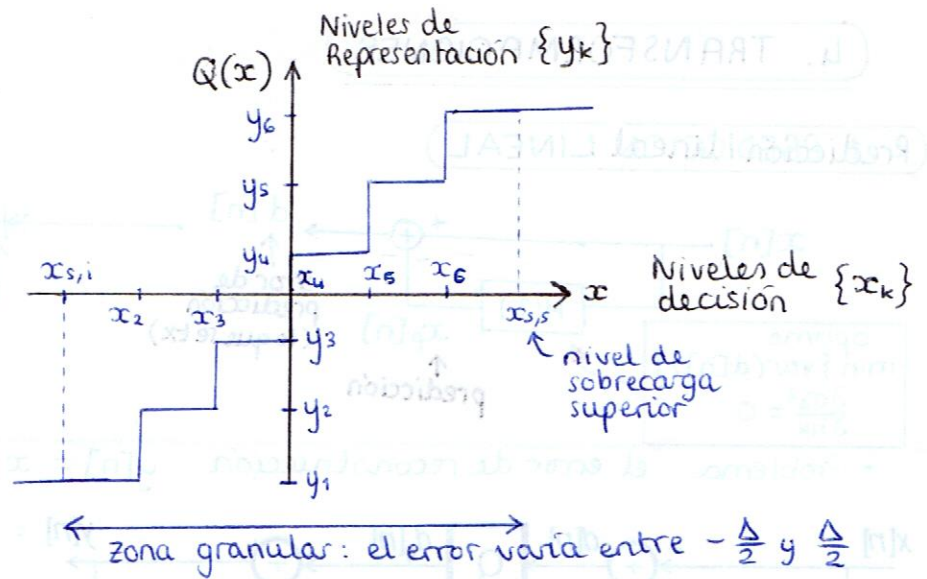
codificar parejas con Huffman, CA, ...

3. CUANTIFICACIÓN

Discretizar una fuente continua

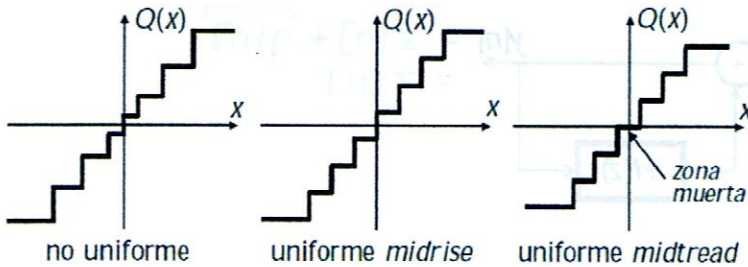


- La cuantificación es un proceso **NO REVERSIBLE**
- La codificación entrópica siempre es reversible
- La transformación es reversible (si utiliza precisión infinita)

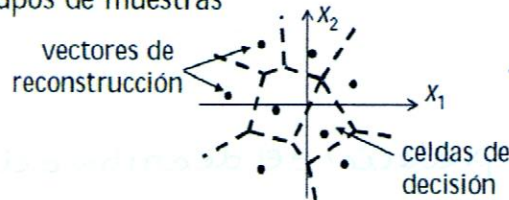


Tipos de cuantificación

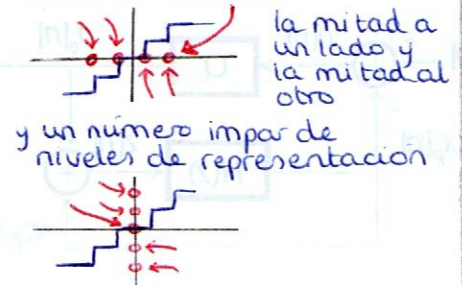
- Escalares: cuantifican cada muestra individualmente



- Vectoriales: cuantifican grupos de muestras



Midtread tiene un n° PAR de niveles de DECISION



Ventajas:

ej: en lugar de cuantificar por separado peso y altura de la gente, cuantificar a la vez para centrarse en zonas frecuentes

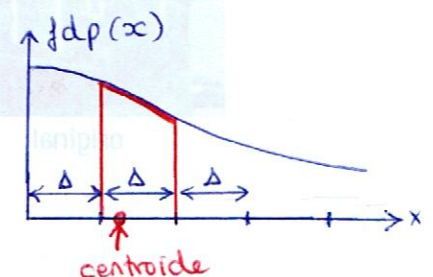
Tasa fija/variable

- Cuantificador de tasa fija: CLF: $R = \lceil \log_2 N \rceil$ ← interesante si los niveles son equiprobables
- Cuantificador de tasa variable: CLV

Suponiendo que $R \approx H(Y)$: obtener los $\{x_k, y_k\}$ que $\min E[(x-y)^2]$

- El uniforme con $N = \infty$ es óptimo para altas tasas
- A bajas tasas, el uniforme no es óptimo, pero:

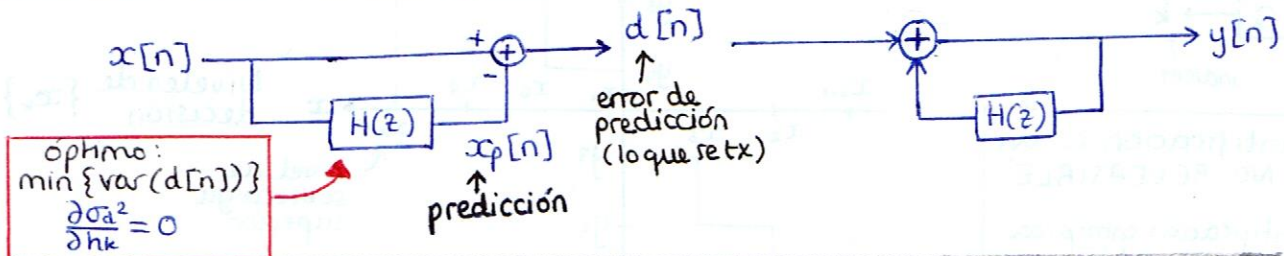
→ Cuantificador casi-óptimo:
intervalos de decisión uniformes Δ
con niveles de reconstrucción en centroide
valor medio de la variable en el intervalo



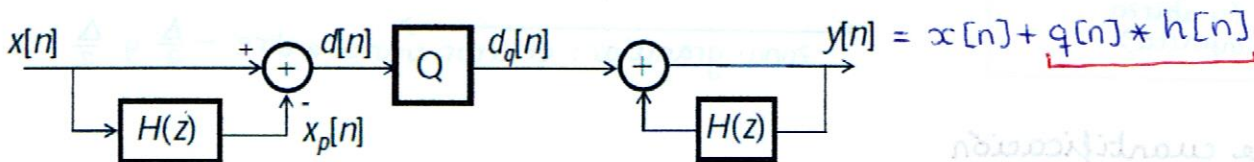
- si f_{dp} nula, a menudo mejora si se amplía el intervalo en cero (zona muerta)
- índice más probable → baja entropía de la fuente

