Recuperación de Información Multimedia

Procesamiento de Videos

(Codecs, Shots, Keyframes)

CC5213 – Recuperación de Información Multimedia
Departamento de Ciencias de la Computación
Universidad de Chile
Juan Manuel Barrios – https://juan.cl/mir/ – 2020

Videos

- CODEC “compressor-decompressor”
  - Método para comprimir frames o audio
  - Compresión de frames
    - h.261, h.264, mpeg-2, Xvid, DV, etc.
  - Compresión de audio
    - MP3, speex, vorbis, aac, etc.
- Container
  - Formato para almacenar frames comprimidos, audio comprimido y metadatos
    - avi, mov, mpg, mkv, ogv, etc.
  - Algunos container soportan solo algunos codecs

Estándares

- Los estándares para codecs se enfocan en:
  - Normar el decoder
  - Favorecer un decoder muy simple
  - Dejar la complejidad para el encoder
- MPEG (Moving Pictures Experts Group)
  - Grupo de trabajo ISO/IEC, creado en 1998
  - MPEG-1, MPEG-2, MPEG-4, MPEG-7, MPEG-21
- VCEG (Video Coding Experts Group or Visual Coding Experts Group)
  - Grupo de trabajo de ITU-T, creado 1994
  - h.261, h.262, h.263, h.264

Estándares MPEG

- MPEG-1: 1993, estándar inicial con 5 partes
  - Parte 1: Definición de archivo container = mpg
  - Parte 2: Codificación de video. Basado en h.261
  - Parte 3: Codificación de audio, 3 formatos posibles:
    - Layer I (mp1), Layer II (mp2), Layer III (mp3)
- MPEG-2: 1995, 11 partes
  - Parte 1: Definición de archivo container incluyendo streams
  - Parte 2: Extensión MPEG-1 para calidad DVD y HDTV = h.262
  - Parte 3: Extensión MPEG-1 para multi-channel = 5.1
  - Parte 7: Advanced Audio Coding (AAC)
- MPEG-4: 1998, 28 partes (aún en desarrollo)
  - Parte 1, 12, 14, 15: Transmisión y formato del container = mp4
  - Parte 2: Compresión de video = h.263
  - Parte 10: Advanced Video Coding (AVC) = h.264
  - Parte 3: Inclusión de audio lossless y otros encoders

Otros estándares MPEG

- MPEG-7:
  - Estándar para incluir metadatos en contenido multimedia
  - Metadata de alto nivel y bajo nivel (descriptores visuales)
- MPEG-21:
  - Marco para intercambio de contenido multimedia
  - Mercado digital, restringiendo derechos de autor

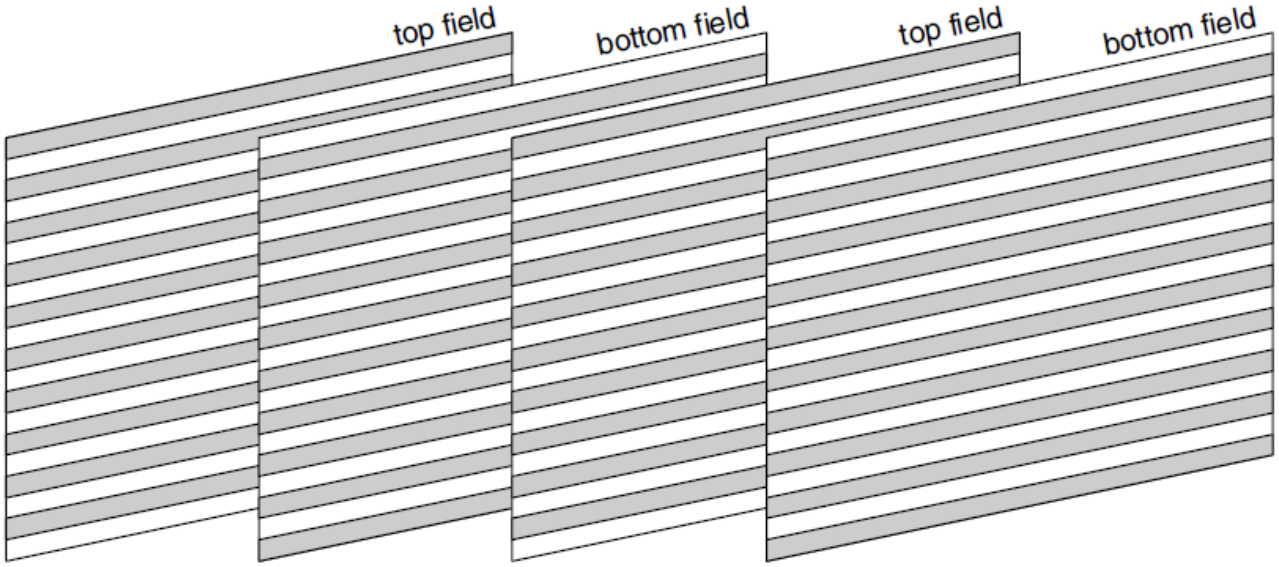
Ver: https://en.wikipedia.org/wiki/Moving_Picture_Experts_Group

Resoluciones de Video

- SD TV (análogo)
  - NTSC: 480i, 29.97fps (333x480, 640x480)
  - PAL: 576i, 25fps (335x576, 768x576)
- HD TV
  - HD-1: 720p, 1280x720
  - HD-2: 1080i, 1080p, 1920x1080
- Ultra HD TV
  - UHD-1: 2160p, 3840x2160, 4K
  - UHD-2: 4320p, 7680x4320, 8K

