



EQUIPOS Y SISTEMAS DE AUDIO Y VIDEO (ESAV)

PRÁCTICA VÍDEO

CURSO 2020-2021

Elena María del Río Galera
70065724-L
em.delrio.2016@alumnos.urjc.es

Índice

Medidas de calidad de vídeo.....	2
----------------------------------	---

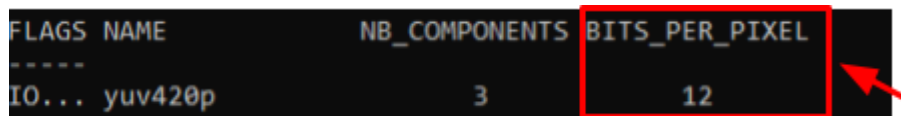
❖ Ejercicio 1. Conversión de vídeo a formato YUV.....	2
❖ Ejercicio 2. Medidas de calidad de vídeo.....	6
❖ Conclusiones	11

PRÁCTICA 4: MEDIDAS DE CALIDAD DE VÍDEO

Ejercicio 1. Conversión de vídeo a formato YUV.

1.1. Ejecute el comando `ffmpeg -pix_fmts` y localice el formato `"yuv420p"`. ¿Con cuántos bits por pixel trabaja dicho formato?. Acompañe la respuesta con una captura de pantalla de la consola de comandos, señalando la información. ¿Cuántos bits por componente utiliza? Desglose la cantidad de bits por componente razonando su respuesta.

Se ha ejecutado el programa, a continuación en la *Figura 1*, se muestra el detalle del yuv420p. Este formato está trabajando con 12 bits por pixel.



FLAGS	NAME	NB_COMPONENTS	BITS_PER_PIXEL
IO...	yuv420p	3	12

Figura 1. Localización del formato "yuv420p".

Estamos con el formato yuv, que es un sistema de señal de video tiene dos componentes: luminancia (Y) y crominancia (UV). La pregunta que tenemos que hacernos es ¿Cuántos bits se destinan a la luma y cuantos al croma?

Sabemos que la luminancia nunca se submuestra, por lo que tendremos 8 bits por la componente de luma.

El croma, se está sub muestreando a 4:2:0, es decir $\frac{1}{4}$ de la imagen original, $\frac{1}{4} * 8$, lo que es lo mismo, 2 bits por componente de croma.

1.2. Ahora convierta el vídeo al formato `"yuv"` con un submuestreo de croma de 4:2:0 ejecutando la siguiente línea en la consola de comandos:

`ffmpeg -i video_HD.mp4 -c:v rawvideo -pixel_format yuv420p video_HD.yuv`.

Sabiendo el número de bits por pixel y la cantidad de frames que componen el video, calcule el tamaño en bytes que ocupa el video descomprimido (`"video_HD.yuv"`), desglosando todos los valores que intervienen en el cálculo. Posteriormente, calcule el ratio de compresión entre el video comprimido (formato `"mp4"`) y el video sin descomprimir (`"formato yuv"`).

Se ha ejecutado el comando en el terminal, para convertir el vídeo al formato deseado, se muestra en la *Figura 2*, el resultado general, y un detalle donde se ven los frames que se han convertido. Son 1193 frames.

Ahora vamos a calcular el tamaño del vídeo en bytes, tenemos 12 bits por pixel, y 1193 frames. Estamos en calidad HD, calculamos los pixeles de esta calidad, **1280x720 = 971600 pixeles**. Calculamos los bits de un solo frame, 12 bits por pixel multiplicado por los pixeles totales, **12 * 971600 = 11059200 bits**. Ahora calculamos los bits de todos los frames en conjunto, **11059200 * 1193 = 13193625600 bits de todos los frames**. Y el último paso es pasar los bits a bytes, 1 byte son 8 bits. Hacemos la conversión:

$$13193625600 \text{ bits} \cdot \frac{1 \text{ byte}}{8 \text{ bits}} = 1.649.203.200 \text{ bytes}$$

```
Frame= 1193 624 q=0.0 lsize= 1610550kB time=00:00:39.80 bitrate=331444.6kb/s speed=21.2x
video:1610550kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: 0.000000%
```

En cuanto a la conversión realizada, para comprobar, se han mirado las propiedades del archivo video_HD.yuv en el sistema. En la Figura 3, podemos comprobar como los cálculos realizados son correctos y el número de bytes del vídeo sin comprimir coincide con la captura.