4. TRANSFORMACIONES

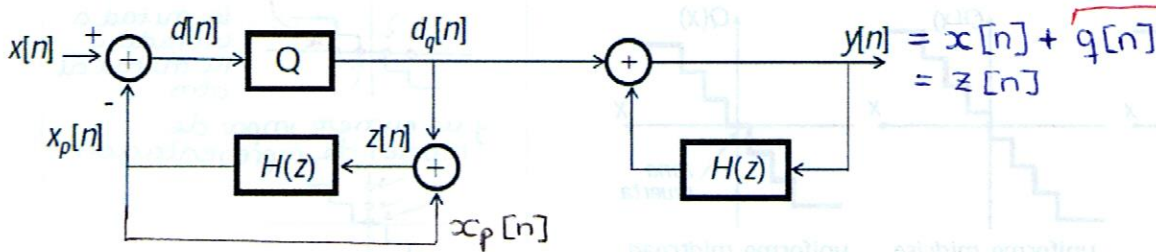
4.1 PREDICCIÓN LINEAL



- Problema: el error de reconstrucción



Solución: predicción en bucle cerrado

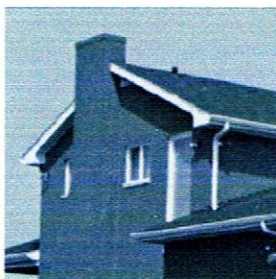


Predicción conmutada

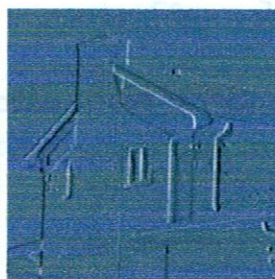
¿qué pixel empleamos para predecir? ¿el de arriba o el de la izquierda?

- Predicción adaptativa (coeficientes óptimos)
- Predicción conmutada

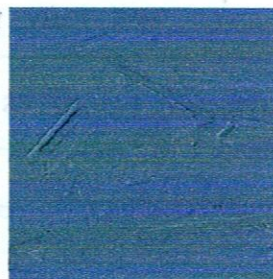
$$x_p = \begin{cases} x_C, & \text{si } |x_C - x_A| < |x_B - x_A| \\ x_B, & \text{si } |x_C - x_A| \geq |x_B - x_A| \end{cases}$$



original



error ($x_p = x_B$)



error (p. conmutada)

4.2 TRANSFORMADAS DE BLOQUE 1D



Si funciones base ortonormales, la energía puede calcularse tanto en tiempo como en frecuencia:

$$\sum_{k=0}^{N-1} |C_k|^2 = \sum_{n=0}^{N-1} |x[n]|^2$$

Además, matricialmente

$$\underline{C} = \underline{B} \cdot \underline{x}$$

$$\underline{x} = \underline{B}^T \cdot \underline{C}$$

con $\underline{B} = \begin{bmatrix} b_0[0] & b_1[0] & \dots & b_{N-1}[0] \\ b_0[1] & & & \\ \vdots & & & \\ b_0[N-1] & & & \end{bmatrix}$

$$x[n] = \sum_{k=0}^{N-1} C_k \cdot \underbrace{b_k[n]}_{\text{funciones base}}$$

$$C_k = \sum_{n=0}^{N-1} x[n] \cdot \frac{b_k[n]}{\sqrt{D_k}}$$

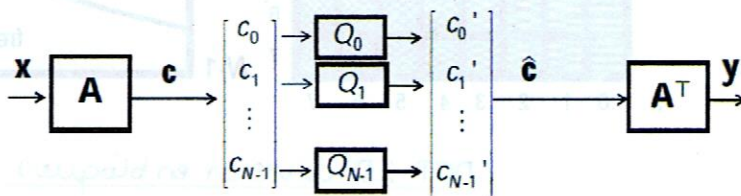
no está conjugado porque suponemos coef. reales

conjunto ortogonal

$$\sum_{n=0}^{N-1} b_k[n] \cdot b_l[n] = D_k \delta_{kl}$$

ortonormal si $D_k = 1$

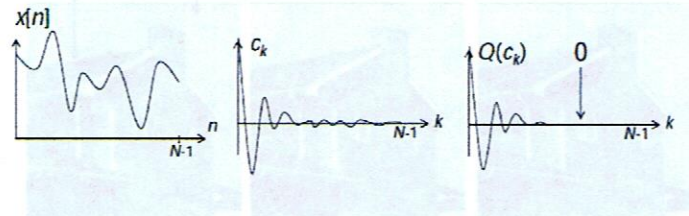
- CODEC: podemos codificar cada coeficiente frecuencial con cuantif. distinto



- Posible gracias a la decorrelación entre coeficientes

- Compactación de la energía

- La energía se concentra en los coeficientes de baja frecuencia
- coef. de alta frec. tienen valores muy pequeños
- Pueden anularse, introduciendo un error pequeño



- Ejemplos de Transformadas Criterios: compactación energía, decorrelación, coste

Adaptativas: Karhunen-Loeve (KLT) → Es la transformada óptima. Cambia para cada imagen. Elevado coste

Fijas:

- Haar (HT)
- DFT
- Walsh Hadamard (DWT)
- Discrete cosine (DCT)

Baja complejidad. Sólo son sumas y restas

$$\underline{A}_{2 \times 2} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \underline{A}_{N \times N} = \frac{1}{\sqrt{2}} \begin{bmatrix} A[N/2] & A[N/2] \\ A[N/2] & -A[N/2] \end{bmatrix}$$

$$C_k = \sqrt{\frac{2}{N}} \alpha_k \sum_{n=0}^{N-1} x[n] \cdot \cos \frac{(2n+1)k\pi}{2N}$$

$$x[n] = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} C_k \cdot \frac{\cos(2n+1)k\pi}{2N}$$

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

- Tiene algoritmo rápido
- Es óptima si $x[n]$ es un AR(1) con $\rho \rightarrow 1$

AR(1) puede modelarse como ruido blanco pasado por filtro todo polos con único polo

$$y[n] = b_0 x[n] + a_1 y[n-1] \rightarrow H(z) = \frac{b_0}{1 - a_1 z^{-1}}$$

$$\rho = \frac{R_x[0]}{R_x[1]} = 1$$

- Se usa en JPEG, H.26x, MPEGx

4.3 TRANSFORMADAS DE BLOQUE 2D

Transformada 2D

$$c_{k,l} = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x[n,m] b_{k,l}[n,m]$$

$$x[n,m] = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} c_{k,l} b_{k,l}[n,m]$$