Why 29.97?: https://www.youtube.com/watch?v=3GJUM6pCpew

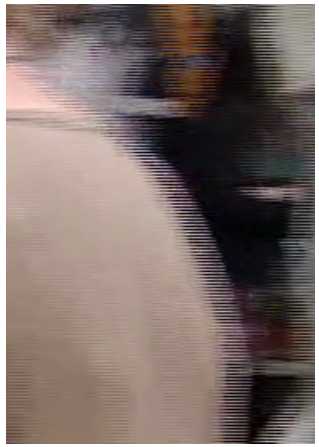
Interlaced Videos



Ver Richardson, cap 2

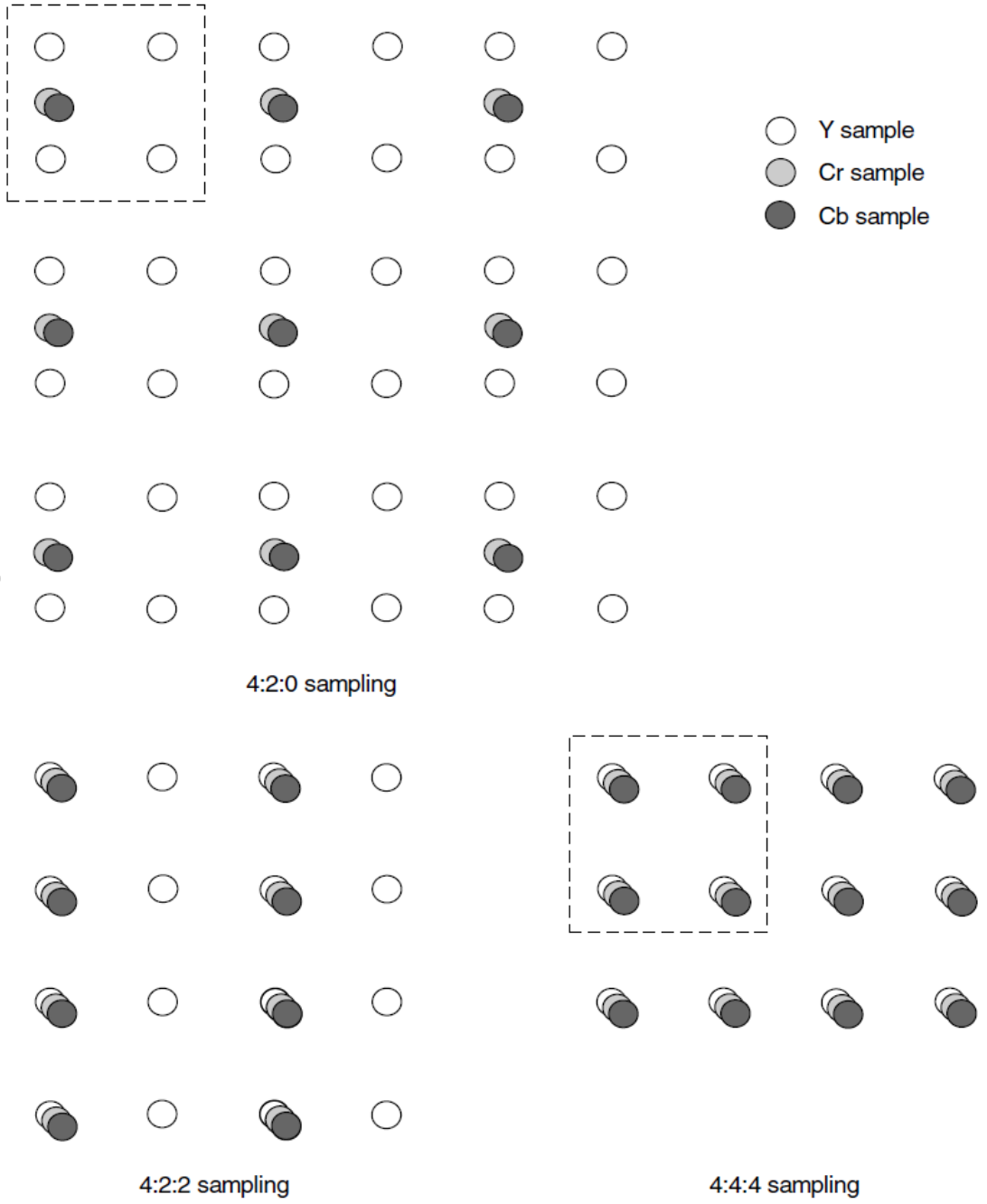
Interlaced Videos

- Aparecen tramas horizontales y bordes dobles cuando hay movimiento



Colores

- Convertir RGB a un canal gris (Y) y dos canales cromáticos (U,V)
- Bajar resolución de los canales de color
  - 4:4:4
  - 4:2:0



Ver Richardson, cap 2

Espacios “Y__”

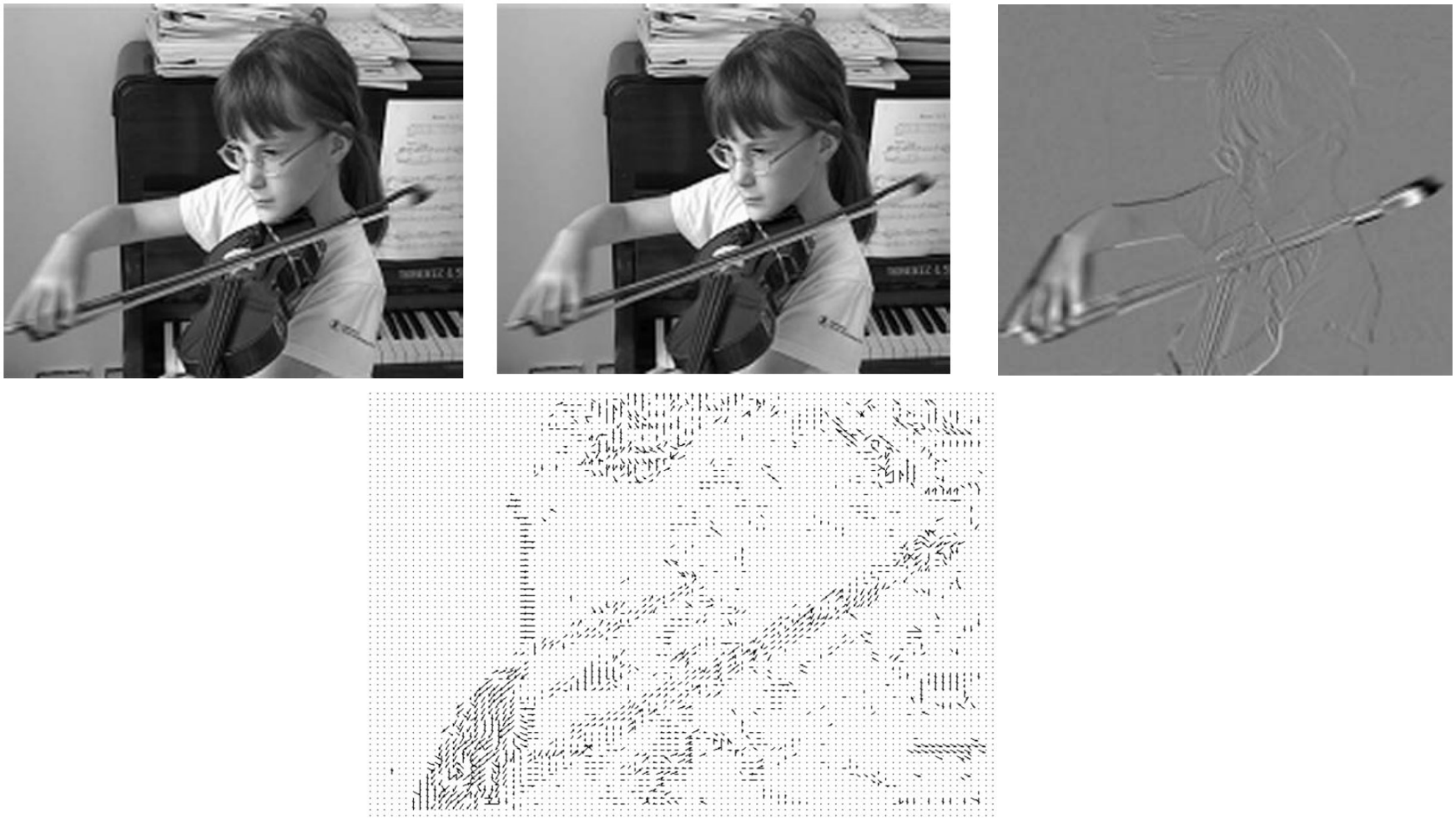
- YUV ≈ Y’UV ≈ YCbCr ≈ YPbPr
  - Más que espacios de color son codificaciones de RGB
- Conversión RGB ↔ YCbCr

$$
\begin{aligned}
Y &= 0.299R + 0.587G + 0.114B + 0 \\
C_B &= -0.169R - 0.331G + 0.499B + 128 \\
C_R &= 0.499R - 0.418G - 0.0813B + 128
\end{aligned}
$$

$$
\begin{aligned}
R &= [(Y + 1.402 \times (C_R - 128)]_0^{255} \\
G &= [(Y - 0.344 \times (C_B - 128) - 0.714 \times (C_R - 128)]_0^{255} \\
B &= [(Y + 1.772 \times (C_B - 128)]_0^{255}
\end{aligned}
$$

https://en.wikipedia.org/wiki/YUV#Y%E2%80%B2UV444_to_RGB888_conversion

Optical Flow



https://docs.opencv.org/master/dc/d6b/group__video__track.html#ga5d10ebbd59fe09c5f650289ec0ece5af