Figura 3. Bytes de video HD.yuv en propiedades.

$$1.649.203.200/14.648.108 = 112.58$$

1.3. Ejecute la aplicación “pYUV” y abra el archivo en formato “.yuv” que ha creado anteriormente. En la pantalla de configuración que aparece cuando abra el video, seleccione la resolución del vídeo correcta, el “frame rate” y el submuestreo que se ha aplicado. Adjunte en la memoria una captura de la pantalla de configuración con los valores seleccionados y otra captura del primer frame del video. ¿Qué ocurre si configura la reproducción del video con una resolución distinta a la que tiene?. Justifique su respuesta adjuntando una captura de pantalla con el primer frame del video en esa situación.

Para escoger el frame rate adecuado se ha consultado en el terminal, en la Figura 4, se puede observar el fps de 29.97, aunque en la configuración final como se ve en la Figura 5, se ha redondeado a 30fps.

Nota: hay que tener cuidado de escoger el frame rate, ya que nos podría confundir el fps de la conversión.

```
Stream #0:0(und): Video: h264 (High) (avc1 / 0x31037661), yuv420p(tv, bt709), 1280x720, 2882 kb/s, 29.97 fps, 29.97 tbr, 30k tbn, 60k tbc (default)
Metadata:
  creation_time   : 2019-01-21T04:03:44.000000Z
  handler_name    : L-SMASH Video Handler
  vendor_id       : [0][0][0][0]
  encoder         : AVC Coding
```

Figura 4. Consulta de fps en el terminal.

La resolución sigue siendo HD, 1280x720, la conversión no afecta a esto. Y el submuestreo tampoco cambia, sigue siendo 4:2:0.

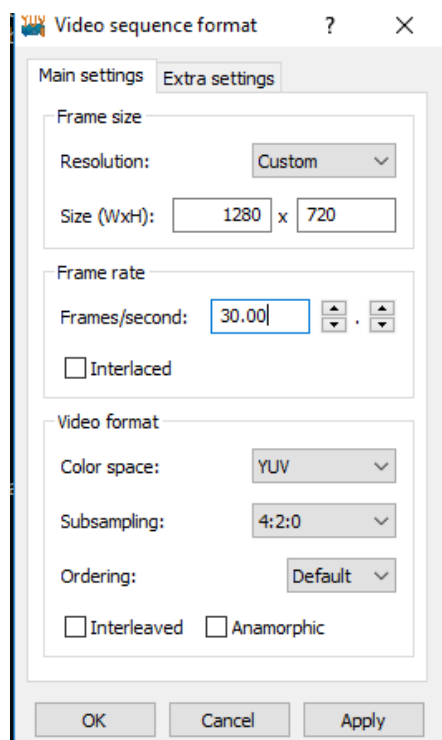
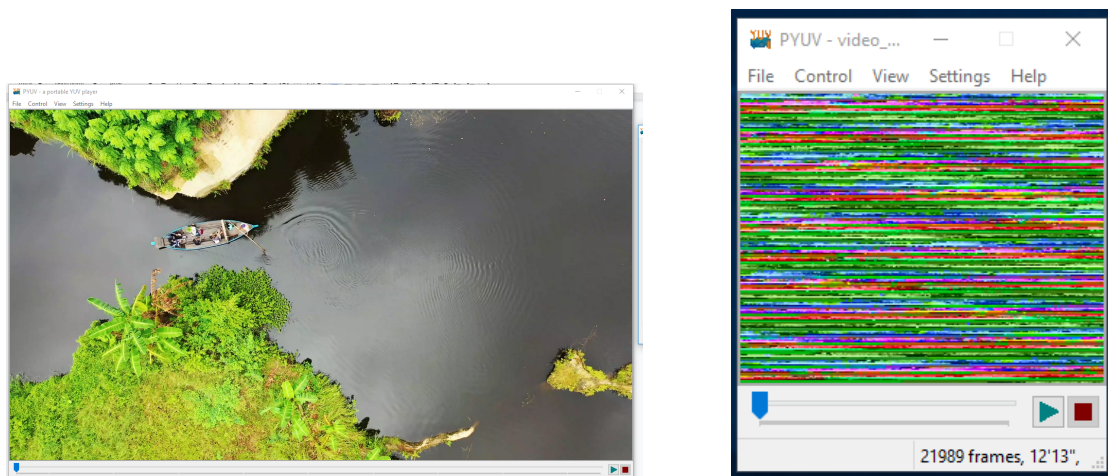


Figura 5. Configuración del vídeo en pYUV..

Si se configura con una resolución distinta lo que ocurre, es que no se ve el vídeo resultante. Se ha probado con una configuración aleatoria y en la figura seis se ha hecho un símil del vídeo con la configuración acertada, y la errónea.