Transformada 2D separable

$$\text{Separable} \iff b_{k,l}[n,m] = b_k[n] \cdot b_l[m]$$

→ menor coste

$$O(N^4) \rightarrow O(N^3)$$

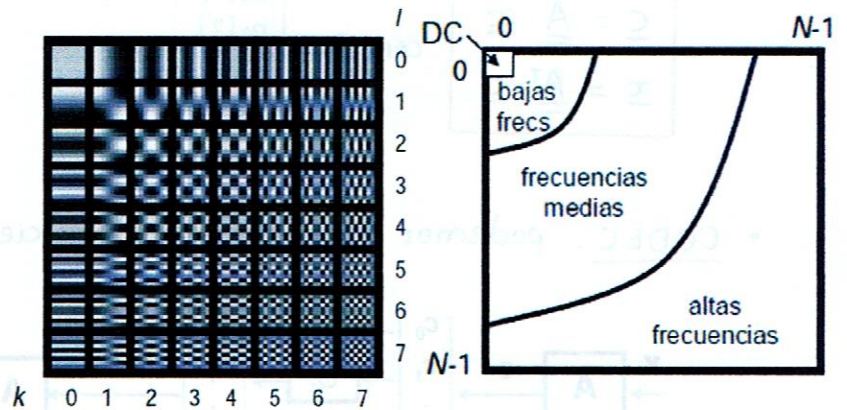
cada uno de los $N \times N$ coeficientes requiere $N \times N$ productos

se puede aplicar primero a las filas y luego a las columnas

DCT-2D

$$b_{k,l}[n,m] = b_k[n] \cdot b_l[m]$$

$$b_k[n] = \sqrt{\frac{2}{N}} \alpha_k \cos \frac{(2n+1)k\pi}{2N}$$



DCT-2D Compactación de la energía



	50 %	25%	10%
Coefs (%)	50	25	10
PSNR (dB)	37.2	32,3	29,0

DCT-2D división en bloques

Se divide la imagen en bloques de $N \times N$ píxeles ($N=8$) para menor complejidad / memoria

N mayores no mejoran la compactación significativamente

Problema: A bajas tasas efecto de bloques

- Discontinuidades en los bordes de los bloques
- Alta visibilidad

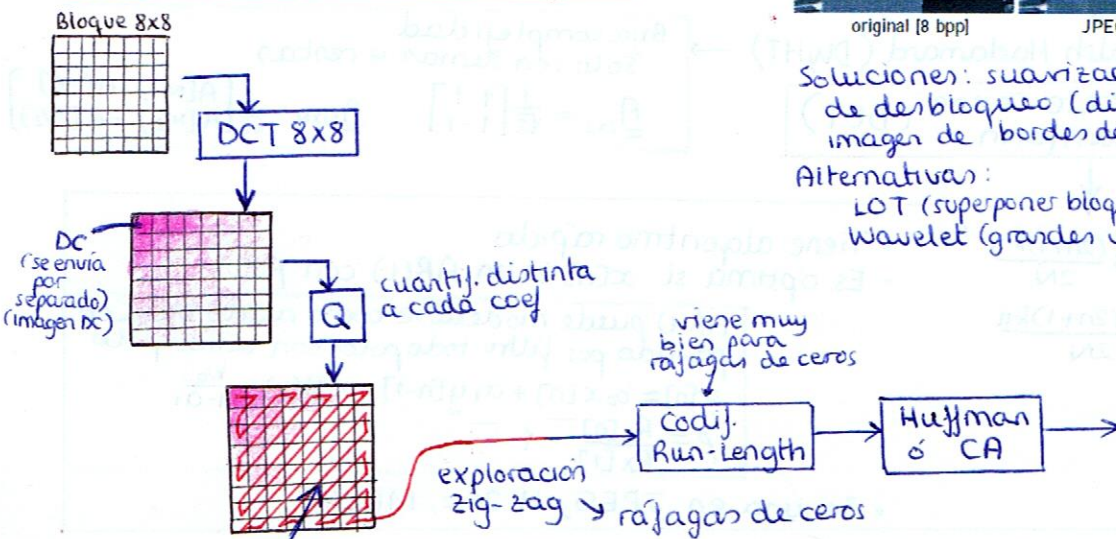


Soluciones: suavizado local, algoritmos de desbloqueo (distinguir bordes de la imagen de bordes de bloques)

Alternativas:

LOT (superponer bloques 16×16 aunque se calculan 8×8)
Wavelet (grandes ventajas) (JPEG 2000)

DCT-2D codificación de imágenes



PARTE 4. COMPRESIÓN DE VÍDEO

- Ya hemos hablado de redundancia dentro de un mismo cuadro (intra-frame)
- Ahora hay que eliminar la redundancia temporal

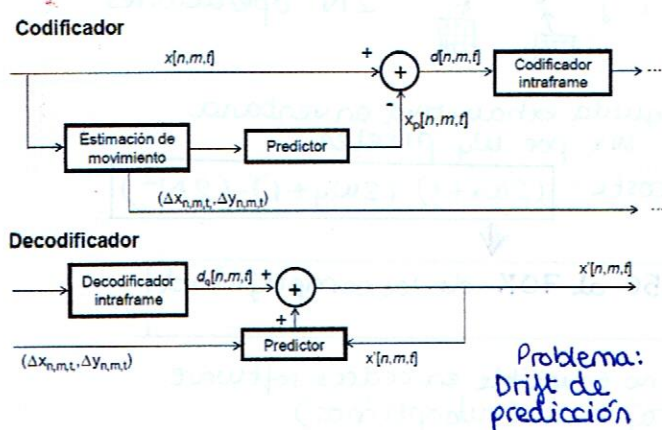
↳ Transformadas 3D (DCT-3D y Wavelet-3D) → está de moda
 ↳ Esquema híbrido

- 1) Predicción compensada en movimiento $x_p[n,m,t] = x[n+\Delta x, m+\Delta y, t-\Delta t]$
- 2) Error de predicción $d[n,m,t] = x - x_p$ ($\Delta x, \Delta y$) vector de movimiento
- 3) Codificación intraframe del error de predicción