Ver Richardson, cap 3

Codificación de frames

- Usualmente dos frames consecutivos son muy similares.
- En vez de guardar dos frames consecutivos en forma independiente basta con guardar sólo la diferencia.
  - Si tiene muchos ceros tendrá mejor compresión.
  - Imagen Residual = Frame 2 – Frame 1

Frame 1

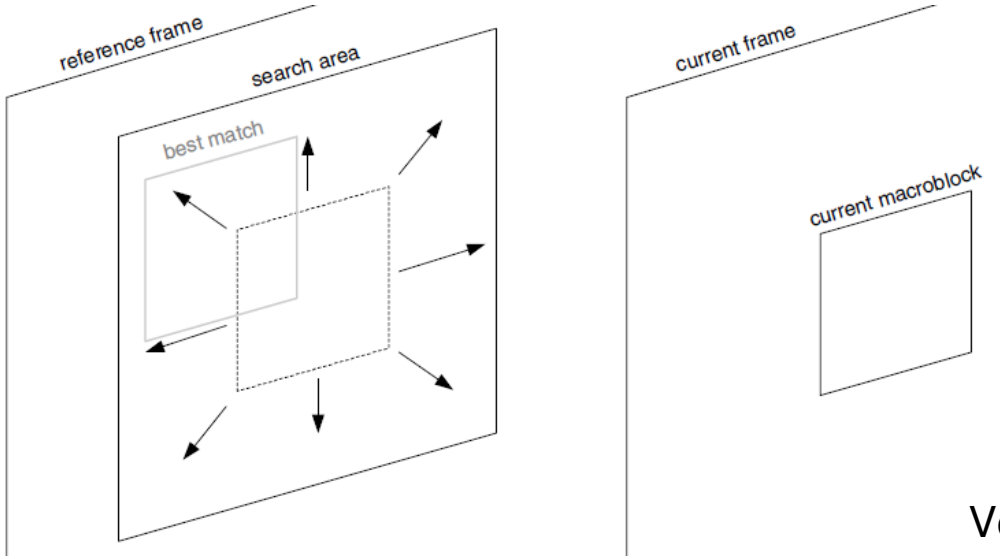
Frame 2

Imagen Residual

Ver Richardson, cap 3

Estimación de movimiento por bloques

- Se hace una estimación del movimiento usando bloques:
  - Se divide un frame en “macrobloques” de NxN.
  - Cada macrobloque del frame 2 se forma con algún bloque de NxN del frame 1 más la imagen residual.
  - El vector que apunta al lugar de donde obtener el bloque base desde el frame 1 se denomina “Motion Vector”.



Ver Richardson, cap 3

Estimación del movimiento

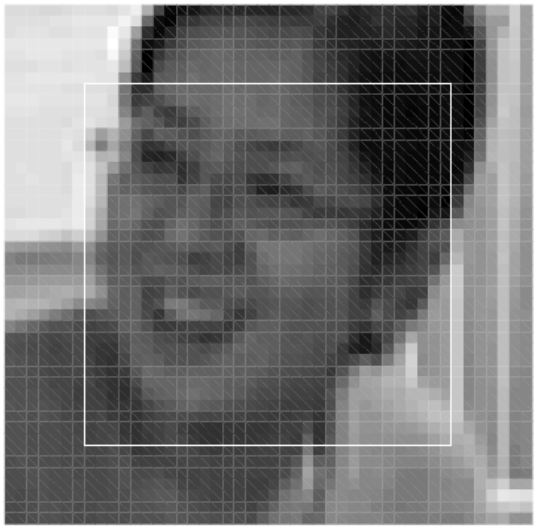
- Dado dos frames, buscar la posición del macrobloque actual dentro de la imagen previa que minimiza el error:

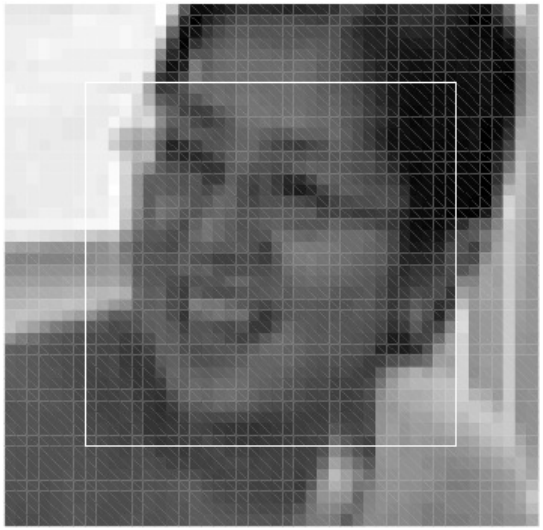
Mean Squared Error: $$MSE = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (C_{ij} - R_{ij})^2$$

Mean Absolute Error: $$MAE = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}|$$

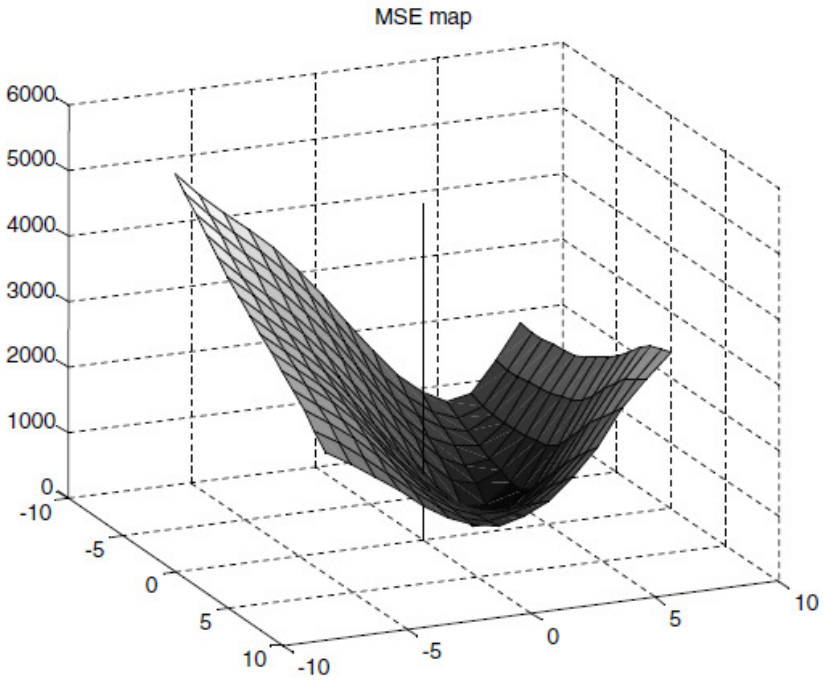
Sum of Absolute Errors: $$SAE = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}|$$