Resolución HD, 1280x720

Resolución aleatoria de 250x200

Figura 6. Comparativa vídeo con resolución correcta y errónea

1.4. A continuación se va a proceder a reducir la resolución del vídeo (a resolución VGA) en formato “.yuv”. Para ello ejecute en la consola de comandos la siguiente línea, sustituyendo N y M por la resolución del vídeo en formato “.yuv” del que se parte; y P y Q por la resolución del video convertido:

`ffmpeg -s:v NxM -r 25 -i video_HD.yuv -vf scale=P:Q -c:v rawvideo -pix_fmt yuv420p video_VGA.yuv`
Cargue el nuevo vídeo que se ha creado en la aplicación “pYUV” y compruebe que se visualiza correctamente (adjunte captura de pantalla, tanto de la configuración de la carga del vídeo como del primer frame del mismo). Aparte de la resolución, ¿Qué otro parámetro ha cambiado, y a qué valor?. Pista: para resoluciones HD, FHD y UHD ese parámetro se mantiene constante.

Lo primero de todo es conocer la resolución de VGA, Video Graphics Array o Matriz de gráficos de vídeo, se utiliza para denominar a una pantalla estándar analógica de una computadora, su resolución es de 640 × 480 píxeles.

Después se ha ejecutado el siguiente comando:

`ffmpeg -s:v 1280x720 -r 25 -i video_HD.yuv -vf scale=640:480 -c:v rawvideo -pix_fmt yuv420p video_VGA.yuv.`

Ahora se carga el nuevo vídeo en pYUV, y se actualiza la configuración, los fps son 25, y la resolución 640x480.

Se puede ver tanto la salida como la configuración en la *Figura 7*. A parte de la resolución el otro parámetro que ha cambiado es la relación de aspecto. En HD, FHD Y UHD, se mantiene en 16:9, mientras que en este caso en VGA, la relación de aspecto ha pasado a ser 4:3.

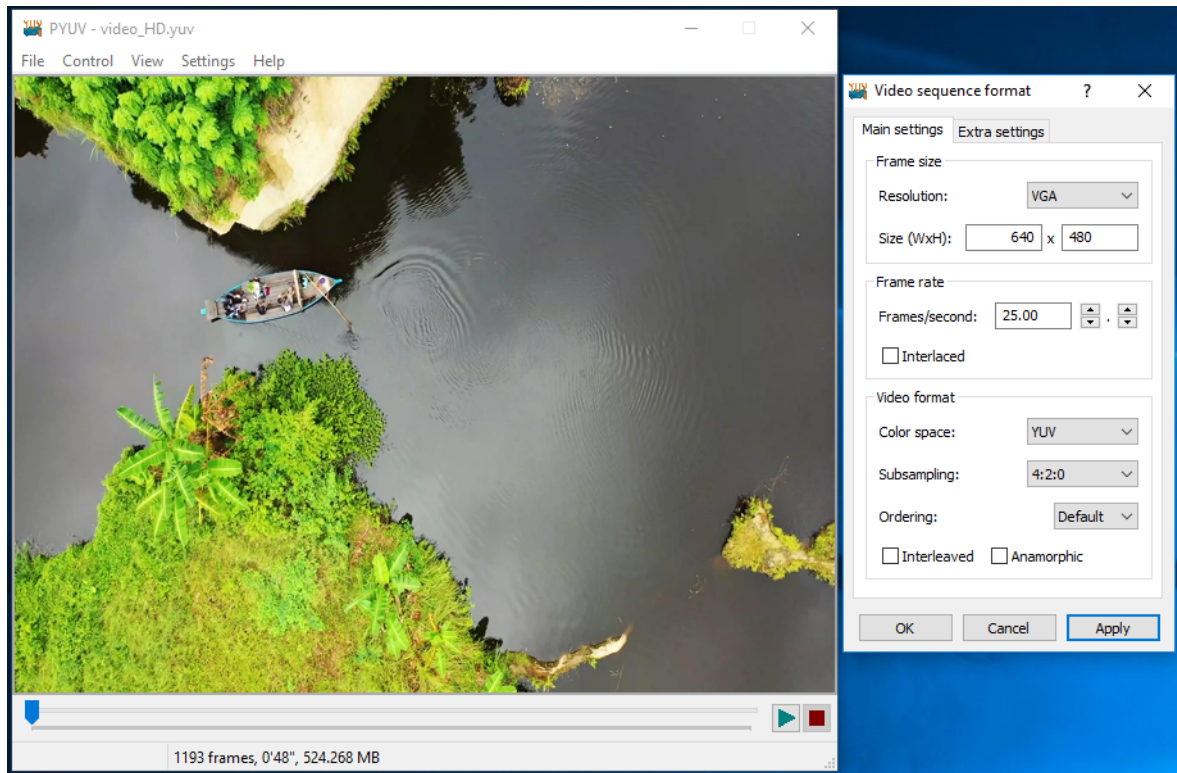


Figura 7. Primer frame vídeo VGA y configuración en pYUV.

