Tema 11. Digitalización de imágenes y vídeo

Procesamiento de Señales, Sonido e Imágenes Digitales

Carlos Quemada Mayoral

UNIVERSIDAD INTERNACIONAL DE LA RIOJA

La Universidad en Internet

23/02/2021

Índice

11.1. Imágenes digitales

- **Digitalización:** permite representar imágenes **analógicas** reales en **digital** haciendo más sencillo su **almacenamiento**, **procesamiento** y **transmisión**.

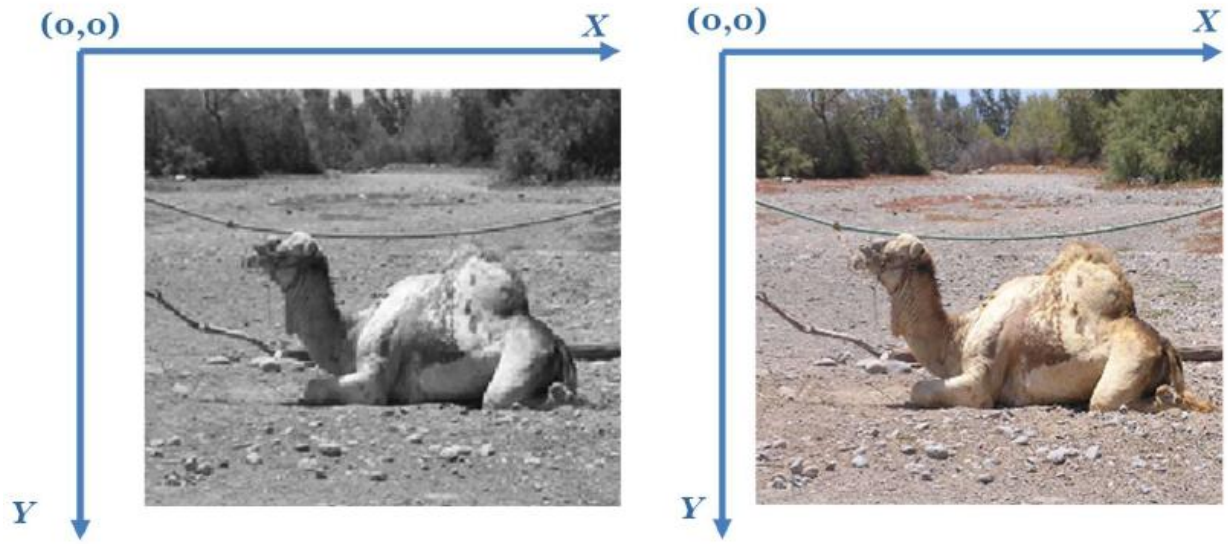
- Dicho proceso engloba otros dos implícitamente: **muestreo** y **cuantificación**, que ya fueron presentados en el tema anterior (Tema 10).

- El resultado de la digitalización es una matriz de píxeles donde el valor de cada píxel corresponde a su **intensidad** (imágenes monocromáticas) o a un **vector de componentes de color** (para imágenes en color).

- Resolución de una imagen se mide en megapíxeles (**MP**): producto del número de filas por el número de columnas que tiene la matriz de píxeles.

11.2. Muestreo

- Una **imagen muestreada** es una matriz bidimensional en la que por convenio se fija el origen en la esquina superior izquierda, tal como muestra la figura:



(a) Imagen muestreada en escala de grises

(b) Imagen muestreada en color

- Los valores muestreados se describen mediante una función bidimensional f(x,y). Para imágenes en escala de grises, el valor de f(x,y) representa la intensidad de luz en ese punto. Para imágenes en color, f(x,y) representa un vector, con las tres componentes de color RGB.

11.2. Muestreo

- **Pixelado cuadrado**. Cuando la densidad horizontal y vertical de píxeles es la misma. En caso contrario se dice que tiene un píxelado **no cuadrado**.