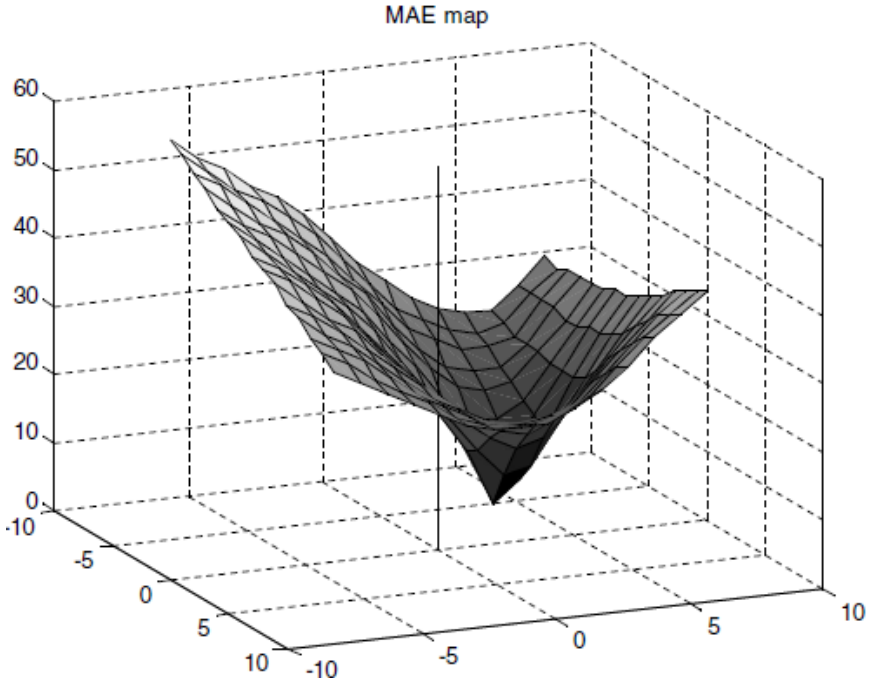
Ver Richardson, cap 7





Mínimo error para el desplazamiento (2,0)





Ver Richardson, cap 7

Motion Vectors para macrobloques 4x4

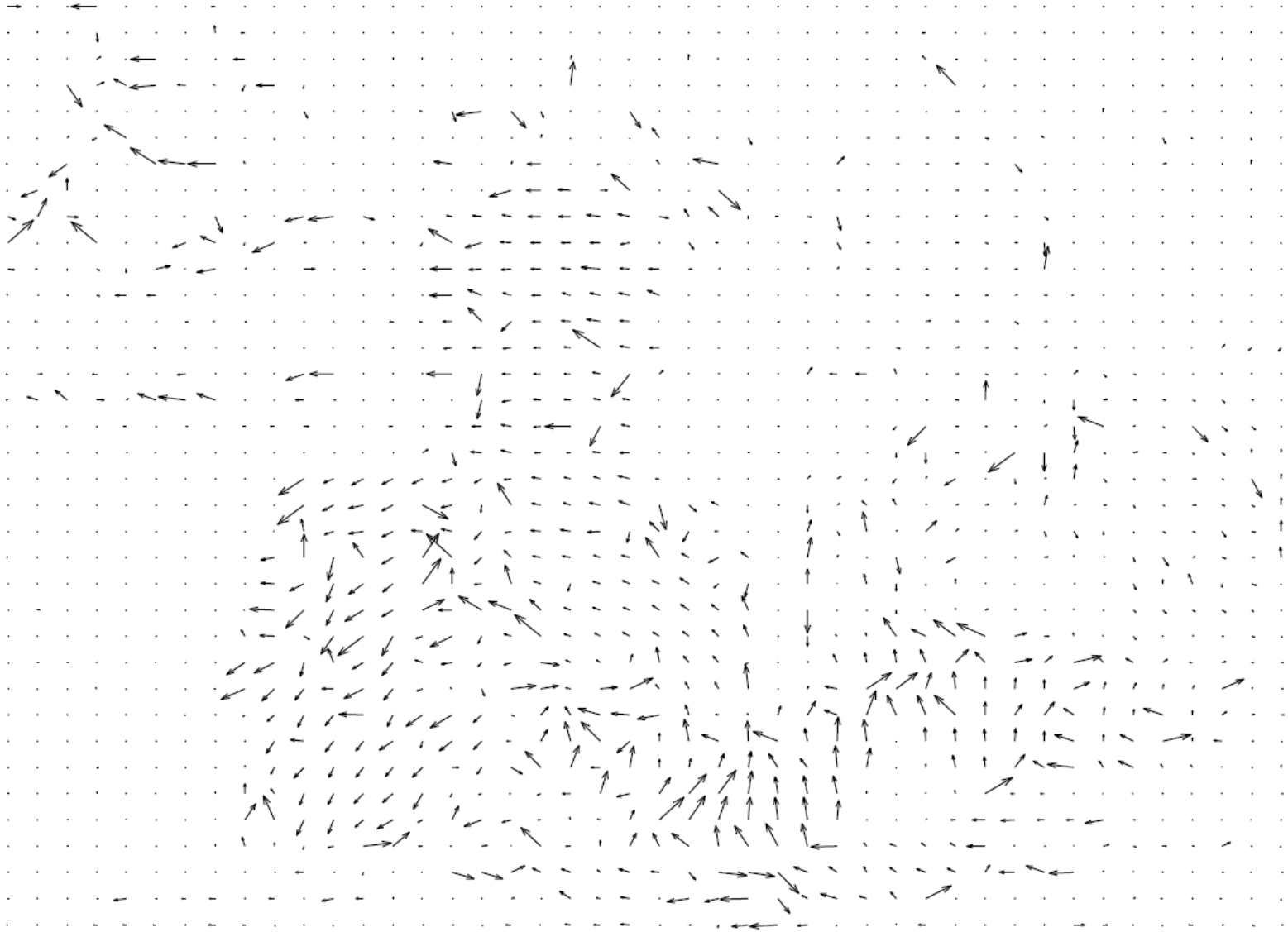
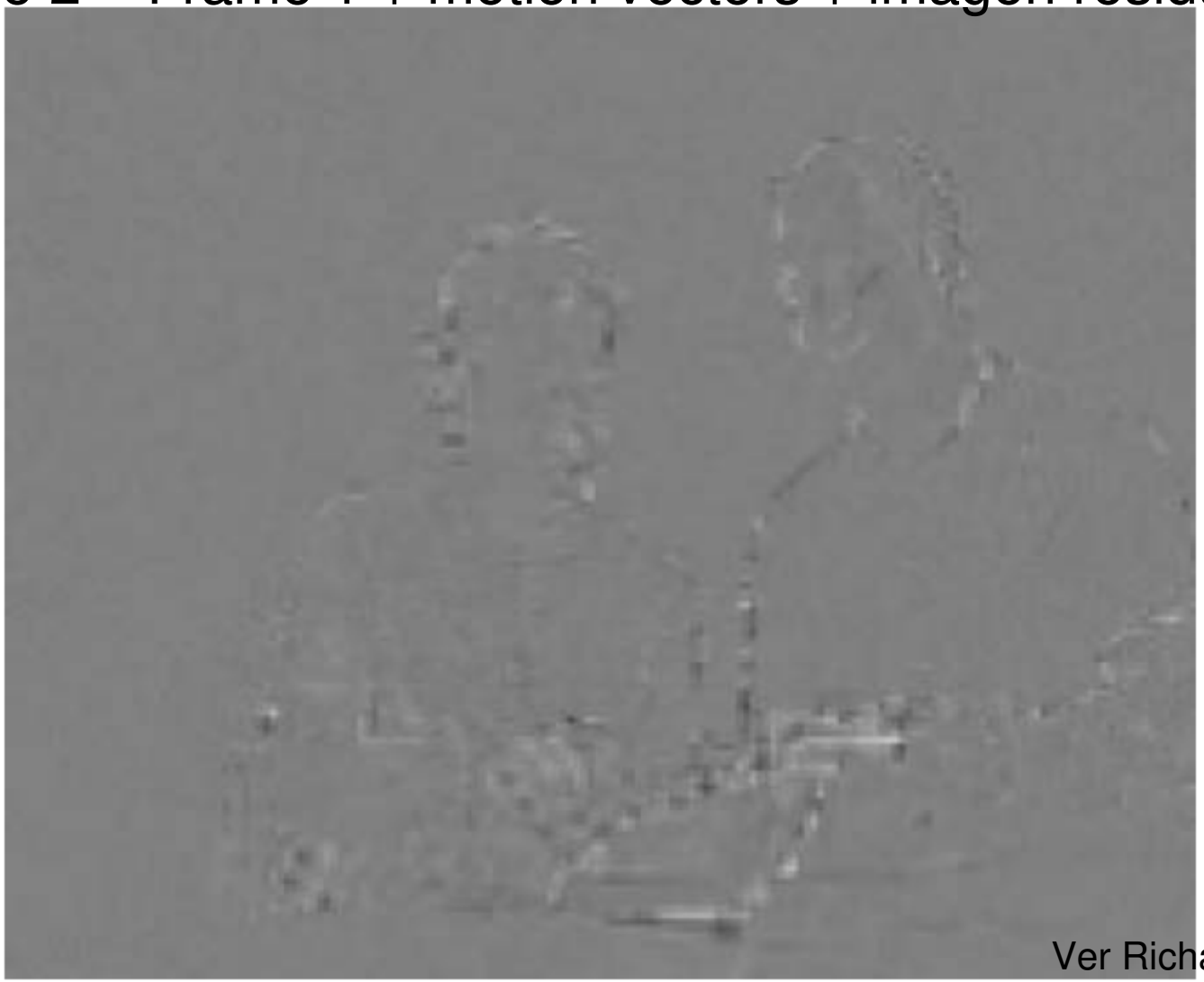


