

Sistemas Distribuidos y Programación Paralela

Codificación de Video

Principios básicos de Video

- Una secuencia de vídeo es una sucesión de imágenes que producen sensación de movimiento.
- El ojo humano ante una sucesión rápida de imágenes tenemos la percepción de un movimiento continuo (persistencia de la retina, físico y etimólogo inglés Peter Mark Roget (1779-1869)).
- Una cámara de cine/vídeo no es otra cosa que una cámara de fotos que "saca fotos muy rápido".

¿Por qué comprimir?

- Para reducir el volumen de datos a ser transmitidos (texto, imágenes, audio, video)
- Para reducir el ancho de banda requerido para la transmisión
- Para reducir el espacio de almacenamiento necesario

¿Por qué comprimir?

Una película sin compresión, con:

- 720 x 480 pixels por cuadro
- 30 cuadros por segundos
- Duración Total de 90 minutos

Implica una cantidad de datos de = 167.96 G Bytes !!

¿Cómo es posible la compresión?

La compresión se hace posible a partir de:

- Redundancia en la información que hay en el audio y video digital
- Propiedades de la percepción humana.

Clasificación de compresión

- **Lossless (compresión sin pérdidas)**

Útil cuando no es posible descartar información y sólo puede explotarse únicamente la redundancia de datos

- **Lossy (compresión con pérdida)**

En material como audio digital, imagen, video donde algunos errores o pérdidas pueden tolerarse, se puede explotar tanto la redundancia de datos como las propiedades de la percepción humana

Example of Lossy Compression



¿En qué se basa la compresión de video?

... en eliminar la redundancia:

- Intraframe Coding (espacial):**

Remueve la redundancia espacial. Cada imagen es tratada como una imagen independiente, y codificada independientemente del contexto. Al contener la información completa de la imagen, es el tipo de imagen elegido como primera imagen en una secuencia de Inter-Codificación, el proceso se realiza a través de coeficientes DCT

- Interframe Coding (temporal):**

Remueve la redundancia temporal. Comprime secuencias de marcos por solamente el almacenamiento de diferencias entre ellos, a través del registro de desplazamiento del objeto. Esto se realiza sobre la base de compensación de movimiento (MC).

Parámetros de codificación básicos

Compensación de Movimiento:

La **compensación de movimiento** es una técnica utilizada en la codificación de vídeo, cuyo principal objetivo consiste en eliminar la redundancia temporal existente entre las imágenes que componen una secuencia, con el fin de aumentar la compresión.

Para ello:

- 1) Existe un algoritmo que examina la sucesión de fotogramas consecutivos
- 2) Detección, análisis y estimación de movimiento entre fotogramas
- 3) Selección de zona de codificación (sólo partes en movimiento)
- 4) Predicción -> Compensación de movimiento basándose en las imágenes anteriores.

Parámetros de codificación básicos

Tipos de Cuadros:

Frames I (Intracodec) -> No tiene predicción con cuadros anteriores y posteriores. Es un frame intracodificado, siendo una imagen completamente construida (se codifica independientemente). También son los conocidos como Keyframes (parámetros de codificación)

Frames P (predictive-coded) -> Tiene predicción con cuadros anteriores para descomprimir. Es decir, la imagen solo registra los cambios respecto de la imagen anterior. Su referencia puede ser a cuadros I y P anteriores. El tamaño promedio de un frame P es de $\frac{1}{3}$ del frame I

Frames B (Bi-directional predictive-coded)-> Utiliza cuadros anteriores y posteriores para alto nivel de compresión. Como se puede imaginar, los perfiles de alto nivel utilizan algoritmos de codificación más complejos y producirán una mejor calidad de vídeo a una velocidad de transferencia equivalente. Esto implica la necesidad de un hardware más potente para codificar / decodificar. El tamaño final es de $\frac{1}{6}$ de un I frame.