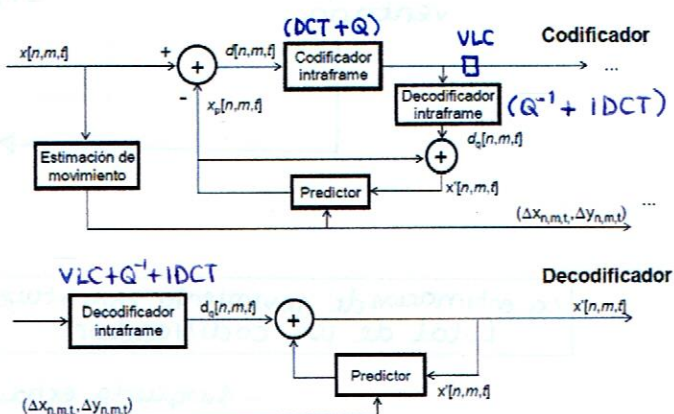
4.1 ESTIMACIÓN DE MOVIMIENTO

- Gran redundancia temporal en la señal de vídeo
 - movimiento de objetos y cámara: fácil
 - regiones descubiertas y cambio de luminosidad: fastidioso

Codec híbrido con bucle abierto:



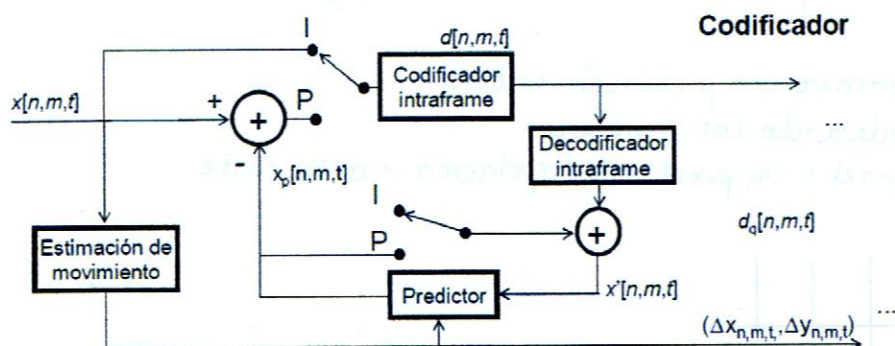
Codec híbrido en bucle cerrado:



- El decodificador usa la misma info que el codificador para predecir
- El codificador incluye un decodificador (con todo excepto el codificación códigos de long. variable VLC que no tiene pérdidas)

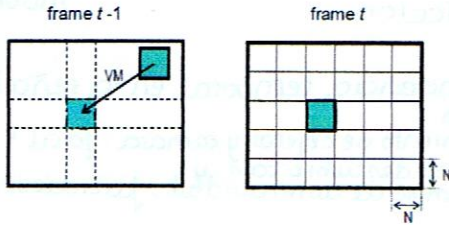
Cuadros Intra(I) e Inter(P)

- dependen de cuadros anteriores
 - solo codificación intraframe: resetear la predicción
 - ventajas: inmunidad a errores de tx, acceso aleatorio
- ... I P P P P P P P I P P P P P P P I P P P P P ...



Block Matching Algorithm (BMA)

- Se divide el frame actual en bloques de $N \times N$ píxeles.
- Para cada bloque se busca el vector de movimiento (VM) que lleva a un grupo de $N \times N$ píxeles con gran parecido al bloque actual en el frame anterior.



Búsqueda exhaustiva: coste ↑
Solución: limitarse a una ventana.

medida de la semejanza entre dos bloques:

SAD (Sum of Absolute Differences)

$$SAD = \sum_i \sum_j |f_i - g_j|$$

$N \times N$ grande ej 16×16
• tiene mas error de predicción a tx

$N \times N$ pequeño ej 4×4
• error de predicción muy bajo

• Debe tx muchos vectores de movimiento (uno por bloque)
¿merece la pena?

Lo ideal es cambiar dinámicamente el tamaño de bloque (codif. última generación)

Coste:
 $2N^2$ operaciones

Búsqueda exhaustiva en ventana de w_x por w_y píxeles

coste: $(2w_x + 1) \cdot (2w_y + 1) \cdot (2N^2)$

La estimación de movimiento constituye del 50 al 70% de la complejidad total de un codificador

- Búsqueda exhaustiva no asumible en codecs software
- Algoritmos de EM rápidos (subóptimos)

Estimación de movimiento (EM) subpíxel

Los vectores de movimiento (VM) reales no están limitados a la rejilla de píxeles, por tanto la EM será mas precisa si se estiman los VM con precisión fraccionaria

precisión $1/2$ → ganancia significativa

$1/4$ → ganancia moderada

$1/8$ → ganancia menor

últimos codecs usan $1/4$ en luminancia (lo que supone $1/8$ en crominancia)

Nota: a veces se calcula en el dominio transformado SADD

→ Solución 1:

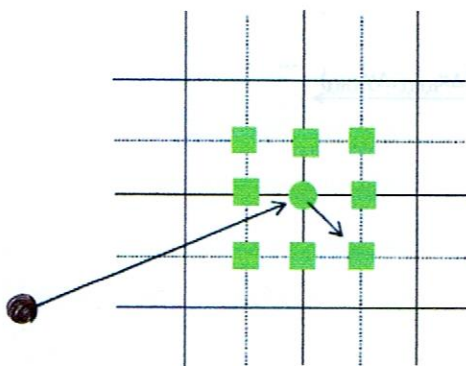
- Interpolan píxeles por M (aumenta memoria)

→ Solución 2:

- Estimar VM primero con precisión entera

- Refinar considerando los 9 vecinos

↳ a $1/2$ píxel o $1/4$ píxel → interpolación → alto coste



4.2 ESTÁNDARES DE VÍDEO

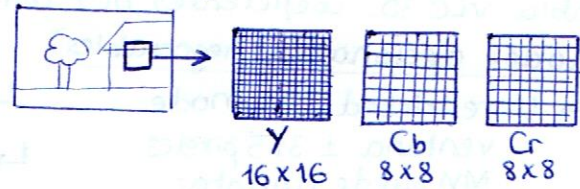
H.261

- Estandar ITU (1990)
- Videoconferencia por RDSI
tasa $p \times 64 \text{ kbps}$ ←
↑ $p = 1, 2, \dots, 30$ —no entrelazado

- Formatos: CIF y QCIF (4:2:0)
- Esquema: DCT + predicción compensada en movimiento (opcional)
- Imágenes Intra (I) e Inter (P)
- Compresión intra-frame:
DCT (8x8) + Q uniforme
+ Zigzag + Runlength + VLC

Predicción:

- Cada frame se divide en Macro Bloques (MB)



- Un vector de movimiento para cada MB
↳ precisión: 1 pixel
↳ ventana: ± 16 píxeles
- Codificación diferencial de los VM
- Filtro de bucle: la imagen que se utiliza para predecir esta filtrada tanto en codif. como de codif. Trata de eliminar bordes de bloques para una mejor predicción (que no aparezcan los bordes trasladados por el VM)
Empeora matemáticamente la SNR pero mejora la calidad subjetiva

MPEG-1 También llamado Video CD (VCD)

- Estandar ISO-IEC (1991)
- Objetivo: CDRom ~ 1 película (sustituir VHS → cintas)
↳ necesita acceso aleatorio
↳ pasar rápido, etc...