Imagen residual luego de ajustar frame 2 con motion vectors para macrobloques de 4x4

Frame 2 = Frame 1 + motion vectors + imagen residual



Ver Richardson, cap 3

Imagen residual luego de ajustar frame 2 con motion vectors para macrobloques de 16x16

(aumenta la información en la imagen residual)

Ver Richardson, cap 3

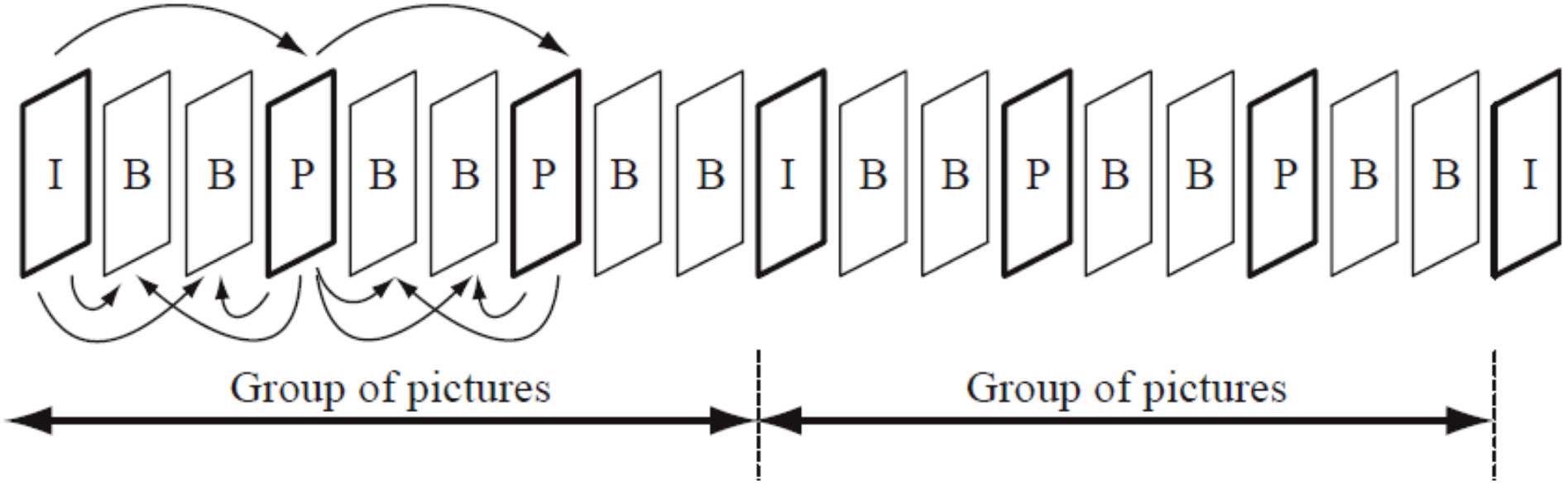
Codificación con MPEG-1

- Dado un video, se quieren comprimir los frames para reducir el tamaño del archivo
- La compresión se basa en:
  - Estimación del movimiento de un frame usando motion vectors
  - Comprimir la imagen residual como en jpg:
    - Transformación de la imagen residual con DCT
    - Compresión de la entropía (compresión sin pérdida como codificación huffman o codificación aritmética)

Tipos de frames

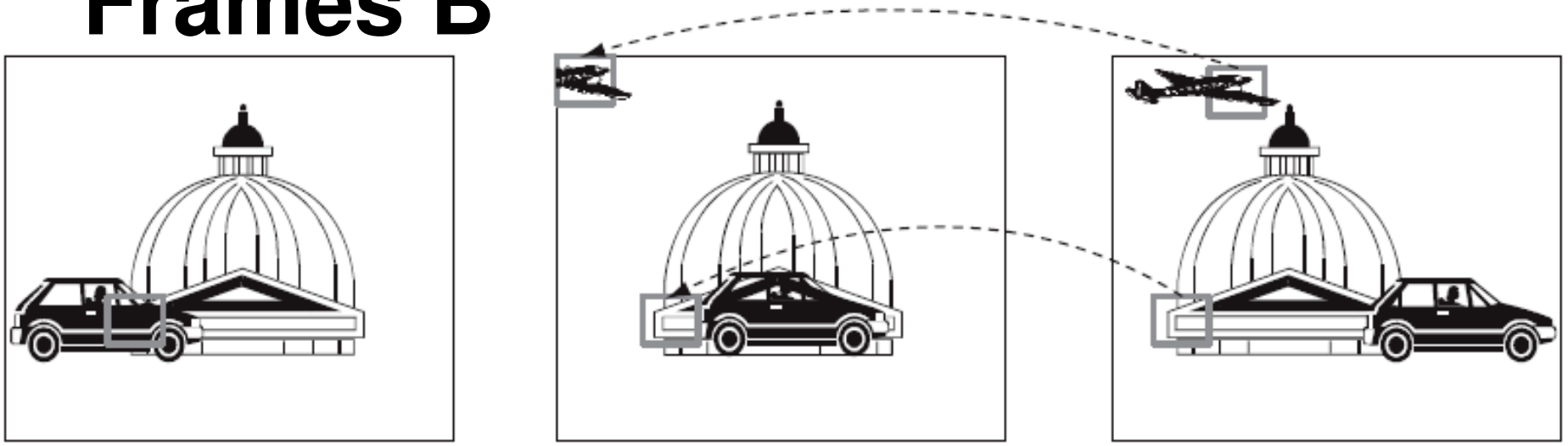
- Los frames del video se clasifican en 3 tipos:
  - **Intra-coded (frames I)**: se comprime como una imagen estática (jpg)
  - **Predictive coded (frames P)**: se comprime usando motion vectors con un frame I o P previo
  - **Bidirectional predicted (frames B)**: se comprime usando como referencia frames I o P previos y posteriores