(a)

(b)

- La imagen (a) presenta pixelado cuadrado. Cada pixel es un cuadrado
- La imagen (b) presenta pixelado no cuadrado. Cada pixel es rectangular.

11.2. Muestreo

- **Parámetros de muestreo**.

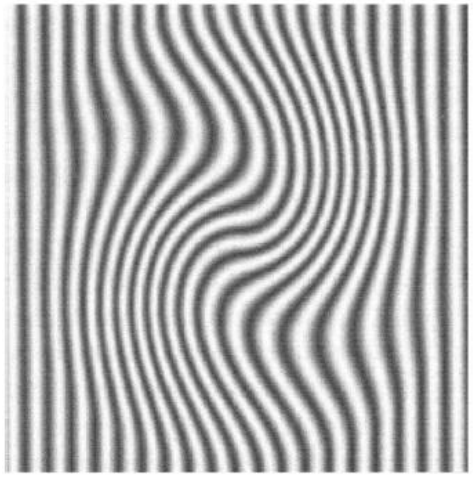
- **Ratio de muestreo o frecuencia de muestreo**: número de muestras que se toman por unidad física de ancho y alto de la imagen.

- **Patrón de muestreo**. Indica la organización física de las muestras. El patrón más utilizado es el **rectangular**, donde las muestras se distribuyen horizontal y verticalmente en filas y columnas, respectivamente. También se utilizan otras organizaciones como la **hexagonal** y la **polarlogarítmica**.

11.2. Muestreo

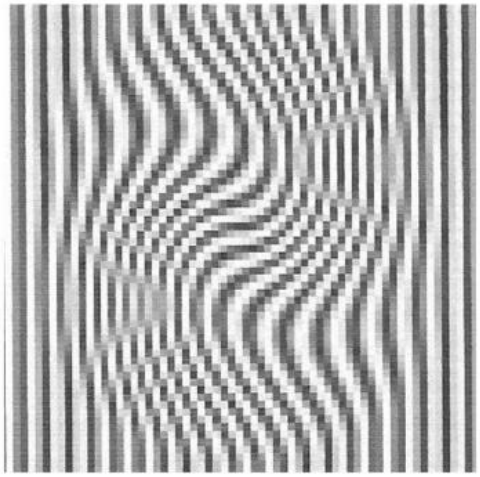
- **Aliasing**. El criterio de Nyquist dice que el **ratio de muestreo** debe ser de al menos dos veces la máxima frecuencia de la señal a muestrear. Si el muestreo se realiza a un ratio inferior, es decir, si hacemos un **inframuestreo** (**undersampling**) se produce **aliasing**.

- En 2D al aliasing se le llama **artefacto de Moiré o patrón de Moiré** y se caracteriza por la aparición de ondas o anillos extraños en la imagen recuperada.



(a)

No aliasing



(b)

Aliasing

11.2. Muestreo

- **Aliasing**. En el caso del vídeo también es posible producir un efecto de aliasing temporal conocido como **efecto de las ruedas del vagón**, en el que los radios de la rueda parecen moverse al revés de la dirección de movimiento del tren.

- Esto se debe a que el número de muestras temporales de la rueda que capta el sistema ojo-cerebro no es suficiente para la velocidad a la que se están moviendo los radios de la rueda.

11.3. Cuantificación

- Consiste en reemplazar los valores de una señal muestreada f(x,y) por un conjunto discreto de niveles de cuantificación.
- A los **niveles de cuantificación n** a veces se les llama niveles de gris porque representan diferentes intensidades en una imagen en escala de grises.
- Por conveniencia **n** suele ser un múltiplo de 2, donde **b** es el número de bits usados para cuantificar cada muestra $\Rightarrow n = 2^b$, n = 0,…, $2^b - 1$.
- Habitualmente b=8, lo cual da lugar a n = 2^8 = 256 niveles de gris.
- En el caso de color, cada nivel de cuantificación tiene asociados tres niveles, uno para cada canal RGB.