Ejercicio 2. Medidas de calidad de vídeo.

1.1. ¿El resultado que se observa en el vectorscopio sobre el video de fondo rojo ("Video_test_rojo.mp4") entra dentro del rango legal, de acuerdo con la teoría? En caso de que el resultado esté fuera de rango, ¿Qué característica del color está fuera de ese rango?. Adjunte la imagen del vectorscopio y justifique su respuesta.

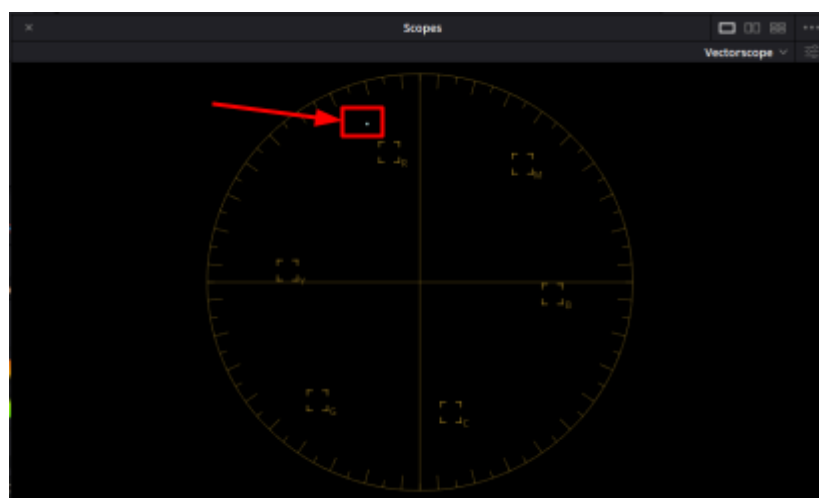


Figura 8. Vectorscopio con la información del vídeo rojo.

El resultado no se encuentra en un rango permitido, como se aprecia en la *Figura 8*, el punto blanco, correspondiente al video se encuentra fuera del cuadrado rojo. Por lo que no es un color legal.

En cuanto al vectorscopio, cuanto más cerca se encuentre un punto del centro, menos saturación. Y la fase, es decir el ángulo, se refiere al tono.

1.2. Utilizando la función “Offset” (circunferencia de color situada en la parte inferior izquierda de la aplicación), modifique el color del video para que en el osciloscopio aparezca en el punto de máxima saturación permitida y adjunte captura de pantalla con la configuración de “Offset” realizada.

Para que sea legal, se han cambiado los valores de offset hasta que el punto blanco ha quedado en el centro del cuadrado R, de rojo. En la *Figura 9* se pueden apreciar las modificaciones.