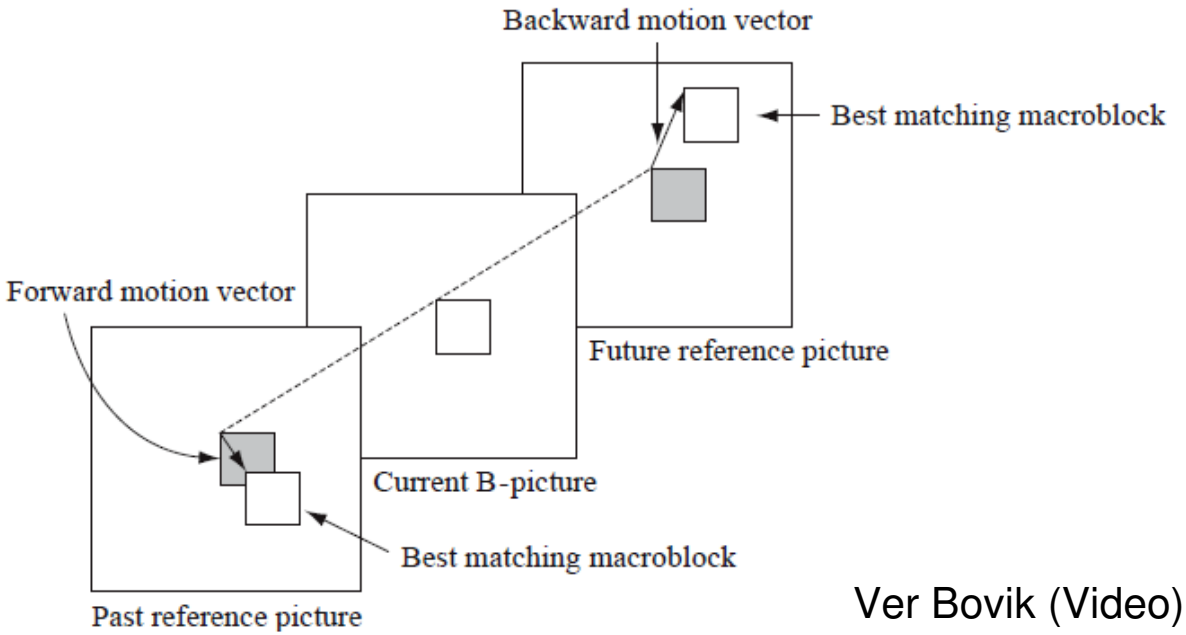
Tipos de frames



Ver Bovik (Video), cap 9.1

Frames B





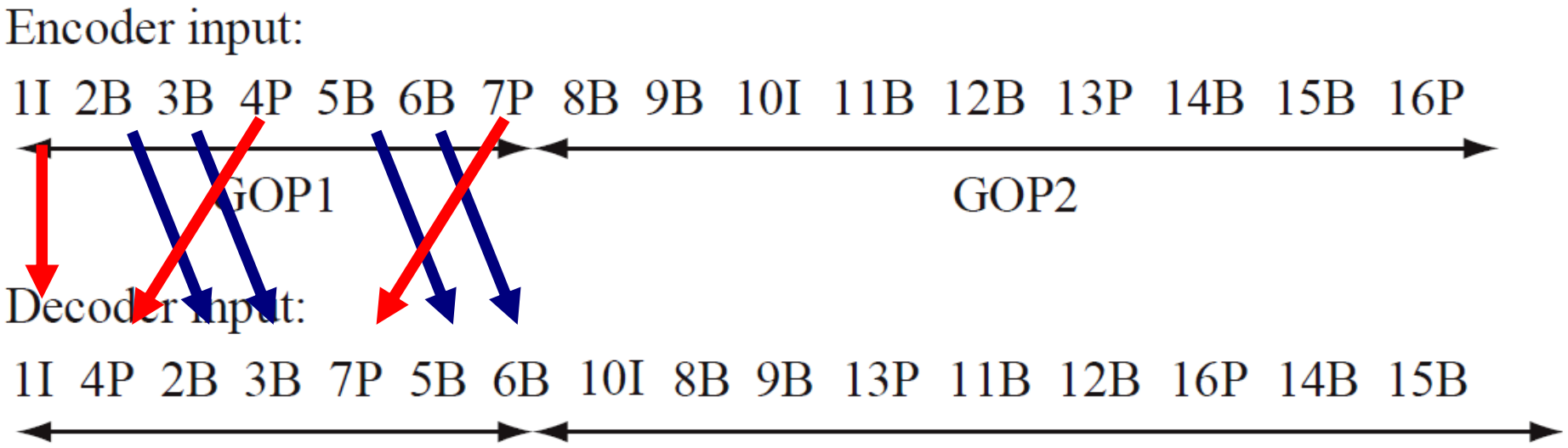
Ver Bovik (Video), cap 9.1

Tipos de frames

- Frames I
  - No dependen de otro frame.
  - Su compresión el relativamente baja.
  - Finaliza la propagación de errores previos.
- Frames P
  - Dependen del frame I o P previo.
  - Propagan los errores que pueden existir en frames I o P previos
- Frames B
  - Dependen de frames previos y posteriores
  - Mayor compresión.
  - Para poder decodificarlos se debe decodificar frame posterior.
  - No hay dependencias sobre frames B por lo que un error en un frame B no se propaga (pero propaga los errores de los frames en los que depende).

Reordenamiento de frames

- Los frames no se guardan en orden correlativo, si no que se deben guardar primero los frames I o P y luego los B
- Se requiere un buffer de decodificación



Ver Bovik (Video), cap 9.1

Ejercicio

- Se tiene un video compuesto de 19 frames, donde un algoritmo MPEG-1 decide que los frames tipo I, P y B serán de acuerdo a la siguiente tabla :

I	B	B	P	B	B	B	P	B	P	B	B	I	B	B	P	B	P	I
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

  - ¿En qué orden se guardarán los frames en el archivo mpg resultante de la codificación?

  - Si ocurre un error al decodificar un frame y se obtiene una imagen verde ¿En qué otros frames se verán manchas de color verde? Específicamente, qué sucede cuando el frame errónero es:
    - El frame 2 , el frame 8, el frame 13

Herramientas

- FFmpeg https://ffmpeg.org/
  - Librería open source (LGPL/GPL) para videos
  - Librerías: libavcodec, libavformat, libavfilter, …
  - Comandos: ffmpeg, ffplay
- x264
  - Librería codec h.264 con licencia GPL
  - Incluido en FFmpeg (GPL)
- VLC https://www.videolan.org/
  - Reproductor de video open source
- Mplayer, Mencoder http://www.mplayerhq.hu
  - Reproductor de video y encoder open source
- ImageMagick https://imagemagick.org/
  - Librería open source para editar imágenes
  - Comandos: convert, display, identify

Algunos Comandos Útiles

- https://trac.ffmpeg.org/wiki
  - https://trac.ffmpeg.org/wiki/Encode/H.264
  - https://trac.ffmpeg.org/wiki/Scaling
  - https://trac.ffmpeg.org/wiki/Concatenate
- https://imagemagick.org/Usage/
  - https://imagemagick.org/Usage/quantize/
  - https://imagemagick.org/Usage/resize/

```
ffmpeg -ss 06:42 -t 60 -i entrada.mp4 -vf scale=-2:600
       -crf 30 -preset veryslow salida.mp4

ffmpeg -i entrada.mp4 -r 1 -f image2 imagen-%03d.png
ffmpeg -framerate 30 -i imagen-%03d.png salida.mp4

convert foto.jpg +dither -colors 3 foto-3.png
convert foto.jpg -dither FloydSteinberg -colors 3 foto-3.png
```

Detección de Shots

Videos

- Frame: unidad mínima
- Shot: secuencia continua de frames procedente de una cámara que representa una acción continua en el tiempo y espacio
- Scene: conjunto de shots en una misma ubicación

Detección de límites de shots

- Los frames pertenecientes a shots distintos presentan un cambio en su contenido
  - Detectar discontinuidades en el flujo del contenido de los frames
- En general, extraer un descriptor global al frame *i* y al frame *i+1*, calcular la distancia *d(i, i+1)*
  - Si es mayor a un umbral entonces hay un cambio de shot

