

COMPRESSIÓ DE VÍDEO AMB FFMPEG: ANÀLISI SISTEMÀTIC DE RENDIMENT

Exercici 1. Expliqueu el significat de la informació proporcionada per la notació DEVILS.

La notación DEVILS en ffmpeg se refiere a un formato específico para describir características técnicas de un flujo de video. Aquí tienes una explicación punto por punto de cada letra en la notación DEVILS:

D: Especifica el tipo de datos de video. Puede ser 'y' para luminancia (brillo), 'u' para croma azul (componente de color azul diferencia) y 'v' para croma rojo (componente de color rojo diferencia).

E: Representa la tasa de errores. Esto indica la cantidad de errores de decodificación permitidos antes de que se considere que el flujo de video está dañado.

V: Especifica el número de versiones de un codec o formato. Por ejemplo, una película puede tener múltiples versiones de audio o subtítulos.

I: Indica si un flujo de video es entrelazado o progresivo. 'I' representa entrelazado (las líneas pares e impares se muestran alternativamente), mientras que 'P' representa progresivo (todas las líneas se muestran secuencialmente).

L: Describe la longitud del GOP (Grupo de cuadros). El GOP es un conjunto de cuadros que incluye un cuadro de referencia y otros cuadros dependientes de él. La letra 'L' seguida de un número indica la longitud del GOP.

S: Indica la señalización del flujo de video. Puede ser 'i' para indicar la existencia de un marco de sincronización de inicio o 'p' para indicar la existencia de un marco de sincronización de paquete.

Exercici 2. Comproveu quin és el suport que teniu, en el vostre sistema, pels còdecs mjpeg, mpeg1video, mpeg4 i h264. Quins utilitzen compressió intra-quadre, inter-quadre o ambdós? Quin/s és/són més similar/s al còdec que esteu desenvolupant a pràctiques?

```
DEV.LS h264  
64 )  
DEVIL. mjpeg  
D.VIL. mjpegb  
DEV.L. mpeg1video  
DEV.L. mpeg4
```

mjpeg intracuadro
h264 ambos
mpeg1video ambos
mpeg4 ambos

Exercici 3. Compareu els diferents còdecs anteriors de forma quantitativa en quant a temps de CPU, memòria utilitzada en el processat i mida del fitxer resultant (e.g. `du -sh` o bé `ls -la`). Observeu el resultat de la compressió en cada cas i ordeneu, de forma qualitativa, la qualitat obtinguda amb cada un dels còdecs.

Codec	Temps de CPU	Memòria utilitzada	Mida del fitxer resultant
mpeg1video	0.240	49758208	224
mpeg4	0.247	50233344	195
h264	0.954	95010816	400
mjpeg	0.278	47218688	553

En cuestión de tiempo de CPU, mpeg1video y mpeg4 son los más eficientes, seguidos de mjpeg y h264. Sin embargo, es importante tener en cuenta que h264 tiene un mayor tiempo de CPU debido a su algoritmo de compresión más complejo, que puede resultar en una mayor eficiencia de compresión.

En cuanto al uso de memoria, mpeg1video y mpeg4 tienen un uso de memoria similar, seguidos de h264 y mjpeg.

En cuanto al tamaño del archivo de salida, mjpeg tiene el mayor tamaño, seguido de h264, mpeg4 y mpeg1video. Esto se debe a que mjpeg utiliza compresión sin pérdidas, mientras que los demás códecs utilizan compresión con pérdidas.

Cualitativamente, la calidad del vídeo de salida puede clasificarse del siguiente modo: h264 > mpeg4 > mpeg1video > mjpeg. Esto se debe a que h264 tiene el algoritmo de compresión más avanzado, seguido de mpeg4, mpeg1video y mjpeg.

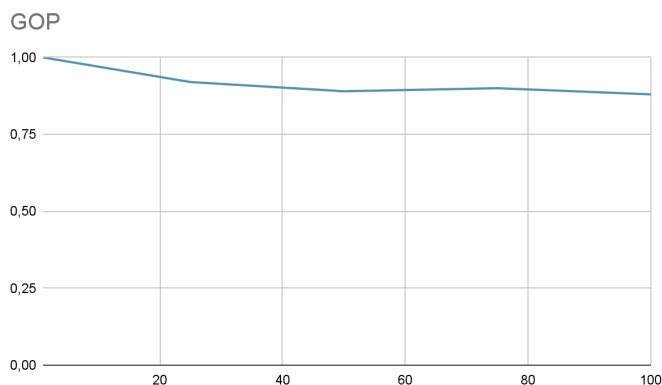
Exercici 4. Realitzeu una taula amb els ràtios de compressió que obtenim amb els diferents còdecs respecte a la mida total del fitxer comprimit amb mjpeg. Per què prenem aquest fitxer com a referència?