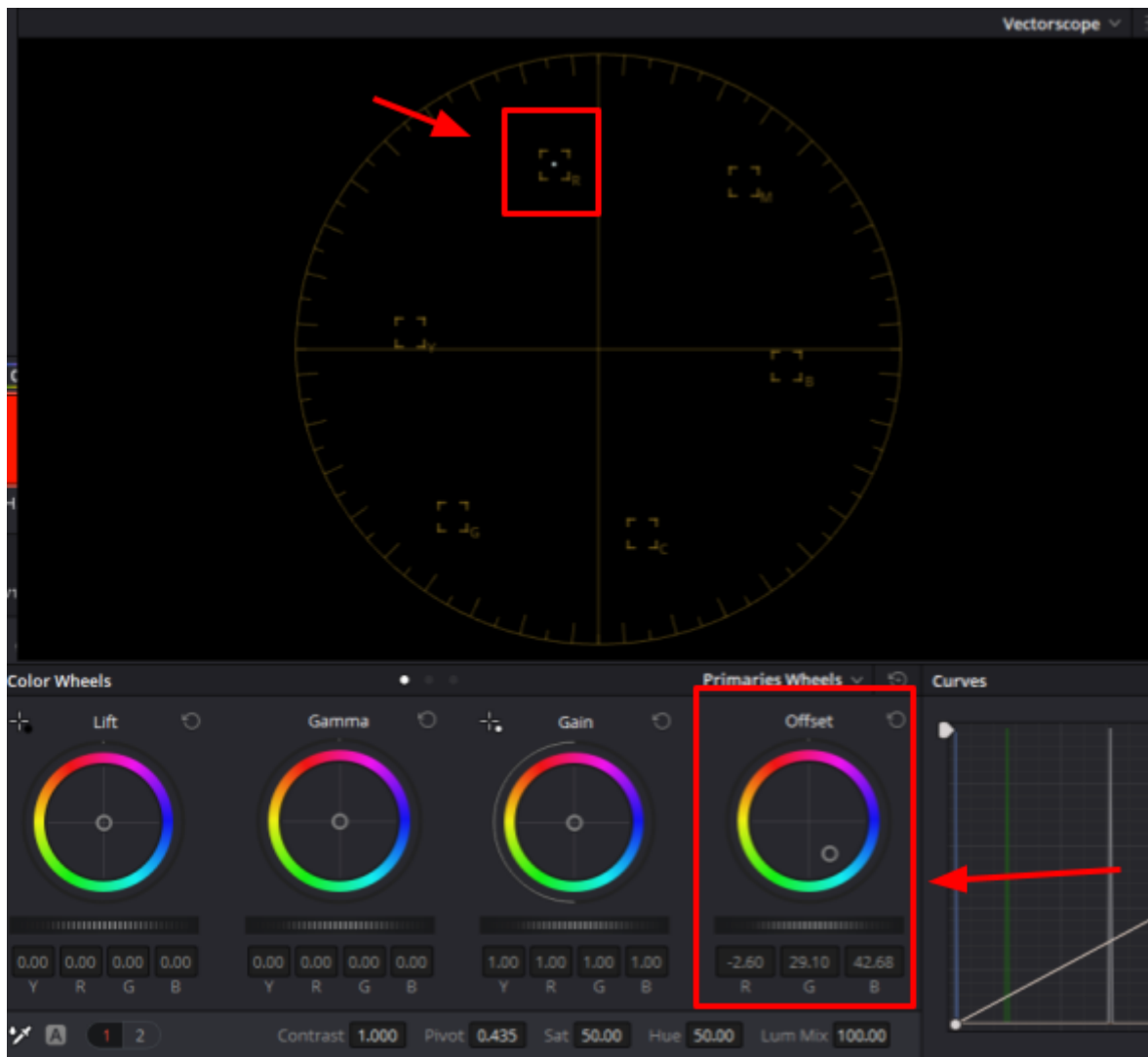


Figura 9. Vectorscopio y configuración de offset para color legal.

1.3. Analice con el programa “DaVinci Resolve” los videos de test (barras de colores) que se adjuntan y determine qué video/s de test es/son correcto/s. Adjunte para cada video una captura de la señal del vectorscopio y justifique cada video de test que sea válido.

El **vídeo 1** de barras de colores **no es válido**, Y, R y M se encuentran fuera de sus respectivos cuadrados como se ve en la *Figura 10*.

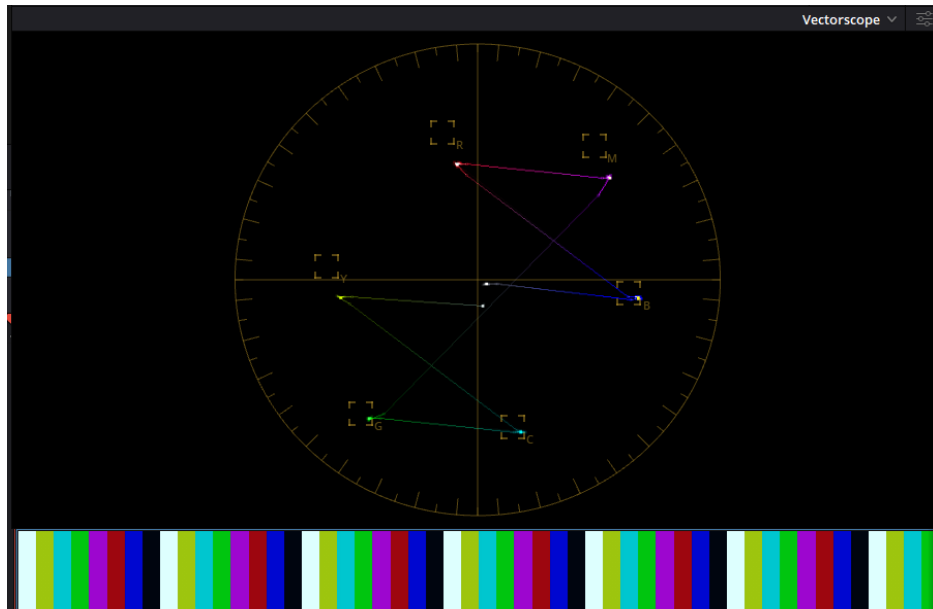


Figura 10. Vectorscopio vídeo barras colores 1.

