



GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Compresión de Video

Prácticas de Aula. Bloque 1. Tema 2

Nombre: _____

Fecha: _____

1. En el directorio CarBW se encuentra un conjunto de B&W frames en ficheros separados con formato bmp que se desean juntar para formar una secuencia de video. Cada frame tiene un tamaño CIF de 288x352 pixels. El fichero Matlab **leerVideoCar.m** contiene una función con el código para leer todos esos ficheros y devuelve una variable **Ivid** que es una matriz 3D de dimensiones width*height*número de frames conteniendo el video compuesto.

Fichero: **leerVideoCar.m**

```
function Ivid=leerVideoCar()
%% Función que lee todas las frames del directorio CarBW
%% Retorna una matriz Ivid(M,N,numero de frames)

numFrame=1; % Inicializacion

% Lazo que recorre todas las frames del directorio ./CarBW
while (exist(sprintf('./CarBW/car%d.bmp',numFrame), 'file')== 2)
    Itemp(:, :, numFrame)=imread(sprintf('./CarBW/car%d.bmp',numFrame));

    numFrame=numFrame+1; % Incremento de frame
end

Ivid=Itmp; %Salida
```

Para reproducir el video ejecutar el siguiente código:

```
I=leerVideoCar(); % Se compone el video
imshow(I); % se reproduce el video
```

2. Proponer una modificación del anterior video para que se reproduzca previa aplicación de un proceso de detección de contornos con algoritmo 'canny'. Reproducir el video.
3. Se desea poner en color rojo el video B&W leído en la función **leerVideoCar.m**. Para ello hay que transformar cada frame monocromática en tres frames correspondientes a cada componente RGB.

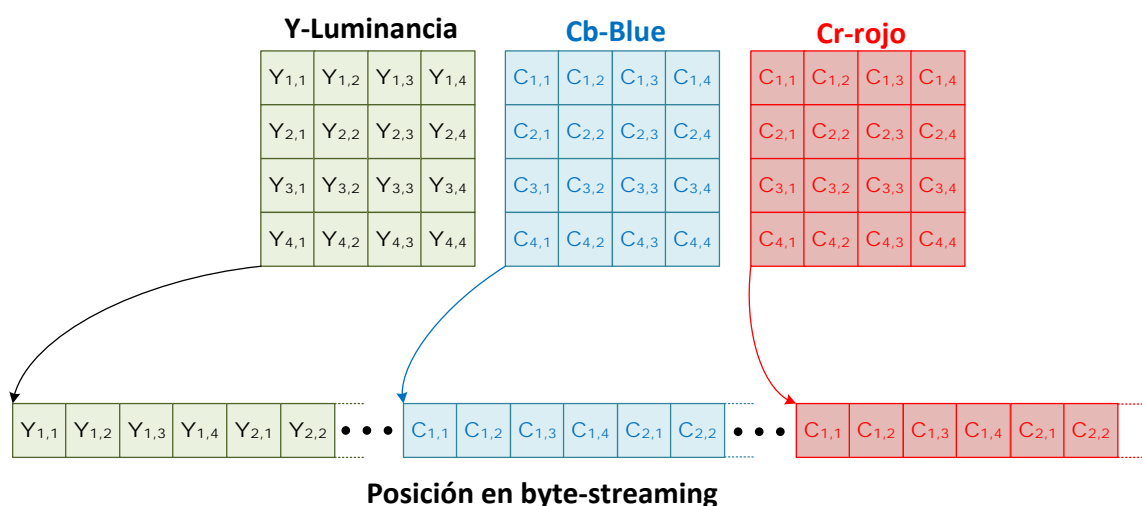
☺ Para generar una matriz nula usar la orden **zeros(M,N)**

4. Repetir el apartado 3 pero en color amarillo.
5. Modificar el código `leerVideoCar.m` y crear un nuevo fichero denominado `leerVideoCarColor.m` que permite leer la misma secuencia pero en formato color que se encuentra en el directorio CarColor. Hay que tener en cuenta que:
 1. En el fichero `leerVideoCarColor.m` la función se denomina `leerVideoCarColor()`.
 2. Esa función va a retornar una matriz 4D de dimensiones `width*height*3*número de frames`.
 3. Para reproducir el video en color, ejecutar:

```
K=leerVideoCarColor (); % Se compone el video
imshow(K); % se reproduce el video
```

6. Una vez obtenido el código de `leerVideoCarColor.m`, a continuación se describe el código contenido en el fichero `videoCarRGB2YUV444.m` que permite guardar en un fichero el video en formato YCrCb y poder ser visualizado con un reproductor de video externo.

En primer lugar se lee la secuencia de video en color para que se almacene en la variable `Ivid`. A continuación usando la función `rgb2ycbcr` se convierte en formato YUV y se almacena en la variable `IcrCb`. Finalmente se guarda la luminancia Y (luma) y crominancias (cromas) Cb y Cr en un fichero denominado 'car444.yuv'. En el proceso de escritura se sigue un formato de byte-streaming, siguiendo el siguiente orden: matriz Y, matriz Cb y matriz Cr. Gráficamente, este proceso se muestra a continuación:



Observar que en formato 4:4:4 no se ahorra espacio respecto al formato RGB al no aplicarse ninguna reducción en el tamaño de las cromas.