11.4. Resolución

- **Nivel de detalle** de una imagen. Dos tipos: resolución **espacial** y resolución de **píxel**.

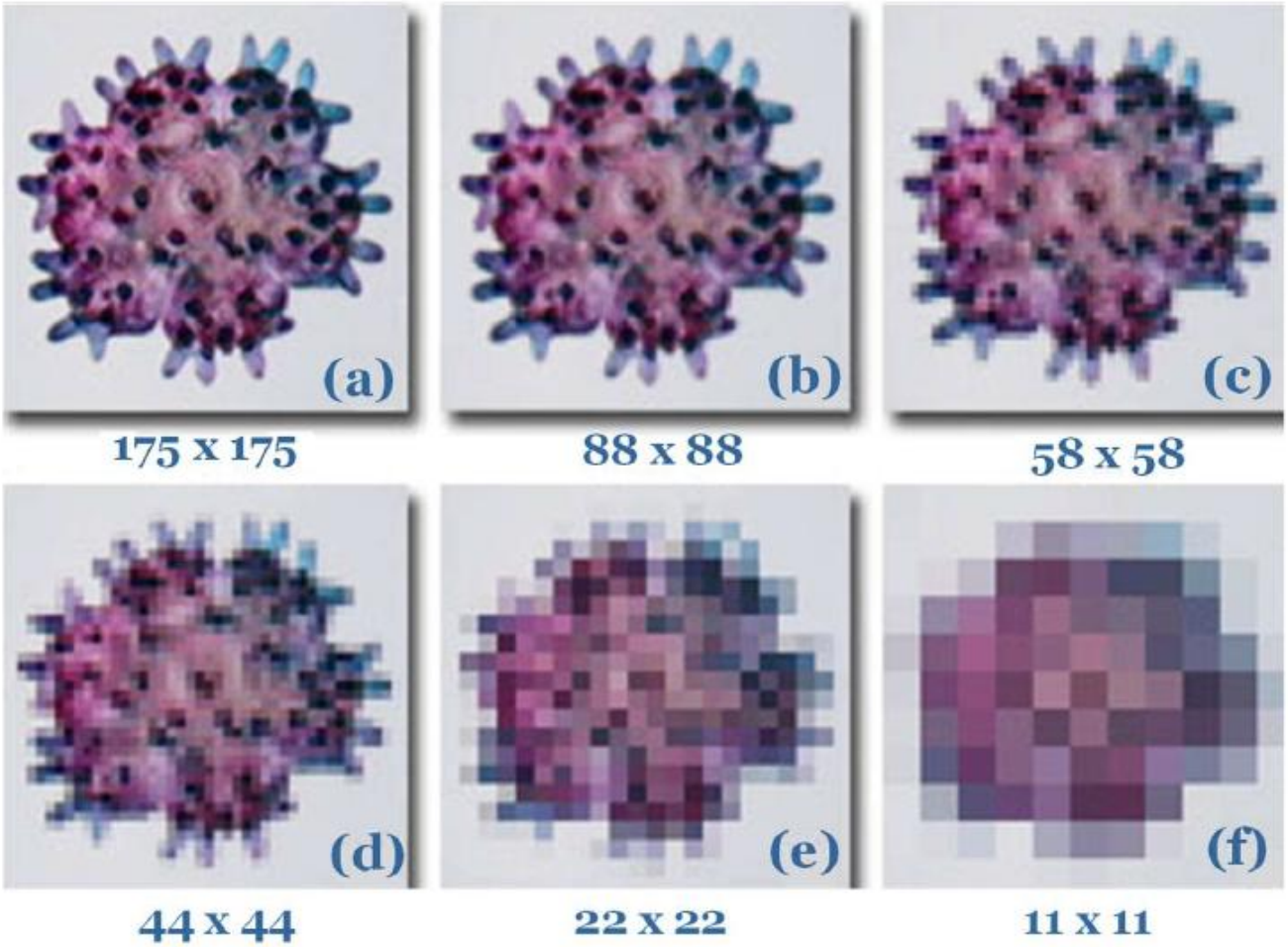
- **Resolución espacial**: Indica el tamaño físico de un píxel, es decir, la densidad de los píxeles de la imagen a representar. Unidad de medida más habitual: **dots per inch** (**DPI**), que indica cuántas muestras se toman en un segmento unidimensional de **1 pulgada** (**2.54 cm**). Se supone pixelado cuadrado.