El **vídeo 2** de barras de colores **no es válido**, Y, C, M, R, G y B, es decir todos, se encuentran fuera de sus respectivos cuadrados como se ve en la *Figura 11*.

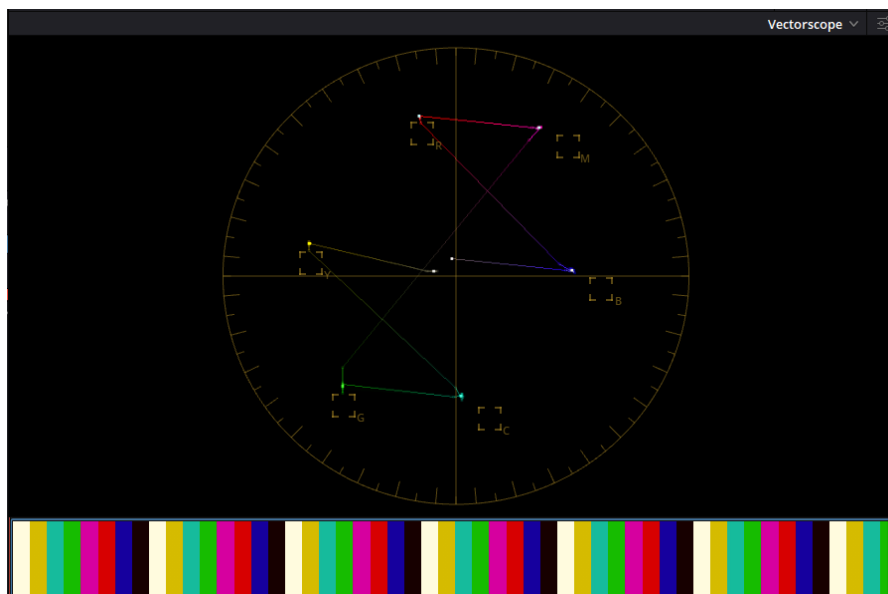


Figura 11. Vectorscopio vídeo barras colores 2.

El **vídeo 3** de barras de colores **sí es válido**, Y, C, M, R, G y B, es decir todos, se encuentran dentro de sus respectivos cuadrados como se ve en la *Figura 12*.

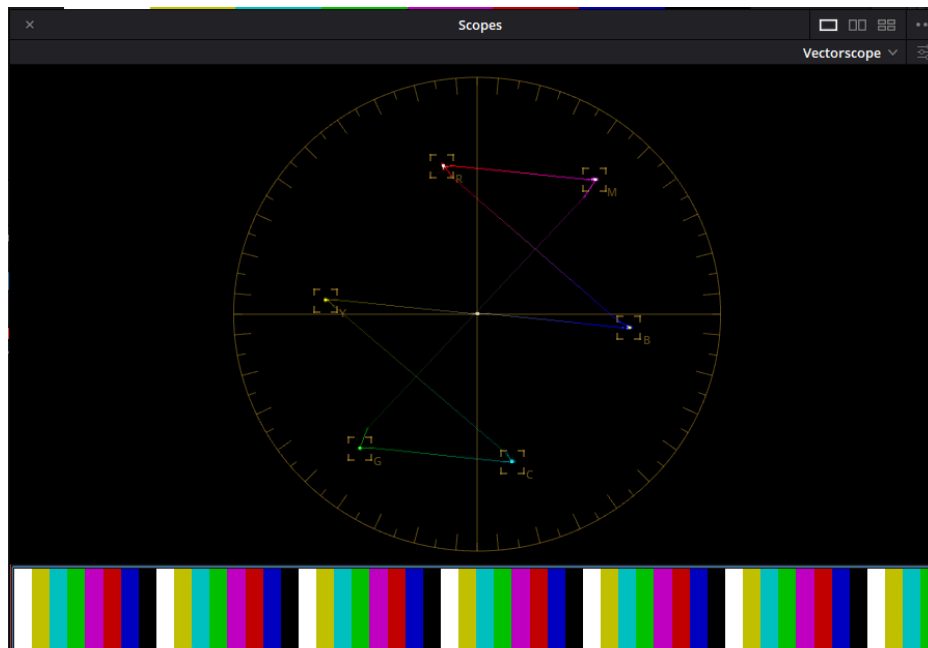


Figura 12. Vectorscopio vídeo barras colores 3.