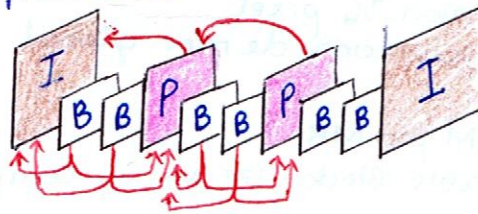
- Formato:
 - no entrelazado
 - CIF → líneas similares VHS
 - 24...30 fps

- Imágenes I, P y Bidireccionales (B)

- Predicción compensada en mov:
 - 1 vector por MB en P
 - 1 ó 2 vectores por MB en B
↳ anterior y siguiente
 - Precisión $\frac{1}{2}$ pixel (interpolación bilineal)
 - Codif. diferencial de las componentes
 - No incluye filtro de bucle (H.261 y H.264 sí)

Imágenes B

- utilizan un VM para imagen I o P anterior y otro para imagen I o P posterior por cada MB (ponderado según distancia temporal)
- Las imágenes B NO se usan para predicción



MPEG-2 → ej: TDT, HDTV, DVD

- Estandar ISO-IEC (1991-97)
- Objetivo: radiodifusión TV estándar y HDTV
- Multiplex: varios programas simultáneos → suaviza la tasa
- Capacidad para manejar video entrelazado
- Tasas 2 a 15 Mb/s (DVD bueno 8 Mb/s)
- Escalabilidad: La calidad va por capas para que bajas capas se ajusten a usuarios de baja tasa
 - escalabilidad de resolución
 - escalabilidad temporal
 - escalabilidad de SNR

- **H. 263** muy bajas tasas, videotelefonía móvil actual
 - 5 formatos (subQCIF, QCIF, CIF, 4CIF, 16CIF)
 - Estandar ITU (1995)
 - logra tasa -30% que MPEG-1
- Compensación de movimiento:
 - 1 VM por MB
 - máximo desplazamiento ± 16 pixels
 - $\frac{1}{2}$ pixel
 - sin filtro de bucle
- Tabla VLC 3D, coeficientes DCT con (last-run-level) $\rightarrow$ bit indica si es el último par de un bloque
- Modos opcionales (negociables)

$\rightarrow$ Unrestricted VM mode

- ventana ± 31.5 pixels
- VM puede apuntar FUERA de la imagen



se rellenan repitiendo la frontera

$\rightarrow$ Arithmetic Coding Mode

$\rightarrow$ Advanced Prediction Mode

- 4 VM por MB
- Overlapped Motion compensation
- VM pueden apuntar fuera de imagen

$\rightarrow$ PB-Frame Mode

Imágenes B codificadas indisolublemente junto a las P en la trama



$\uparrow$
tramas H.263 y MPEG-4 son compatibles si se eliminan los modos opcionales

- **MPEG-4**
 - Cine, TV, internet, videotelefonía
 - ej: DivX, XviD

- Estándar ISO-IEC (1998)

- Objetivo: todo (gran rango de aplicaciones)

$\rightarrow$ video progresivo y entrelazado

- Formatos: 4:2:0, 4:2:2, 4:4:4

- Alta compresión, escalabilidad, robustez frente a errores

- Compensación de movimiento

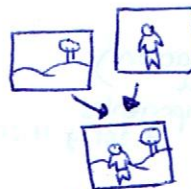
- Precision $\frac{1}{4}$ pixel
- Compensación de mov. global (útil en panning o tilt de toda la imagen) (vector global que se suma a los de cada bloque)
- 4 VM por MB (Variable Block Size Motion Compensation)

• Objetos de video

- Segmentación en objetos de video

- VOP (video object plane)

= VO en un cierto instante



AVC

δ MPEG-4 part 10
δ H.264

H.263 → H.263+
H.263++ → H.264

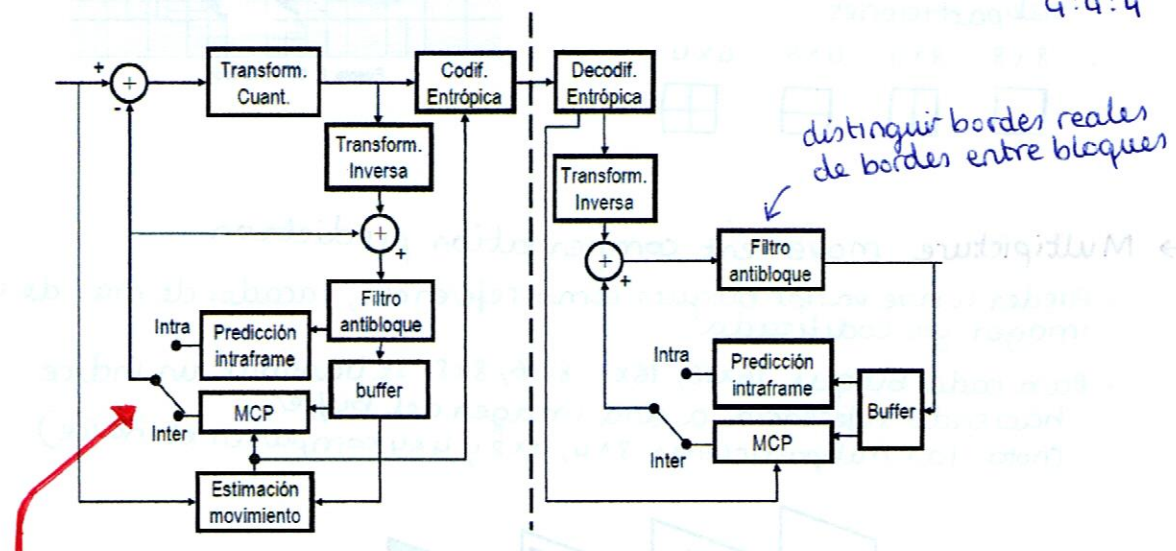
ITU-T : VCEG (Video Coding Experts Group)
ISO : MPEG (Motion Picture Expert Group) } → 2001 Joint Video Team