- DPI debería usarse para imágenes impresas (puntos por pulgada) y para imágenes representadas en un dispositivo digital (computador, cámara, etc) debería usarse la unidad **PPI** (**píxels per inch**). Ambas usan la misma unidad espacial que es la pulgada (**2.54 cm**).

- **Resolución de píxel**: Número de niveles que se usan para cuantificar cada muestra. En el caso monocromático también se llama **resolución de escala de grises**. En las imágenes en color la resolución de píxel se aplicaría a cada uno de los tres colores primarios.

11.4. Resolución

- **Efecto de reducir la resolución espacial**.



11.4. Resolución

- **Efecto de reducir la resolución de escala de grises**.

256 niveles

128 niveles

64 niveles

32 niveles

16 niveles

8 niveles

4 niveles

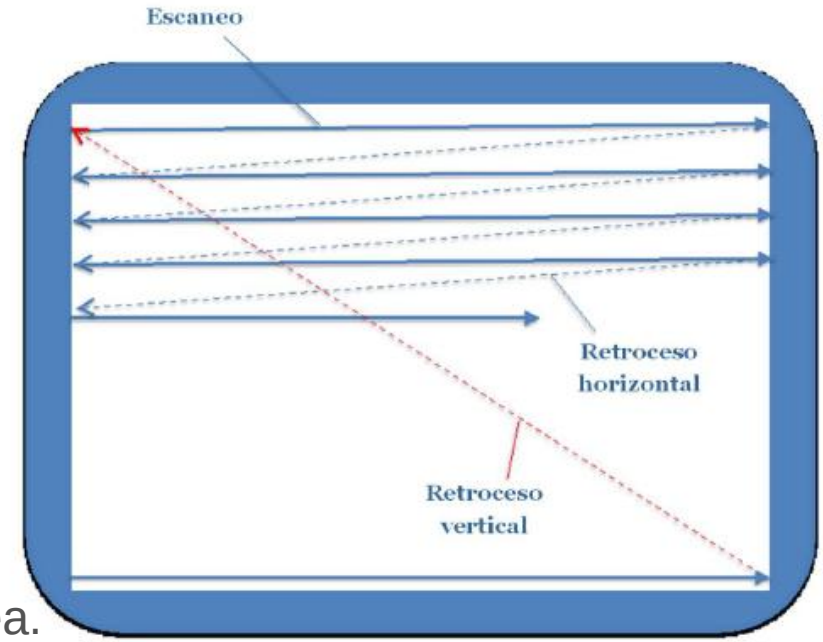
2 niveles

- La calidad de imagen es aceptable cuando tenemos 8 niveles de gris o más, pero a partir de 4 niveles surgen lo que se llaman falsos contornos que hacen que se pierda la información de forma y que en consecuencia la **resolución de píxel** se vuelva inaceptable.

11.5. Vídeo analógico

- Es el que nos encontramos en las **televisiones de tubo de rayos**.
- La señal de vídeo analógico se genera mediante el proceso de escaneo de la información que llega a una cámara.
- **Proceso de escaneo**. Mediante este proceso una señal eléctrica continua f(x,y,t) es generada y guardada para después poder ser enviada al dispositivo de visualización que vayamos a emplear. Esta señal depende del punto (x,y) del cuadro que estemos visualizando y del tiempo.
- Dicha señal se almacena en cintas magnéticas que son materiales fotosensibles que reciben los cambios de intensidad de luz.