El **vídeo 4** de barras de colores **sí es válido**, Y, C, M, R, G y B, es decir todos, se encuentran dentro de sus respectivos cuadrados como se ve en la *Figura 13*, aunque es cierto que algunos están más ajustados, por lo que el más correcto sería el vídeo 3.

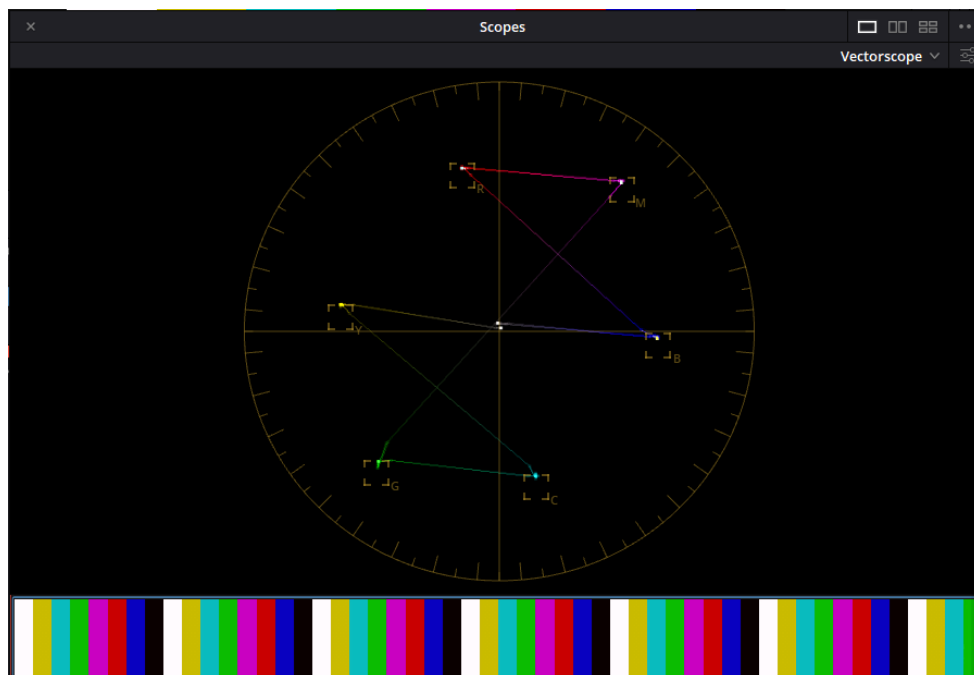


Figura 14. Vectorscopio vídeo barras colores 4.

1.4. Cierre y vuelva a abrir el programa “DaVinci Resolve” y después cargue el archivo “Video_fuegos_artificiales.mp4”. Ayudándose del vectorscopio, detenga el vídeo en un instante en el que el color predominante sea el rojo, de acuerdo con la lectura del vectorscopio. ¿Entraría dentro del nivel de saturación legal?. Adjunte una captura del fotograma así como de la ventana del vectorscopio.

En la *Figura 15*, se ve el vídeo de fuegos artificiales detenido en el instante de predominancia de color rojo. También se ve en el vectorscopio que el color rojo adquiere importancia en el frame seleccionado. Pero sin embargo, no estaría dentro del nivel de saturación legal, recordando lo que se ha expuesto en puntos anteriores, hacia el centro del vectorscopio cada vez hay menos saturación, así que el rojo de este frame está en un rango inválido por estar con una saturación más baja de lo que debería.

