

## Processament de Vídeo: Seminari 2

Github: [https://github.com/dbarreropl/Video\\_Encoding\\_S2](https://github.com/dbarreropl/Video_Encoding_S2)

He integrat tots els exercicis en un script de python el qual permet seleccionar l'opció que volem fer, de l'1 al 4, o sortir de l'aplicació amb el 5.

```
1. Macroblocks and Motion Vectors
2. Extract Video and Audio and package into an .mp4
3. Which broadcasting standards are compatible
4. Add subtitles
5. Exit
Option:
```

### Exercici 1:

Pel primer exercici per tal de mostrar els macroblocks i els motion vectors he utilitzat la comanda de la documentació de ffmpeg:

<http://trac.ffmpeg.org/wiki/Debug/MacroblocksAndMotionVectors>

### Exercici 2:

Per tal d'agafar 1 minut de vídeo utilitzem la comanda de la pràctica anterior però per no agafar l'àudio utilitzem "-an":

<http://ffmpeg.org/ffmpeg.html#Audio-Options>

Ara agafem només l'àudio en mp3 i en aac, utilitzo "-b:a 128k" per codificar en un baix bitrate.

Finalment ho fem tot dins d'un container mp4, utilitzem "-map" per indicar la posició on ha de guardar cada input, primer el vídeo i després els dos àudios.

<https://trac.ffmpeg.org/wiki/Map>

També utilitzem "-c copy" per copiar els fitxers, ja que no fa falta torna a codificar. També amb "-metadata:s:a:0" es pot indicar un títol i altres atributs interessants com l'idioma de l'àudio.

Amb "ffmpeg -i vídeo.mp4" podem veure que el container conté el vídeo h264, i àudios mp3 i aac correctament:

```
Stream #0:0[0x1](und): Video: h264 (High) (avc1 / 0x31637661), yuv420p(progressive), 1920x1080 [SAR 1:1 DAR 16:9], 3308 kb/s, 30 fps, 30 tbr, 15360 tbn (default)
Metadata:
  handler_name      : GPAC ISO Video Handler
  vendor_id         : [0][0][0][0]
Stream #0:1[0x2](und): Audio: mp3 (mp4a / 0x6134706D), 48000 Hz, stereo, fltp, 128 kb/s (default)
Metadata:
  handler_name      : SoundHandler
  vendor_id         : [0][0][0][0]
Stream #0:2[0x3](und): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, 5.1, fltp, 393 kb/s
Metadata:
  handler_name      : SoundHandler
  vendor_id         : [0][0][0][0]
```

### Exercici 3:

Ara per seleccionar els standards de broadcast que son compatibles necessitem primer tenir quin es el còdec que utilitzen. Així utilitzem la comanda trobada:

<https://stackoverflow.com/questions/2869281/how-to-determine-video-codec-of-a-file-with-ffmpeg>

Per tal d'utilitzar aquest valor al codi creem un fitxer temporal que guarda els resultats, el llegim i guardem els valors a la variable "codecs", i eliminem l'arxiu.

Ara només hem de comparar els códecs compatibles per cada standard i al final printarem els compatibles o si no hi ha cap que es pugui utilitzar.

### Exercici 4:

Finalment en aquest exercici descarregarem l'arxiu de subtitlles amb la comanda "wget", ho guardarem i l'aplicarem al vídeo, després eliminem l'arxiu:

<https://stackoverflow.com/questions/8672809/use-ffmpeg-to-add-text-subtitles>