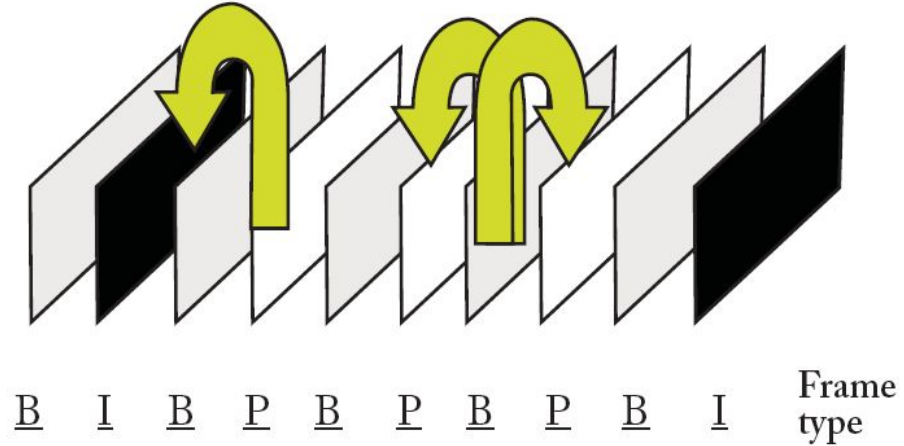
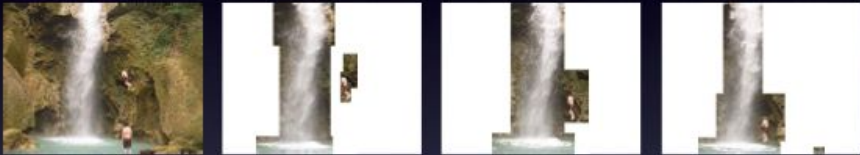
Parámetros de codificación básicos

Tipos de Cuadros:

Full I-Frames



Compression using P-Frames



Parámetros de codificación básicos

GOP (Group of Pictures)

- Es un conjunto de cuadros (frames) consecutivos que pueden ser decodificados sin referencia a otro frame (por lo general 8, 12 o 15 frames)
- Se comienza con un frame I. La estructura GOP suele estar referenciada por dos números, por ejemplo, $M=3, N=12$. El primero de ellos nos dice la distancia que hay entre dos imágenes tipo I o P. El segundo nos dice la distancia que hay entre dos imágenes enteras, es decir, entre dos imágenes tipo I: es la **longitud del GOP**
- Su estructura típica es I-B-B-P-B-B-P. El I-frame se utiliza para predecir la primera trama P y estos dos también se utilizan para predecir la primera y la segunda trama B. El segundo fotograma P se predice usando la primera P y que se unen para predecir las tercera y cuarto fotogramas B. Siempre un cuadro I (Completo) arranca la secuencia GOP.
- Cuantas más imágenes tipo I haya en un stream de vídeo más fácil será su edición, pero en contraposición este stream ocupará más tamaño. Para ahorrar ancho de banda y espacio en el disco, los vídeos preparados para su difusión en Internet, solo tienen una imagen tipo I por GOP.