- Todo tipo de aplicaciones (permite video entrelazado)
- Significativa mejora de la eficiencia de codificación

Diagrama de bloques

Generalidades:

YUV muestreo 4:2:0
4:2:2
4:4:4



SIEMPRE se hace predicción

Imágenes Inter: Predicción con compensación de movimiento
↳ como siempre pero nuevas cosas: Tree-structured skip

Imágenes Intra: Predicción espacial (nuevo en este estándar!!)
además de las transformadas de siempre (mejoradas)

Predicción Intra (redundancia espacial)

Dos tipos:

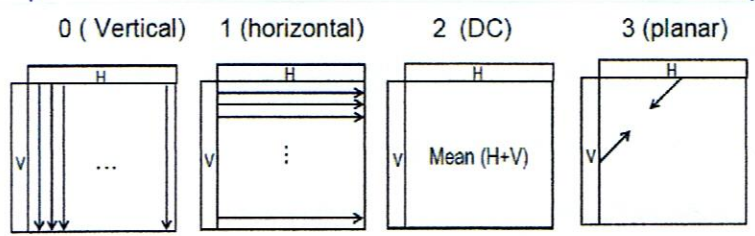
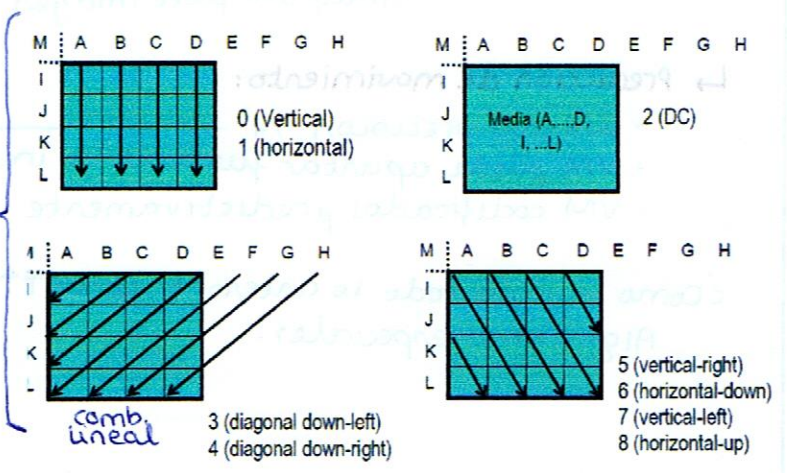
↳ Intra 4x4 (Luma)

- El MB se divide en 16 subbloques 4x4
- Predicción de píxeles ya codificados → 8 modos

(El estándar PERMITE todo esto, pero no especifica cuándo usarlo)

↳ Intra 16x16 (Luma)

↳ todo el MB
↳ 4 modos



Planar: función lineal entre muestras de H y V

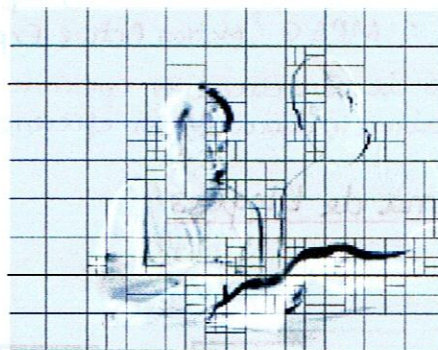
La croma Cb y Cr hacen algo similar al 16x16

- Predicción imágenes Inter (Frames Tipo P)

↳ Tree structured movement compensation

- Partir MB en
 16×16 16×8 8×16 8×8

- A su vez el 8×8 puede tener subparticiones:
 8×8 8×4 4×8 4×4

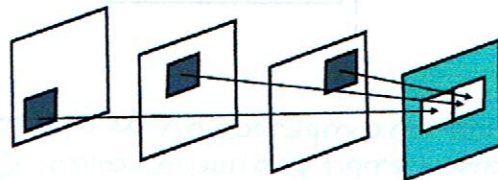



Predicción de movimiento de bloques con tamaño adaptativo

Fuente: E. G. Richardson

↳ Multipicture movement compensation prediction

- Pueden usarse varios bloques como referencia, sacados de mas de una imagen ya codificada
- Para cada bloque 16×16 , 16×8 , 8×16 , 8×8 se transmite un índice haciendo referencia a una imagen del buffer (nota: las subparticiones 8×4 , 4×8 y 4×4 comparten el índice)



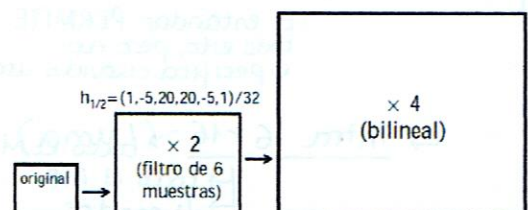
- Permite predicción ponderada (muy útil para fundidos de una imagen a otra)

↳ Modo skip: se envía sólo el VM, no el error de predicción, de forma que el bloque se copia directamente del origen (muy útil para imágenes quietas o barrido)

↳ Predicción de movimiento:

- VM con precisión $1/4$
- VM puede apuntar fuera de la imagen (extrapolación)
- VM codificados predictivamente

¿Cómo se logra todo lo anterior? Corte ↑↑
Algoritmos especiales:



• AVC: Filtro antibloqueo → reduce tasa 5-10%

- Se aplica a los MB decodificados
- Reduce el efecto de bloques, mejora la calidad subjetiva
- Se aplica exclusivamente sobre contornos verticales y horizontales
- Si la frontera tiene diferencias mayores a las esperadas por el efecto de bloques, no se filtra la frontera



- Codificación de residuos → error de predicción → transformada (una o dos!) → zig-zag + codificación entropica de los coeficientes

Transformada de bloque:

- se usa similar a la DCT 4x4 

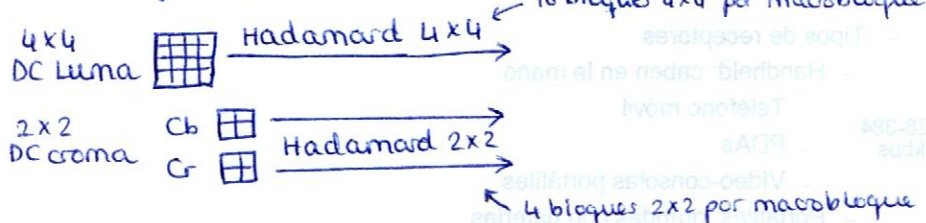
$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

(en realidad es separable y se aplica por filas y columnas)