Fichero: videoCarRGB2YUV444.m

```
%% Conversion de RGB a YUV
Ivid=leerVideoCarColor(); % Se lee el video y se almacena en Ivid

% Se lee el número de frames
[height width dim NumberOfFrames]=size(Ivid);

% Se convierte a YUV frame a frame
for numFrame=1:NumberOfFrames
    YcrCb(:, :, :, numFrame)=rgb2ycbcr(Ivid(:, :, :, numFrame));
end

%% Proceso de escritura de YcrCb en un fichero
fid = fopen('car444.yuv', 'w'); % Se abre un fichero para escribir
for numFrame=1:NumberOfFrames
    % Se escribe la luma
    fwrite(fid, transpose(YcrCb(:, :, 1, numFrame)), 'uint8');
    % Se escribe la Cr
    fwrite(fid, transpose(YcrCb(:, :, 2, numFrame)), 'uint8');
    % Se escribe la Cb
    fwrite(fid, transpose(YcrCb(:, :, 3, numFrame)), 'uint8');
end

fclose(fid); % Se cierra el fichero
```

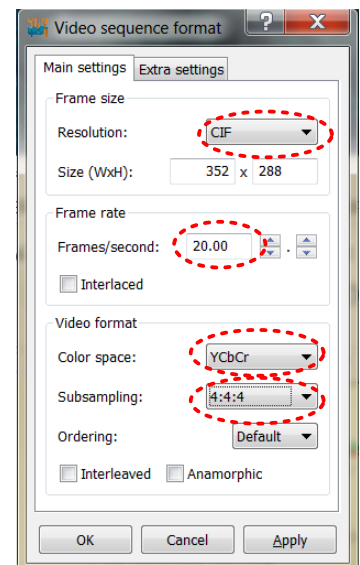
Para poder visualizar el fichero 'car444.yuv' se utilizará el



reproductor **pyuv.exe**. Al arrancar esta herramienta, se abre en primer lugar el fichero yuv que se desea visualizar: *File*→*Open* (Ctrl+O); en nuestro caso 'car444.yuv'. Se despliega una ventana donde hay que modificar los siguientes parámetros:

- Formato CIF: 352x288 pixels
- Frame rate: 20 fps
- Color space: YCbCr
- Subsampling: 4:4:4

Pulsar Ok y visualizar el video.



7. Observar que sucede si se elimina la operación **transpose** en el anterior código.
8. Modificar el código del anterior fichero para que se guarde la imagen en formato submuestreo 4:2:0. Este formato afecta solo a las componentes de crominancia de forma que hay que redimensionar esas matrices a la mitad. Para ello utilizar la función de Matlab

```
I420=imresize(I,0.5);
```

En este ejemplo, la función **imresize** hace un escalado de 0.5 de la matriz **I** y devuelve una matriz **I420** de tamaño mitad.

Utilizando esta función, crear un fichero denominado 'car420.yuv' en donde las cromas deben ser escaladas a la mitad. Visualizar este fichero con el reproductor



9. Existe la posibilidad de visualizar simultáneamente ambos videos, 'car444.yuv' y 'car420.yuv', utilizando otra herramienta de visualización: **yuvplayer.exe**. Arrancar yuvplayer.exe y seleccionar *File→Open Video 1* el fichero car444.yuv. Los parámetros a modificar son: *Video Size* es CIF(362x288) y *Video Format* 444. En *File→Open Video 2* seleccionar el fichero car420.yuv con los parámetros: *Video Size* a CIF(362x288) y *Video Format* a 420. Reproducir ambos videos y observar si hay alguna diferencia apreciable.
10. Generar el código para guardar la imagen en formato submuestreo 4:2:2. En este caso al no ser un escalado no lineal, hay que utilizar para el redimensionado de las cromas a formato 4:2:2. Comprobar el resultado con alguno de los anteriores reproductores de video anteriores.
11. En el fichero **videoCarYUV444RGB.m** convierte el video 'car444.yuv' a formato RGB para que pueda ser reproducido por **implay**. Modificar este código para quea capaz de leer los ficheros 'car422.yuv' y 'car420.yuv' y convertirlos a formato RGB para su posterior visualización.

Fichero: videoCarYUV444RGB.m

```
%% Especificaciones del video
NumberOfFrames=20;
% Formato CIF
height=288;
width=352;

%% Proceso de lectura de YcrCb en un fichero
fid = fopen('car444.yuv', 'r'); % Se abre un fichero para leer
% Se lee frame a frame
for numFrame=1:NumberOfFrames
    % Se lee la luma
    YCbCr(:, :, 1, numFrame)=uint8(fread(fid, [ width height ], 'uint8'));
    % Se lee la Cb
    YCbCr(:, :, 2, numFrame)=uint8(fread(fid, [ width height ], 'uint8'));
    % Se lee la Cr
    YCbCr(:, :, 3, numFrame)=uint8(fread(fid, [ width height ], 'uint8'));
end
fclose(fid); % Se cierra el fichero

% Se convierte RGB a YUV frame a frame y se gira
for numFrame=1:NumberOfFrames
    RGB90=ycbcr2rgb(YCbCr(:, :, :, numFrame));
    RGB(:, :, 1, numFrame)=rot90(RGB90(:, :, 1), -1);
    RGB(:, :, 2, numFrame)=rot90(RGB90(:, :, 2), -1);
    RGB(:, :, 3, numFrame)=rot90(RGB90(:, :, 3), -1);
end
imshow(RGB); % Se reproduce el video
```