Parámetros de codificación básicos

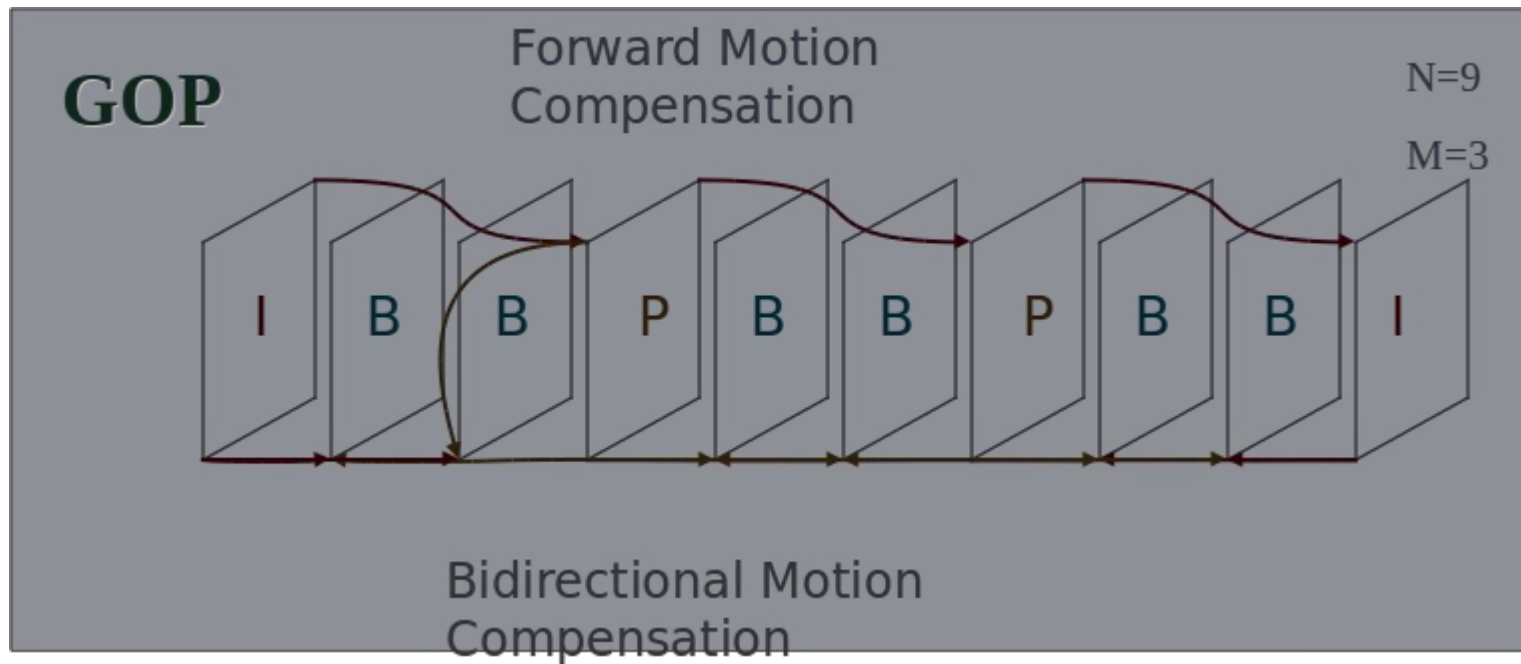


Imagen inicial (I Frame) ->



Y que los cuadros siguientes son:

Sin codificar, los
siguientes frames
serían ->



La información Inter-Frame que se guardaría sería:

Usando P frames ->



Parámetros de codificación básicos

Perfiles de Codificación H.264:

Baseline

(Línea base): Este perfil es usado cuando el video es liviano como en el caso de las videoconferencias o reproducción en teléfonos celulares con limitaciones en poder de cómputo. Provee la compresión menos eficiente entre las tres opciones y el consumo más bajo de CPU en la decodificación. No existen frames B

Main

(Principal): Este perfil es más capaz que Baseline, lo que generalmente se traduce en mayor eficiencia, si bien trae unido un mayor demanda de poder de procesamiento (menos que el perfil High). Este perfil se usa en los videos de calidad media para las aplicaciones de video en la web

High:

(Alto): El perfil principal para la transmisión y almacenamiento en disco. Particularmente es utilizado en aplicaciones de televisión de alta definición (este es el perfil adoptado en HD DVD y Blu-ray Disc, por ejemplo).

Parámetros de codificación básicos

Niveles:

Los **Perfiles** (baseline, main, high) abordan el problema de la complejidad del código y potencia de procesamiento.

Los **niveles** abordan el problema de ancho de banda, resolución máxima, y los problemas de memoria en el lado del decodificador.

Cualquier dispositivo dado soporta un perfil y un nivel en función de la memoria y su resolución máxima disponible. Los niveles más bajos significan resoluciones más bajas, bitrates bajos y uso de memoria inferior

L@	L@	L@	L@	L@
1	2	3	4	5
1.1	2.1	3.1	4.1	5.1
1.2	2.2	3.2	4.2	
1.3				

Los niveles definen:

- Máximo de macrobloques
- Máximo de Tamaño de Frame
- Máximo video bitrate
- Resoluciones soportadas

LVL 1:

Max Video: 768 kbit/s

Max Resol: 352×288

LVL 2:

Max Video: 4 Mbit/s

Max Resol: 720×576

LVL 3:

Max Video: 20 Mbit/s

Max Resol: 1280×1024

LVL 4:

Max Video: 50 Mbit/s

Max Resol: 2048×1088

LVL 5:

Max Video: 240 Mbit/s

Max Resol: 4096×2304

Llevando todo esto a la realidad

Streaming Adaptativo:



Llevando todo esto a la realidad

Streaming adaptativo -> **Realidad:**

- **Múltiples dispositivos**
 - <> Tamaño de Pantalla
 - <> Arquitectura y potencia de cálculo
 - <> Soporte a codecs
- **Ancho de banda disponible** (Wifi, 2G, 3G, 4G LTE, etc)

Entonces necesito:

- **Varios streams** (ABR), codificados en **varios perfiles** (baseline, main, high) dependiendo de las características y soporte disponible.