11.5. Vídeo analógico



- **Proceso de escaneo**.
- Las líneas **gruesas** corresponden al proceso de escaneo. Empiezan en la esquina **superior izquierda** y la línea de escaneo se mueve horizontalmente y en paralelo. Cuando el escaneo de una línea llega a la derecha, el proceso de escaneo se interrumpe y el lector se coloca en la siguiente línea.
- A este proceso se le llama **retroceso horizontal** (horizontal retrace): líneas discontinuas azules.
- Cuando se escanea la última línea se interrumpe el escaneo y se produce un **retroceso vertical** (vertical retrace): línea roja discontinua. Se llama **frame** al escaneo completo de una pantalla

11.5. Vídeo analógico

- **Proceso de escaneo**.
- La secuencia de escaneo también ocurre cuando **se reproduce el vídeo en un monitor analógico**. La principal diferencia es que en vez de captar la señal f(t) con un sensor, el monitor reproduce la señal f(t) con un cátodo de luz.
- El escaneo de líneas una a una de arriba abajo se llama **escaneo progresivo**.
- Las limitaciones técnicas que existían en los primeros sistemas de televisión analógica hicieron que se popularizada el **escaneo entrelazado**.
- En el **escaneo entrelazado** cada **frame** se escanea dos veces: primero las líneas impares y luego las pares.
- A cada pasada se le denomina **field** y contiene la mitad de líneas que el frame. Los fields se escanean en la mitad de tiempo que los frames y esto permitía reducir el efecto parpadeo que producía reproducir un vídeo con escaneo progresivo. Se reduce al aumentar la frecuencia de escaneo (x2).

11.5. Vídeo analógico

- **Formatos de vídeo analógicos**.

- **Frecuencia de refresco**: número de frames que se muestran en un segundo. Se suele dar en frames/seg, fields/seg o como Hz.

- Criterios usados para clasificar los formatos de vídeo: Número de líneas horizontales escaneadas, frecuencia de refresco, indicación de si es entrelazado (2:1) o progresivo (1:1). Por ejemplo, el formato **NTSC**, usado en EEUU y Japón, presenta **525/60/2:1** $\Rightarrow$ 525 líneas, escaneo entrelazado, frecuencia de 60 fields/seg. Si hubiera sido 1:1, nos hubieran dado la frecuencia en frames/seg.

11.5. Vídeo analógico

- **Representación del color del vídeo analógico**. Modelos RGB y YUV.

- Originariamente el vídeo analógico era en blanco y negro ⇒ se enviaba la **luminancia Y** de la señal en cada punto (x,y) del barrido.

- Modelo **RGB**: el más **tradicional**. Un color se representa mediante la mezcla por adición de los tres colores de luz primarios: rojo (R), verde (V) y azul (B).

- Modelo **YUV**. En este sistema la señal de luminancia **Y** se envía por un canal y las señales de **cromanencia U**=R-Y, **V**=B-Y se envían por otro canal adicional. Estas señales de cromanencia se calculan como:

$$U = R - Y = 0.701R - 0.587G - 0.114B$$
$$V = B - Y = -0.299R - 0.587G + 0.889B$$

11.6. Vídeo digital

- El vídeo analógico representaba una señal continua.
- Por el contrario, en el vídeo digital los sensores CCD (Charge Coupled Device) o CMOS (Complementary Metal Oxide Semiconductor) son los encargados de capturar el vídeo como una secuencia (en cada instante, una matriz) de matrices bidimensionales de píxeles llamadas *frames*.
- Ventajas del vídeo digital frente al analógico:
  - **Robustez**. La señal digital no se degrada por existir defectos o ruido.
  - **Postprocesado**. Es más sencillo modificar y añadir efectos especiales al vídeo en formato digital.