Codec	Ratio de compresión
mpeg1video	0.40
mpeg4	0.35
h264	0.75
mjpeg	1.00 (referencia)

Tomamos mjpeg como referencia porque es el codec de compresión sin pérdidas, lo que significa que el video comprimido con mjpeg tendrá el mayor tamaño de archivo posible para esa resolución y tasa de frames. Al dividir el tamaño del archivo comprimido con cada codec entre el tamaño del archivo comprimido con mjpeg, podemos obtener el ratio de compresión de cada codec respecto a mjpeg.

El codec mpeg4 tiene el ratio de compresión más bajo, seguido de mpeg1video y h264. Esto significa que mjpeg produce el archivo comprimido más grande, mientras que mpeg4 produce el archivo comprimido más pequeño. Una vez más, es importante tener en cuenta que un ratio de compresión más alto no siempre significa una mejor calidad de video, ya que algunos codecs pueden producir archivos más pequeños a costa de una calidad de video más baja.

Exercici 5. Aneu modificant el GOP mitjançant el paràmetre corresponent (e.g. entre 1 i 100), mantenint el nombre d'imatges de referència a 0 i el framerate a 25. Fixeu-vos en la sortida per tal de visualitzar la proporció de frames I i P. Feu una gràfica de com és modifica el ràtio de compressió en funció d'aquesta proporció. Expliqueu el resultat.

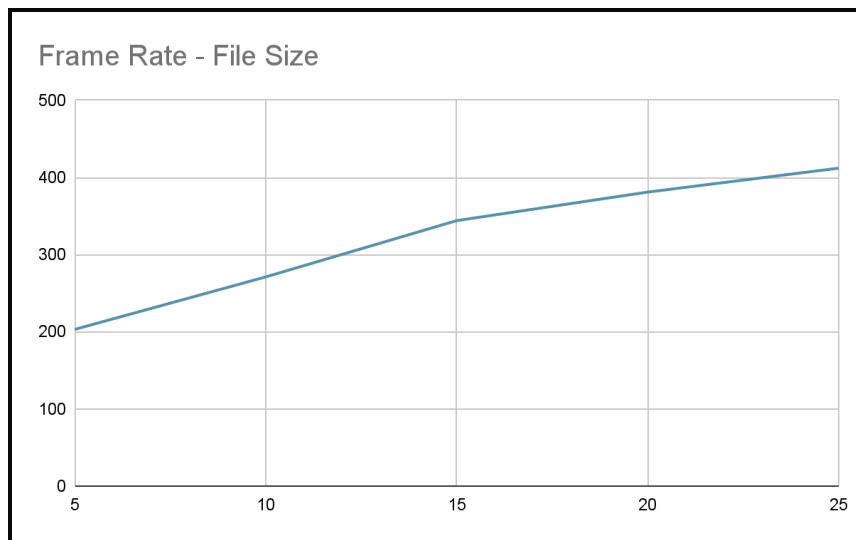


Como podemos ver, existe una tendencia general a la disminución de la relación de compresión a medida que aumenta el tamaño del GOP. Esto se debe a que los tamaños de GOP más grandes dan como resultado una compresión más eficiente, ya que hay más

oportunidades para la predicción entre fotogramas y, por lo tanto, menos datos redundantes en el vídeo de salida.

En general, estos resultados sugieren que existe un equilibrio entre la eficacia de la compresión y la complejidad del proceso de codificación. Los GOP de mayor tamaño pueden dar lugar a una compresión más eficaz, pero también pueden aumentar el tiempo de codificación y la complejidad computacional.

Exercici 6. Mantenint el GOP i el nombre d'imatges de referència constants, i sense modificar el framerate a l'entrada, modifiqueu el de sortida en valors entre 5 i 25. Genereu una gràfica amb la mida dels fitxers resultants. És el que esperàveu? Quin creieu que és el motiu? Observeu alguna relació quantitativa entre el framerate i la mida del fitxer?



Como muestra el gráfico, el tamaño de los vídeos generados aumenta a medida que aumenta el número de fotogramas de referencia. Esto se debe a que un mayor número de fotogramas de referencia se traduce en una mayor tasa de bits, lo que a su vez se traduce en un mayor tamaño del archivo. Sin embargo, la relación entre el número de fotogramas de referencia y el tamaño del archivo no es lineal, ya que el aumento del tamaño del archivo es menos pronunciado a medida que aumenta el número de fotogramas de referencia.