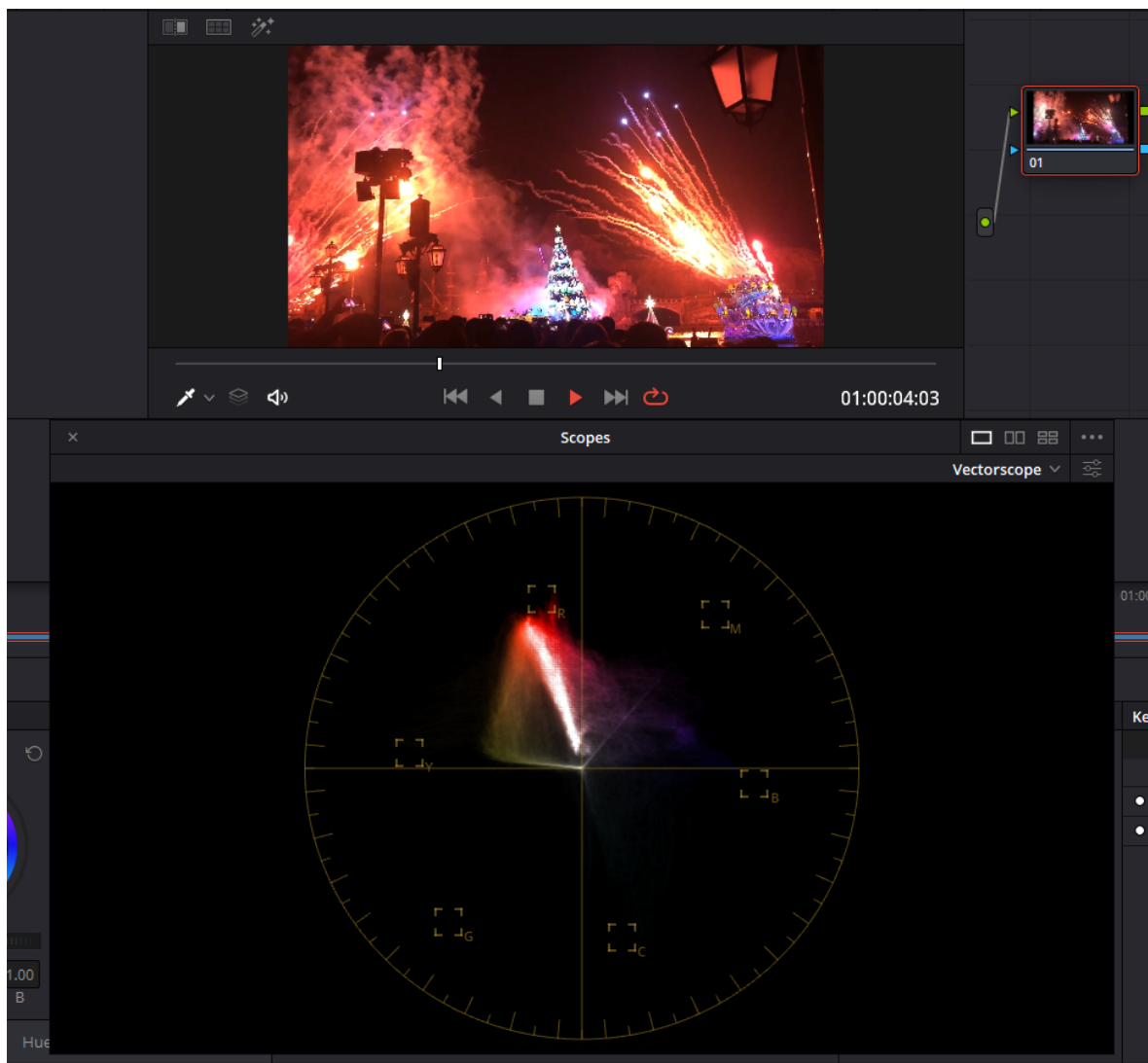


Figura 15. Vectorscopio vídeo fuegos artificiales con predominancia de rojo.

Conclusiones

En cuanto a la última práctica realizada de la asignatura se pueden sacar en claro varios aspectos.

Se han **fijado conceptos** ya conocidos como las resoluciones, las relaciones de aspecto de los vídeos, el tamaño en bits o bytes, así como además se ha refrescado el uso de la herramienta ffmpeg, que yo personalmente había usado el año pasado en la asignatura de estándares.

En cuanto a la segunda parte de la práctica, ha sido francamente interesante conocer el programa Davinci resolve, como una **herramienta profesional**, la mayoría de nosotros conoce y ha usado programas similares para el montaje y edición de vídeos caseros, (me refiero a programas de software como Movie maker, wondershare filmora o imovie). Pero ver como un software como Davinci puede ser tan potente y versátil ha sido un gran descubrimiento además de despertar las ganas de seguir probando e investigando con este programa por nuestra cuenta.

La conclusión general que se puede sacar es muy positiva, ya que el guión de prácticas permite un correcto seguimiento de la mismo, y un **aprendizaje**, así como **fijación de conceptos**. Se han disfrutado las prácticas por verles una aplicación real y directa. Además como ya he comentado líneas arriba la aclaración de conceptos que en esta práctica eran más familiares aunque nunca se habían visto de una manera tan pragmática.