11.6. Vídeo digital

- **Conversión de vídeo analógico a digital**.
- En la sección anterior vimos que el vídeo analógico se representa como una señal 1D f(t).
- Por el contrario, el vídeo digital se representa como una señal 3D f(x,y,t).
- El vídeo analógico es convertido en digital por medio de un conversor analógico-digital (ADC). Este proceso implica tres pasos:
  - **Muestreo**. El ADC escanea a intervalos predefinidos de tiempo la señal analógica f(t) y obtiene el valor de cada píxel (x,y) del *frame* digital f(x,y,t).
  - **Cuantificación**. El ADC asigna a cada muestra un valor discreto dependiendo del número de bits usados para este proceso.
  - **Codificación**. El ADC convierte los *frames* digitales en una representación más concisa mediante algún formato de codificación de vídeo.

11.6. Vídeo digital

- **Parámetros del vídeo digital**.
- **Frame rate** (**fps**): número de *frames* que se muestran por segundo. Δ_t= 1/fps sería el tiempo que permanace un *frame* en la pantalla.
- **Número de líneas** (**h**): número de líneas por *frame*. Dada una pantalla de altura H, H/h sería la altura de un pixel de la pantalla.
- **Número de columnas** (**w**): número de columnas por *frame*. Dada una pantalla de anchura W, W/w sería la anchura de un pixel de la pantalla.
- **Relación de aspecto del píxel** (**PAR**): ratio entre la altura y la anchura de cada píxel. Si un monitor tiene una relación de aspecto AR=H/W, la relación de aspecto de sus píxeles será:

$$PAR = AR\frac{w}{h}$$

- La mayoría de los monitores tienen PAR=1 pero existen estándares de TV como **NTSC** donde el PAR=8/9, o **PAL** donde el PAR=16/15.

11.6. Vídeo digital

- **Parámetros del vídeo digital**.

- **Profundidad de color** (**Nb**): indica el número de bits necesarios para representar un color. Un *frame* con 256 escalas de grises tiene una profundidad de color de Nb=8 bits (2^8=256). Si el *frame* tiene canales RGB, su profundidad de color será Nb=24 bits (8 bits por color).

- **Bitrate** (**bps**): número de bits que se transmiten por unidad de tiempo. Si tenemos el framerate (**fps**), **h** líneas, **w** columnas y profundidad de color **Nb** podemos calcular el bitrate de una secuencia de vídeo como:

$$bps = fps \cdot h \cdot w \cdot N_b$$

11.6. Vídeo digital

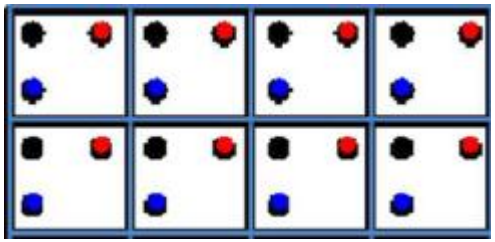
- **Representación del color**.
- En vídeo digital el color se puede representar en **RGB** o en **YUV**.
- La ventaja de usar YUV es que las componentes de color se pueden representar con menor resolución espacial gracias al **submuestreo** del color.
- En vídeo digital, al modelo YUV se le suele llamar YCbCr. Por simplicidad, en esta asignatura emplearemos la terminología YUV.

11.6. Vídeo digital

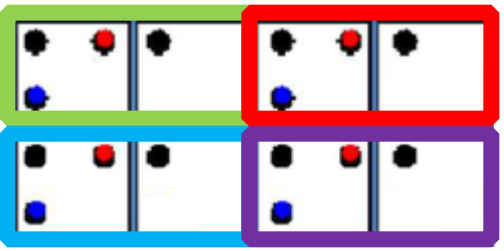
- **Representación del color**.
- Existen diferentes notaciones para representar el submuestreo del color pero la más común es j:a:b aplicado a un bloque de dimensión 2x4 píxeles.
- j hace referencia al número de columnas del bloque o patrón de píxeles seleccionado (4 píxeles habitualmente).
- a hace referencia al número de píxeles de la fila superior que están adquiriendo muestras de color.
- b hace referencia al número de píxeles de la fila inferior que están adquiriendo muestras de color.