Detección de límites de shots

- Diferencia de frames:
  - Hay un cambio de shot si la distancia L1 entre frames consecutivos es mayor a un umbral
- Cantidad de pixeles cambiados:
  - Se define que un pixel cambia cuando la diferencia de intensidad entre dos frames supera un umbral2
  - Hay un cambio de shot cuando el número de pixeles que cambian sea mayor a un umbral1
- Reducir la imagen o usar filtro gaussiano para reducir ruido

Detección de límites de shots

- Diferencias estadísticas
  - Dividir cada frame en zonas y conocer la media y varianza del canal Y para cada zona en el video
  - Cuando las zonas se alejan de la media hay un límite de shot
- Histogramas
  - Cambio cuando la distancia entre histogramas consecutivos supera un umbral
  - 4x4 zonas, histograma por zona, eliminar las 8 zonas con más cambios, hay cambio cuando la suma de las 8 menores supera un umbral

Detección de límites de shots

- Falso positivo:
  - Fotografías con flash.
  - Comparar 2 pares:
    - Diferencia es min{ $d(i,i+1), d(i-1,i+2)$ }
- Falso negativo:
  - Transiciones suaves entre shots.
  - Detectores específicos para transiciones (fade-in, fade-out)

Detección de límites de shots

- TRECVID durante 2001-2007 evaluó la detección de shots:
  - Resultados en:

    http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html#2007
- Algunos Papers:
  - J.S. Boreczky and L.A. Rowe. “Comparison of video shot boundary detection techniques”. 1996.
  - S.Eickeler and S.Müller. “Content-Based Video Indexing Of Tv Broadcast News Using Hidden Markov Models”. 1999.

Selección de Keyframes

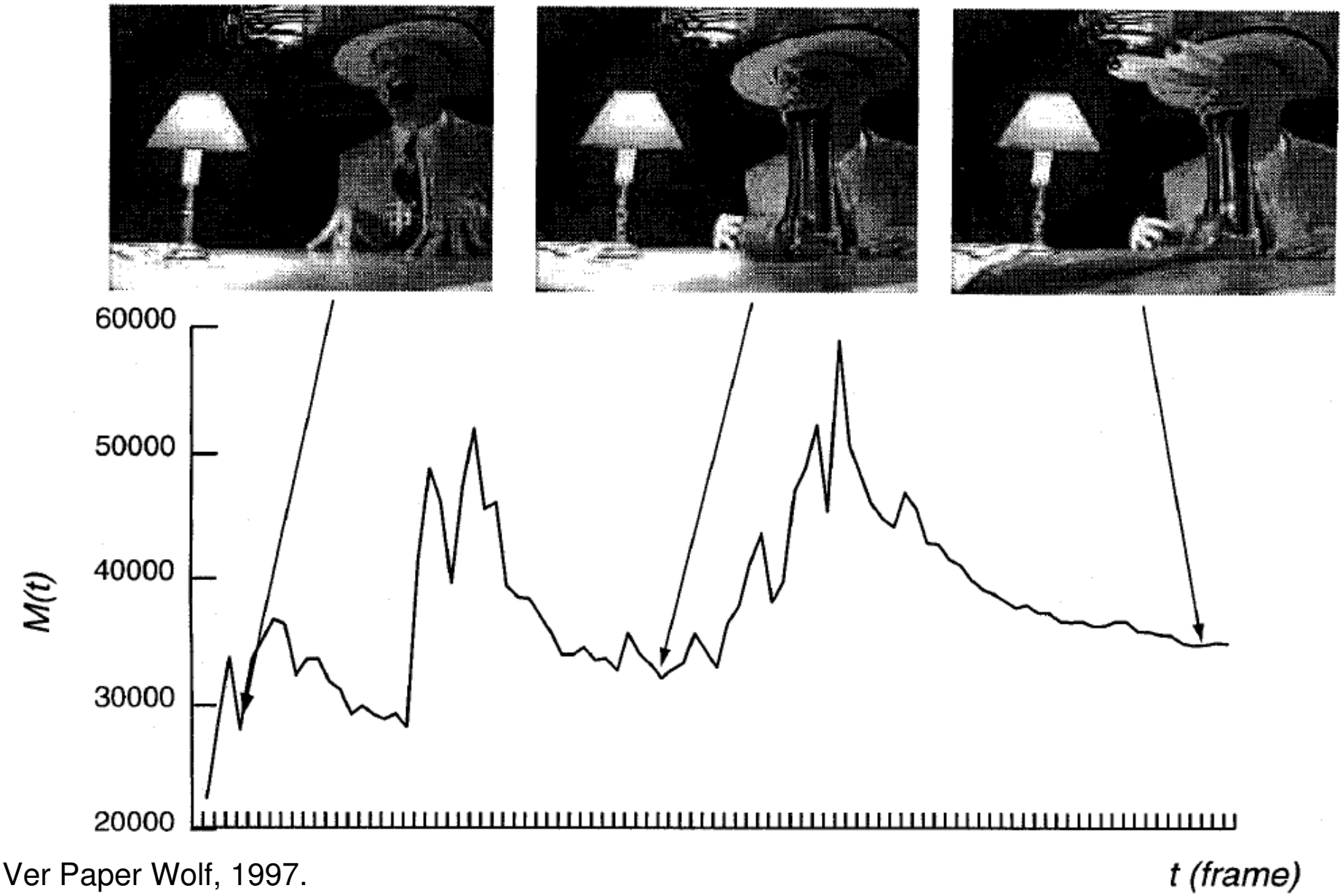
Selección de Keyframes

- Selección constante, por ej.:
  - 1 frame por segundo
  - 5 frames por segundo
  - 1 frame cada 3 segundos
- Calcular un descriptor global para todos los frames, clusterizar, y seleccionar los frames más cercano a los centroides

Selección de Keyframes

- Dividir en shots, para cada shot tomar los frames estables
  - Menor diferencia con el anterior según un descriptor global
  - Menor movimiento según el optical flow

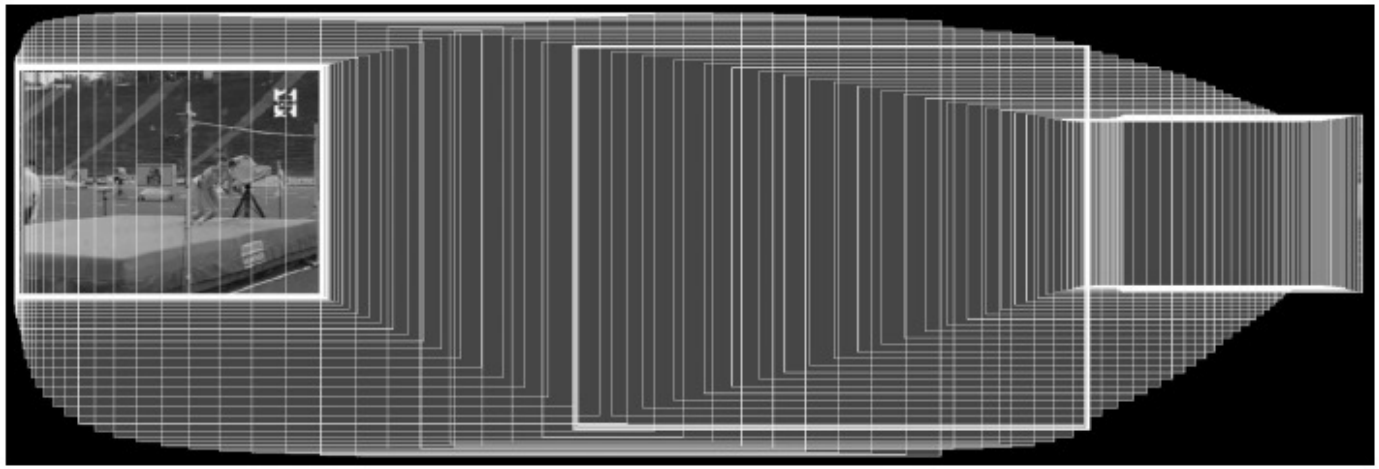
$$M(t) = \sum_i \sum_j |o_x(i,j,t)| + |o_y(i,j,t)|$$



Ver Paper Wolf, 1997.