- sumas/restas/desplazamientos/coef. enteros
- menor coste computacional
- menor correlación en el residuo
- menor ruido mosquito

no hay errores de redondeo

- Además, en el tipo Intra 16x16, al resultado de la DCT se le aplica una NUEVA transformada para comprimir aun más la información



Codificación entropica:

- Se usa codificación entropica adaptativa (estadística no estacionaria)
- 3 métodos:
 - UVLC (Universal VLC) ← variable length coding
 - CAVLC (Context Adaptive VLC)
 - mas rápido pero comprime menos
 - CABAC (Context-Based Adaptive Binary Arithmetic Coding)
 - mas costoso, utiliza codificación aritmética

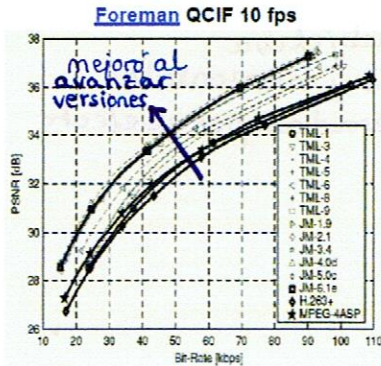


128-384 Kbps
25-50 stream video H.264
pantalla pequeña

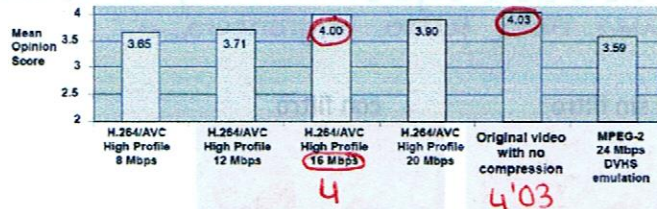


4-8 Mbps
3-4 programas
pantalla grande

AVC: resultados video



- Comportamiento perfil High
- Mejoras de calidad subjetiva
- Evaluación Blue-Ray Disk Association (BDA) [4]



- Calidad superior a MPEG-2 24 Mbps con 8 Mbps
- Calidad indistinguible del original con 16 Mbps
- Nivel 3 adecuado para HD según BDA (< 8 Mbps)

AVC: Productos Hardware y Software

Adopting Bodies and Consortia

- 3GPP and 3GPP2 mobile environments
- ARIB (Japan), DMB (Korea), DAB (Europe), and DVB (Europe) broadcast standards
- AVC Alliance, IMTC, and MPEGIF promotional organizations
- Blu-Ray Disc Association and DVD Forum high-definition storage formats
- IETF audio-video transport real-time protocol payload packetization
- ITU-R broadcast and professional usage standards and ITU-T real-time multimedia conferencing systems
- MPEG storage file format and multiplex system standards
- NATO military specifications, etc.

Productos Hardware

- Mobile phones by Nokia, Samsung, LG, etc.
- Broadcast HD encoders by Harmonic, Modulus, Scientific Atlanta/Cisco, Tandberg TV, Tut Systems, Thomson, etc.
- Direct-broadcast satellite deployments by BSKyB, DirecTV, Dish Network, Euro1080, Premiere, and ProSiebenSat.1.
- All HD DVD and Blu-Ray Disc players
- The Sony Playstation Portable and Apple iPod portable media player devices, etc.

Productos Software

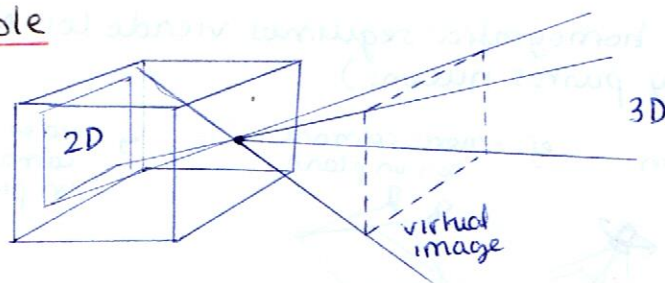
- Media coder/player software packages, including Apple Quicktime, Cyberlink, Nero Digital, etc.
- DSP implementations for various processors
- Hardware accelerators for PC software decoders using graphics processing units (GPUs) by ATI, nVidia, etc.

- Tipos de receptores

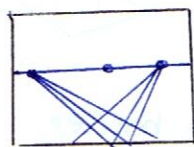
- Handheld: caben en la mano
 - Teléfono móvil
 - PDAs
 - Video-consolas portátiles
- Portátiles: grandes con baterías
 - Pantallas planas TV
 - DVDs con pantalla
 - Ordenador portátil
- Integrados en vehículos: distintos tipos imágenes
 - Información
 - Actualizaciones mapas, tráfico, carreteras

AVC: Ejemplo DVB-H



Modelo pinhole

Dirección en un plano $\leftrightarrow$ punto de fuga
 todo un plano $\leftrightarrow$ línea de horizonte
 (conjunto de puntos de fuga)

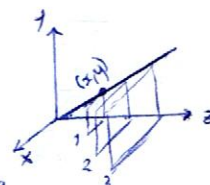
Coordenadas homogéneas

Representar los puntos de un plano bidimensional con 3 coordenadas!

Punto en coordenadas (x, y) 'normales' $\rightarrow$ Punto en coordenadas homogéneas $k \cdot (x, y, 1)$

cualquier valor

en realidad representa el vector de la línea 3D que en pinhole se convierte en un punto



Transformación lineal en coordenadas homogéneas

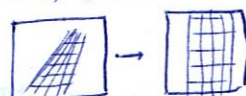
$$\begin{pmatrix} x'_i \\ y'_i \\ z'_i \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}$$

en realidad tiene 8 grados de libertad

ya que $[\dots] = k \cdot [\dots]$

Relación de homografía:

- cómo se ve un plano en una cámara
- Dada la imagen de un plano en una cámara, ¿cómo se vería el plano desde OTRA cámara?



ej: mirar desde arriba

¿Cómo saber si una transformación es HOMOGRAFÍA?

con una homografía seguimos viendo los MISMOS puntos
(no hay puntos nuevos)

ej: zoom



ej: - mover cámara con un plano



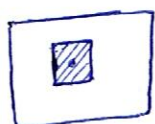
- o cambiar la dirección de apuntamiento

ej: No es homografía mover cámara si no hay sólo un plano

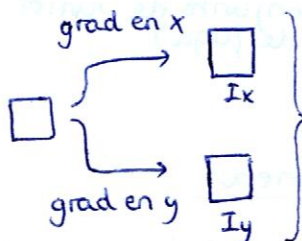


puede ver objetos por detrás!