11.6. Vídeo digital

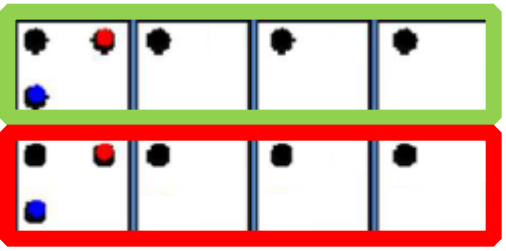
- **Representación del color**. Patrones de submuestreo más típicos:



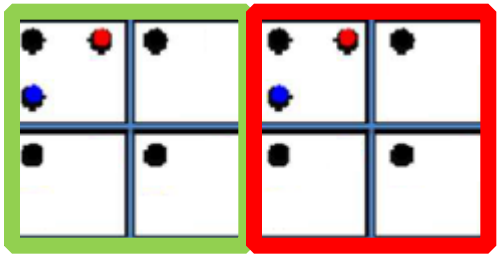
Patrón 4:4:4. No existe submuestreo



Patrón 4:2:2. Solo dos píxeles de la fila superior reciben color y solo dos de la inferior. Puesto que el píxel (1,2) no recibe color, tomará el color del píxel (1,1). El (1,4) tomará el de (1,3), el (2,2) tomará el de (2,1) y el (2,4) tomará (2,3). De esta forma, solo es necesario almacenar la mitad de la información.



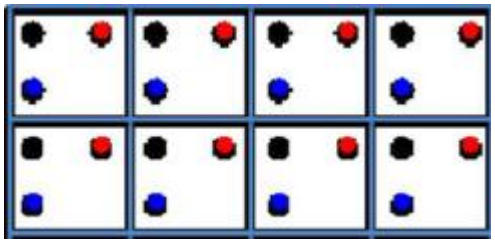
Patrón 4:1:1. Solo un píxel de la fila superior recibe color y solo otro de la inferior. Puesto que los píxeles (1,2), (1,3) y (1,4) no reciben color, tomarán el color del píxel (1,1). Los píxeles (2,2), (2,3) y (2,4) tomarán el color del píxel (2,1). De esta forma, solo es necesario almacenar un cuarto de la información.



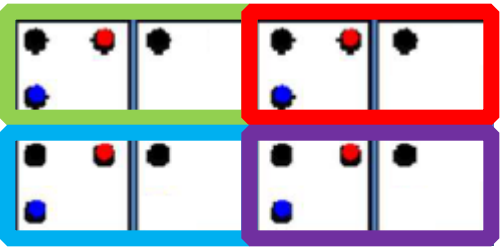
Patrón 4:2:0. Solo dos píxeles de la fila superior reciben color y ninguno de la inferior. Puesto que los píxeles (1,2), (2,1) y (2,2) no reciben color, tomarán el color del píxel (1,1). Los píxeles (1,4), (2,3) y (2,4) tomarán el color del píxel (1,3). De esta forma, solo es necesario almacenar un cuarto de la información.

11.6. Vídeo digital

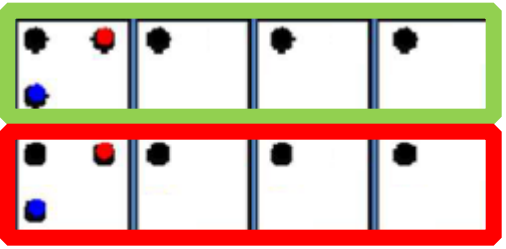
- **Representación del color**. Patrones de submuestreo más típicos:



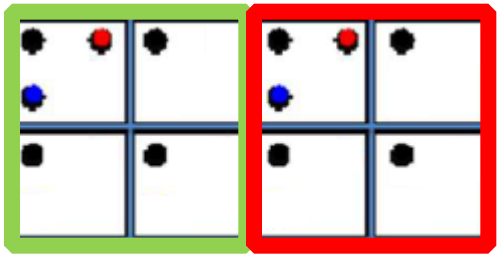
Patrón 4:4:4. Si se emplean 24 bits por cada píxel de color y 8 por cada píxel de luminancia (solo escala de grises), se necesitarían 24 bits por píxel.