Selección de Keyframes

- Por estimación del movimiento:



Ver Paper Fauvet et al., 2004.

Imagen + Audio

Combinación de descriptores

- En un video, para un segmento se puede obtener un descriptor visual y un descriptor de audio. Por ej:
  - Visual: Promedio de los valores de un descriptor de bordes (como Edge Histogram)
  - Visual: Promedio de los histogramas de color de todos los frames dentro del segmento
  - Audio: Promedio del vector de coeficientes de la escala Mel o del cepstrum
- ¿Como combinar toda esta información?
  - “Early Fusion” versus “Late Fusion”

Late Fusion

- Se realiza la búsqueda cada modalidad por separado y se obtienen resultados finales (o casi finales)
- Realizar combinación de resultados
  - Unión, intersección, suma de scores
- Requiere que cada modalidad por separado pueda obtener resultados razonables

Early Fusion

- Opción 1: Descriptor combinado
  - Escalar las dimensiones y crear un único descriptor
  - Usar distancia Euclidiana (u otra)

- Opción 2: Distancia combinada
  - No modificar los descriptores
  - Escalar valores de distancia de cada descriptor para que sean comparables

$$\delta_{av}(q,r) = \frac{w_1}{\tau_1} * L_1\text{-Eh}(q,r) + \frac{w_2}{\tau_2} * L_1\text{-Rgb}(q,r) + \frac{w_3}{\tau_3} * L_1\text{-Aud}(q,r)$$

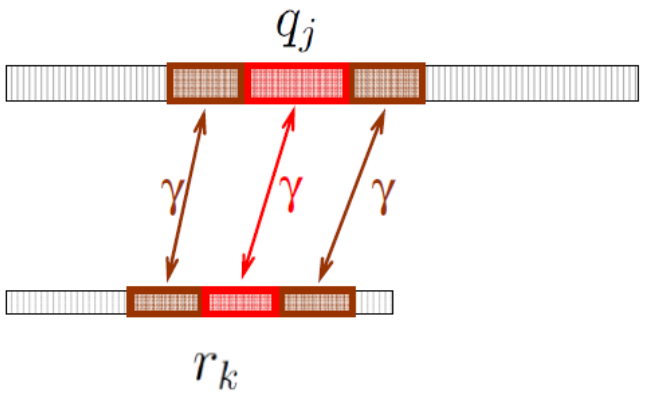
Early Fusion

- Se realiza una única búsqueda con la distancia combinada
- Puede localizar elementos que requieran usar varias modalidades a la vez
- Los resultados son afectados por descriptores ruidosos
  - Se pueden descartar descriptores ruidosos en forma dinámica
    - Ver capítulo de multi-métricas

Distancia Temporal

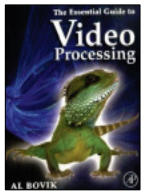
- Se puede aumentar la robustez de la distancia si se incluyen los segmentos anteriores y posteriores
  - Para ventana W:

$$\delta(q_j, r_k) = \frac{1}{W} \sum_{w=-\lfloor W/2 \rfloor}^{\lfloor W/2 \rfloor} \gamma(q_{j+w}, r_{k+w})$$

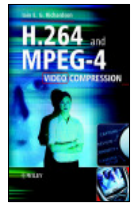


Bibliografía

- **The Essential Guide To Video Processing**. Bovik. 2009.
  - Cap 9

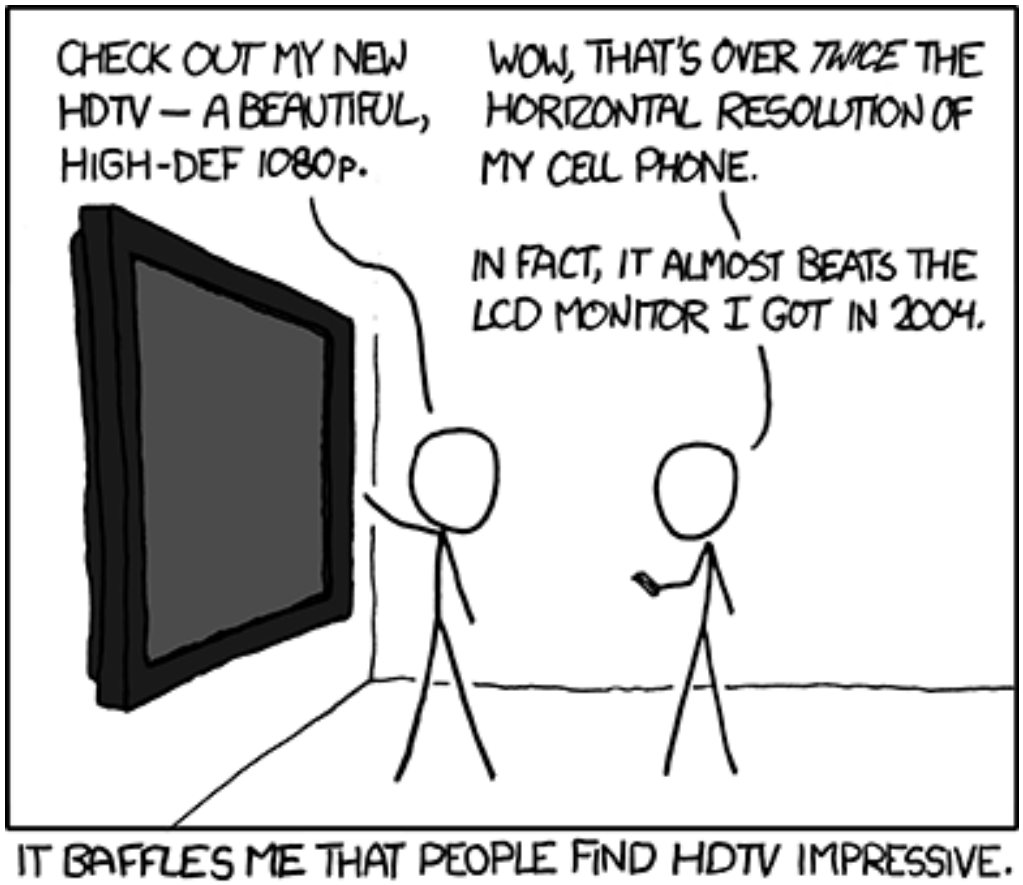


- **H.264 and MPEG-4 Video Compression**. Richardson. 2003.
  - Cap 3 y 7



Papers

- Wolf. “Keyframe selection by motion analysis”. 1996.
- Fauvet et al. “A Geometrical Key-Frame Selection Method Exploiting Dominant Motion Estimation in Video”. 2004.
- Sun et al. “Content-based representative frame extraction for digital video”. 1998.
- Zhuang et al. “Adaptive Keyframe Extraction Using Unsupervised Clustering”. 1998.



"*We're also stuck with blurry, juddery, slow-panning 24fps movies forever because (thanks to 60fps home video) people associate high framerates with camcorders and cheap sitcoms, and thus think good framerates look 'fake'.*"

http://xkcd.com/732/