Detección de bordes



Para cada pixel cogemos una ventana



$$I_x^2$$

$$I_x \cdot I_y$$

$$I_y^2$$

$$C = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

sumatorio en todos los pixels de la ventana

umbral de Harris

Por ser C simétrica, puede convertirse en matriz diagonal sin más que hacer una rotación

$$C = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

si λ_1 y λ_2 son grandes, el pixel es una esquina
si sólo λ_1 o λ_2 es grande, el pixel es un borde

El determinante y la traza son invariantes con la rotación

$$\det C = \lambda_1 \lambda_2$$

$$\text{tr } C = \lambda_1 + \lambda_2$$

por tanto no hace falta efectuar la rotación.
Podemos obtener λ_1 y λ_2 directamente con $\det C$ y $\text{tr } C$



↳ Método Harris:

$$cim = \underbrace{\left(\sum I_x^2 \cdot \sum I_y^2 - (\sum I_x \cdot I_y)^2 \right)}_{\substack{\uparrow \\ \text{filtro gaussiano}}} - k \underbrace{(\sum I_x^2 + \sum I_y^2)^2}_{\text{tr } C}$$

coger cada pixel de cim y sustituirlo por el máximo en un cierto radio

$$sze = 2 * \text{radius} + 1$$

$$\text{numpix} = sze.^2$$

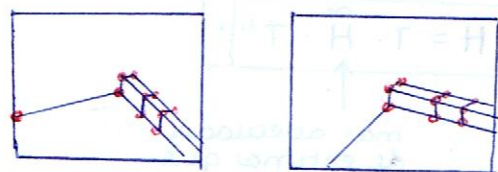
$$mx = \text{ordfilt2}(cim, \text{numpix}, \text{ones}(sze))$$

encontramos máximos locales con

$$cim == mx$$

Estimar homografía

Obtención de parejas



coord. homogéneas

$$x_i \xrightarrow{H} x_i' = H \cdot x_i$$

Una vez eliminados los outliers $\rightarrow$ claramente se cumple:

$$x_i' \times H x_i = 0$$

producto vectorial entre 2 cosas iguales

sistema de 3 ecuaciones (sólo 2 independientes)

por de momento busquemos las 9 incógnitas con 9 ecuaciones
 $\downarrow$
las soluciones podrán tener un factor de escala

Necesitare 4 puntos (8 ecuaciones) para estimar los 8 grados de libertad de H

Sistema

$$A \cdot h = 0 \quad \left. \begin{array}{l} 9 \text{ ecuaciones} \\ \text{con 9 incógnitas} \end{array} \right\} \rightarrow \begin{array}{l} 8 \text{ independientes} \\ 8 \text{ grados de libertad} \end{array}$$

$$\text{rango} = 8$$

solución trivial

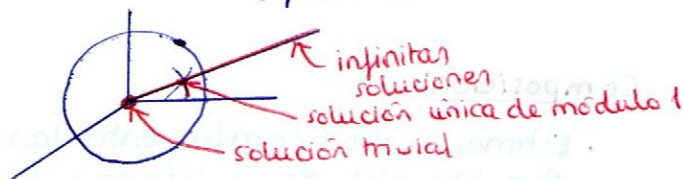
todos los múltiplos de h

soluciones al sistema (infinitas)

Problema: al añadir puntos rango A se hace 9 $\rightarrow$ única solución trivial ($h=0$)

Solución: Exigir $\|h\|=1$

espacio 9D



Se puede demostrar que la solución corresponde al autovector correspondiente al menor autovalor de $A^T \cdot A$

corresponde a minimizar por mínimos cuadrados

- Detección de corners en las 2 imágenes
- Emparejamiento $\rightarrow$ match by correlation (correlando ventana alrededor de los corners detectados) (normalizar correlación para inmunizarnos a cambios de brillo)

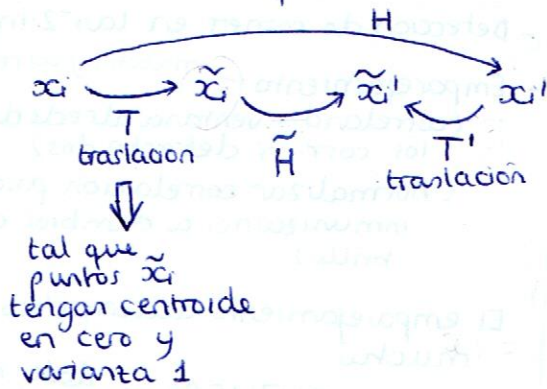
El emparejamiento automático falla mucho

OUTLIERS $\rightarrow$ datos muy malos

La estimación debe evitar esas parejas, y minimizar el error cuadrático sólo con las parejas buenas

$\rightarrow$ ver algoritmo RANSAC

Al estimar H interesa que el centroide de x_i y x_i' sea el origen



$$H = T \cdot \tilde{H} \cdot T'^{-1}$$

mas adecuada de estimar que directamente H
(el sistema de coordenadas de las imágenes sí afecta a la H estimada; conviene sistema de coordenadas centradas)

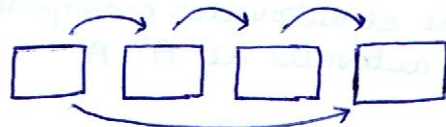
Estima de los parámetros (sabiendo que pueden haber OUTLIERS)

- M-estimator: - pondera por valores bajos los puntos con más error
- funciona bien sólo si hay pocos outliers y con error pequeño
 - RANSAC:
 - Tomar una muestra (mínimo n° de puntos para estimar el modelo)
↳ Estimar el modelo a partir de esa muestra (tomada aleat.)
 - Mirar cual es el SUPPORT / APOYO del modelo
calculado como el n° de puntos por debajo de una distancia t del modelo (consensus set)
 - Repetir para k muestras y seleccionar el modelo de mayor SUPPORT
 - Podemos descartar los outliers de ese modelo y ahora aplicar mínimos cuadrados a los inliers
- ¿cuánto? según el número de outliers frente a inliers
con probabilidad:

 - Probabilidad de todas las puntos de una muestra sean inliers es w^n
 - La probabilidad de que tras coger k muestras al menos una tenga todos los puntos inliers es $p = 1 - (1 - w^n)^k$

Composición

- Estimar la homografía entre las fotos que se solapan
- Para las que no se solapan, la homografía puede obtenerse como composición de las homografías de las fotografías intermedias



Blending

- Promediar píxeles para que no se noten los 'costurones'
- Nota: los costurones se notan más en las zonas de baja frecuencia (ejemplo: cielo), mientras que los promediados se notan mucho en las zonas de alta frecuencia
- Solución: promediado adaptativo.