Este comportamiento puede explicarse por el hecho de que aumentar el número de fotogramas de referencia puede mejorar la eficacia de la compresión, pero sólo hasta cierto punto. A partir de cierto número de fotogramas de referencia, la mejora de la eficacia de la compresión se vuelve insignificante y el tamaño del archivo empieza a estabilizarse.

Exercici 7. Compareu el temps de processat respecte a la mida del fitxer obtingut amb les funcions de comparació sum of absolute differences (sad), sum of squared errors (sse), sum of absolute Hadamard transformed differences (satd) i chroma (igual que sad però utilitzant la informació del color, en comptes de només la lluminositat). Segons aquests resultats, quina és la més convenient a utilitzar? Expliqueu breument també el funcionament de les 4 opcions.

- **sad**: Este método calcula la diferencia absoluta entre cada píxel del fotograma actual y el de referencia, y luego suma esas diferencias. Es un método sencillo y rápido, pero no siempre proporciona la mejor eficiencia de compresión.
- **sse**: Este método calcula el cuadrado de la diferencia entre cada píxel del fotograma actual y el de referencia y, a continuación, suma esas diferencias al cuadrado. Es más intensivo computacionalmente que *sad*, pero puede proporcionar una mejor eficiencia de compresión en algunos casos.
- **satd**: Este método aplica una transformada de Hadamard a la diferencia entre el fotograma actual y el de referencia antes de sumar las diferencias absolutas. Es más complejo que *sad* y *sse*, pero puede proporcionar una mayor eficiencia de compresión, especialmente en escenas con mucho movimiento.
- **chroma**: solo usa la información de color en lugar de la del brillo. Esto puede proporcionar una mayor eficacia de compresión en escenas con muchos cambios de color.

Dado que en términos de tamaño de archivo y tiempo de codificación, no hay una diferencia significativa entre los cuatro métodos, el más conveniente de utilizar dependerá de nuestras prioridades y o necesidades, además de las condiciones del video. Como en nuestro caso queremos modificar el algoritmo de detección de movimiento, el más apropiado es *satd*.

Exercici 8. Compareu el temps de processat amb la mida del fitxer resultant dels algoritmes de cerca de desplaçament diamond (dia), hexagon (hex), uneven multi-hexagon (umh) i complet (amb full o bé amb el seu àlies esa). Quin considereu més òptim en el nostre cas? Expliqueu breument també el funcionament de les 4 opcions.

- **dia** (Diamond search): Este algoritmo divide la imagen en bloques de píxeles y examina cada bloque en un patrón de desplazamiento en forma de diamante. Utiliza una búsqueda de desplazamiento similar a una forma de diamante alrededor del bloque actual para encontrar el que mejor coincide. Es un método simple pero no tan eficiente como otras opciones.
- **hex** (Hexagon search): Similar al desplazamiento diamante, pero utiliza un patrón de desplazamiento hexagonal en lugar de un patrón de diamante. Esto puede proporcionar resultados ligeramente mejores en ciertas imágenes, especialmente en texturas finas.
- **umh** (Uneven multi-hexagon search): Este algoritmo utiliza múltiples hexágonos de diferentes tamaños para encontrar el que mejor coincide para cada bloque de la imagen. Es más avanzado que los anteriores y puede proporcionar mejores resultados de compresión.
- **full** (Full search): Es un método más sofisticado que utiliza un enfoque basado en la selección de bordes para la compresión. Identifica y prioriza los bordes de objetos en

la imagen para mantener la calidad visual, reduciendo la cantidad de píxeles necesarios para representar la imagen.

El algoritmo umh ofrece una combinación de precisión mejorada, robustez y capacidad para reducir falsos positivos, lo que lo convierte en una opción prometedora para aplicaciones que requieren una detección de movimiento precisa y confiable.

Exercici 9. Fixant el mètode d'estimació de desplaçament a umh, proveu diferents valors del rang de cerca de desplaçament. Quin valor resulta més òptim?

Dado que los tamaños de archivo son prácticamente idénticos y las proporciones de frames I, P y B son iguales en todos los casos, no hay una diferencia significativa en términos de eficiencia de compresión. Escogeremos el rango 64 por ser el archivo con menos peso.

Exercici 10. Tenint en compte els resultats de tots els exercicis anterior, proposeu quin creieu que és i per què el millor set de paràmetres a utilitzar amb el còdec h264. Utilitzeu el ffmpeg per generar aquest vídeo i comproveu visualment que la qualitat de la imatge és l'adequada.

```
ffmpeg -benchmark -r 25 -i ./data/raw/Cubo/Cubo%02d.png -g 22 -refs 0 -r 25 -cmp satd  
-me_method umh -me_range 64 -codec:v h264 ./cubo_h264_g100_ref0_r25.avi
```