Parámetros de codificación básicos

- **Bitrate**

Información almacenada/transmitida por unidad de tiempo

Usualmente medida en Kbps o Mbps (dependiendo del tipo de servicio). El rango puede ir desde pocos kbps (< 1Mbps) a mayores de 40 Mbps

- **Resolución**

Número de píxeles por cuadros (Frames)

comienza desde 160x120 (móvil) hasta 4K (Smart TV)

Ejemplo:
Pantalla con
resolución
720 X 480

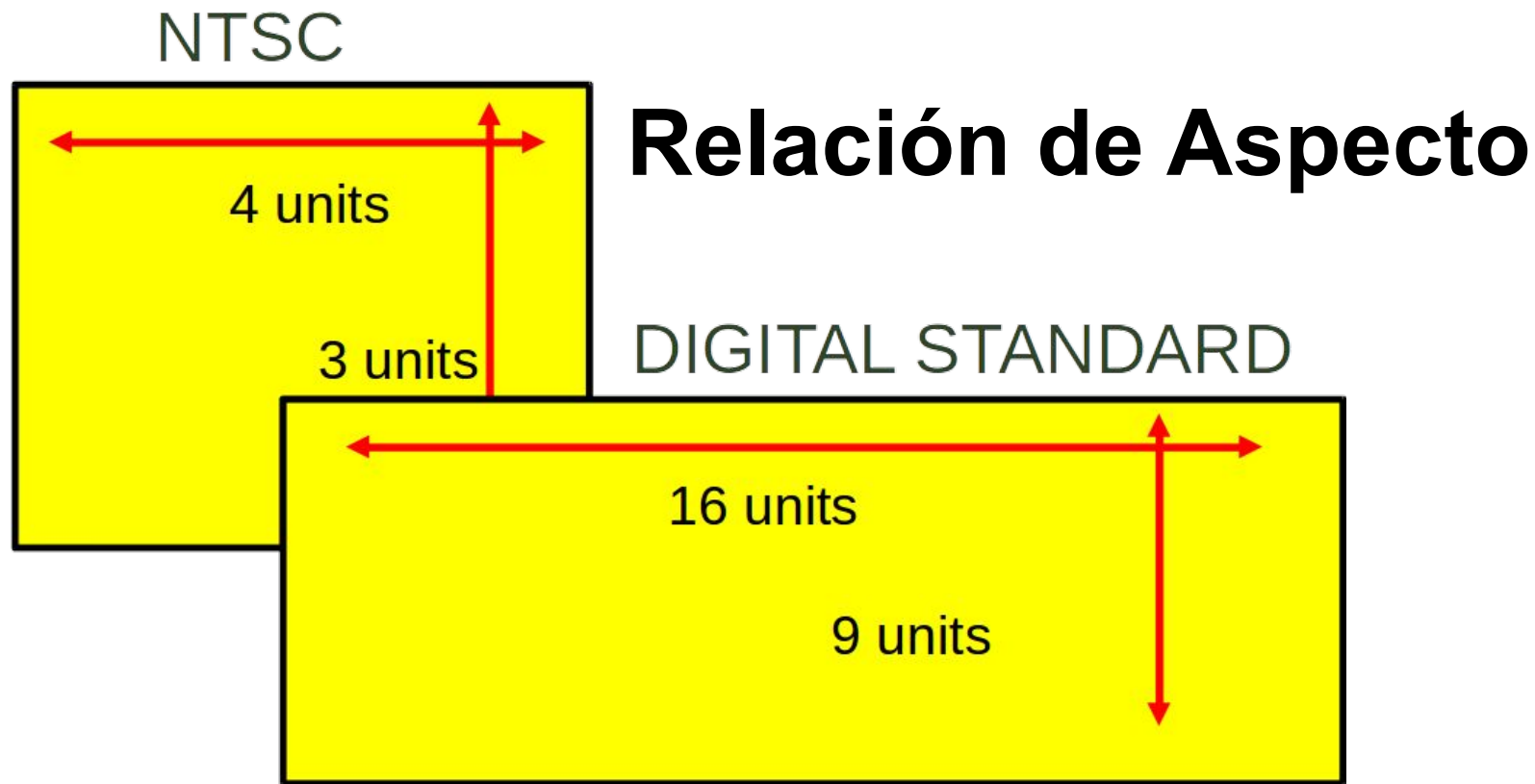
720
x480
345,600
pixels

- **FPS (frames per second) -> Cuadros por segundo**

Usualmente 15, 24, 25, 30, o 60

A partir de 60, se desprecia la ventaja obtenida debida a las limitaciones del ojo humano

Parámetros de codificación básicos



Llevando todo esto a la realidad

En base a un relevamiento real, para brindar un servicio de streaming que alcance todos los dispositivos sería necesario lo siguiente:

- Codificar en **BASELINE**, para los dispositivos antiguos y con poca potencia de cálculo (<2010)
- Codificar en **MAIN**, para los dispositivos de rango medio (2011+), para Webcast y Streaming de video a través de internet
- codificar en **HIGH**, para dispositivos actuales (2013+), y streaming en 2K y 4K con calidad similar a la televisión.

Llevando la realidad a FFMPEG (x264)

Ejemplo de codificación **BASE**

```
ffmpeg -y -i part_5.mp4 -pass 1 -s 320x180 -aspect 16:9 -c:v libx264 -g 50 -b:v 220k -profile:v baseline -level 3.0 -r 15 -preset ultrafast -threads 0 -c:a aac -strict experimental -b:a 64k -ar 44100 -ac 2 -f mp4 /dev/null &&  
ffmpeg -y -i part_5.mp4 -pass 2 -s 320x180 -aspect 16:9 -c:v libx264 -g 50 -b:v 220k -profile:v baseline -level 3.0 -r 15 -preset ultrafast -threads 0 -c:a aac -strict experimental -b:a 64k -ar 44100 -ac 2  
part_5_perfil1base.mp4
```

Donde:

- y Sobreescribe el archivo resultante (sin preguntar)
- i recurso fuente
- pass número de pasada (1, 2 o n). 1 para estimar, 2 para comprimir en base a estimación, uso en ABR
- r -> Frames per sec. (cuadros por segundo)
- s -> Resolución (Tamaño de Pantalla)
- aspect -> Aspecto 4:3 o 16:9 (primer valor -> Ancho ; segundo valor alto)
- c:v -> Librería de encoding de video (En este caso x264)
- b:v -> Video Bitrate (ABR) -> Es un promedio. Donde hay poca info encodeo a baja tasa de bits y reservo para donde hay mucha información, dando mayor cantidad de bits y manteniendo calidad. Necesito 2 pasadas.
- profile:v -> Perfiles (baseline, main, high)
- level -> Nivel de compresión
- qp -> Encodeo sin pérdida

Llevando la realidad a FFMPEG (x264)

Ejemplo de codificación **BASE**

```
ffmpeg -y -i part_5.mp4 -pass 1 -s 320x180 -aspect 16:9 -c:v libx264 -g 50 -b:v 220k -profile:v baseline -level 3.0 -r 15 -preset ultrafast -threads 0 -c:a aac -strict experimental -b:a 64k -ar 44100 -ac 2 -f mp4 /dev/null &&  
ffmpeg -y -i part_5.mp4 -pass 2 -s 320x180 -aspect 16:9 -c:v libx264 -g 50 -b:v 220k -profile:v baseline -level 3.0 -r 15 -preset ultrafast -threads 0 -c:a aac -strict experimental -b:a 64k -ar 44100 -ac 2  
part_5_perfil1base.mp4
```

```
ffmpeg -y -i bigbuckbunny_1500.mp4 -s 3096x2048 -aspect 16:9 -c:v libx264 -g 50 -b:v 22000k -profile:v high -level 5.1 -r 60 -preset slow -threads 0 -c:a aac -strict experimental -b:a 256k -ar 48000 -ac 6  
part_5_perfil1base.mp4
```

Donde:

- bf -> Máximo consecutivo de frames b (no en baseline)
- b_strategy -> 0 (sin ubicación dinámica); 1 (ubicación rápida, rápido pero poco eficiente); 2 (ubicación lenta pero eficiente)
- refs -> Número de frames referenciados, la recomendación es no mayor a 6
- preset -> Son un conjunto de opciones que proveen la velocidad de encodeo a un determinado nivel de compresión (ultrafast a veryslow)
- threads -> Asignación de cores al proceso. 0 automático -> FFMPEG decide los threads. Caso contrario, tengo que asignarlos yo en base a mi decisión

Fin!