Patrón 4:2:2. Cada color necesitaría 24 bits de color y 8 de luminancia (solo escala de grises). Como son dos píxeles por color, necesitaríamos (24+8)/2 = 16 bits por píxel.



Patrón 4:1:1. Cada color necesitaría 24 bits de color y 24 de luminancia (solo escala de grises). Como son cuatro píxeles por color, necesitaríamos (24+24)/4 = 12 bits por píxel.



Patrón 4:2:0. Cada color necesitaría 24 bits de color y 24 de luminancia (solo escala de grises). Como son cuatro píxeles por color, necesitaríamos (24+24)/4 = 12 bits por píxel.

11.6. Vídeo digital

- **Señal de vídeo compuesta y de componentes**.

- **Señal de vídeo compuesta**: señal donde los tres canales de color viajan multiplexados, por lo tanto solo se necesita un medio de transmisión. Por ejemplo, si se envía la señal por cable solo se necesita un cable.

- **Señal de vídeo de componentes**: señal donde cada canal de color viaja por un medio de transmisión distinto. Ventaja: se reducen las interferencias entre canales.

(a) Cable de vídeo compuesto

(b) Cable de vídeo de componentes

11.6. Vídeo digital

- **Televisión digital terrestre (TDT)**. Permite recibir señales digitales aprovechando las antenas de televisión analógicas convencionales junto con un decodificador de TDT que suele ir incluido en la TV.

- **Ventaja principal**: mejor aprovechamiento del espectro RF $\Rightarrow$ mejora de la definición de la imagen y en un aumento del número de canales de TV

- La TV **analógica** solamente permite la transmisión de un único programa de televisión por cada canal UHF. Además los canales adyacentes al que tiene lugar la emisión han de estar libres para evitar las interferencias.

- La **codificación digital** de los programas permite que en el ancho de banda disponible en un solo canal UHF se puedan transmitir varios programas.

- Los estándares de TDT han sido legislados por los gobiernos y existen multitud de ellos dependiendo de la zona geográfica o país.

- En Europa se implantó el estándar DVB-T y luego fue sustituido por el DVB-T2 (permite un 30% más de capacidad de canal debido al uso de formatos de codificación más eficientes).

11.6. Vídeo digital

- **Televisión digital terrestre (TDT)**. Resoluciones de DVB-T2 en función de la relación de aspecto de la pantalla.

i: entrelazado
p: progresivo

Resolución	Tamaño (Píxeles)	Relación de aspecto
480i	720x480	4:3
480p	720x480	4:3
576i	720x576	45:36
576p	720x576	45:36
720p	1280x720	16:9
1080i	1920x1080	16:9
1080p	1920x1080	16:9

- La TDT emite vídeo escalable. En función del tamaño de la pantalla del receptor el vídeo utiliza parte o toda la información recibida por la antena.

11.7. Vídeo en Octave o Matlab

- **Configuración del entorno**.

- Primero, para comprobar si está instalado paquete «video», debemos entrar en Octave y **ejecutar el comando pkg list**:
  - $ octave
  - octave:1> pkg list

- Si no está instalado bájate (**http://octave.sourceforge.net/video/**) el fichero **video1.0.2.tar.gz** y ejecuta el comando: pkg install video-1.0.2.tar.gz

- Después activa el paquete «video» con el comando: **pkg load video.**

11.7. Vídeo en Octave o Matlab

- **Leer ficheros de vídeo**.
- Para consultar información sobre un fichero AVI:
  - aviinfo('tortuga.mov')
  - Incorrect chunk size information in AVI file.
- El comando ha fallado porque el fichero no usa el formato **AVI**.
- Para convertir a formato AVI usar el comando: **ffmpeg**:
- Ahora ya sí podemos consultar la información del fichero: aviinfo('tortuga.avi')

11.7. Vídeo en Octave o Matlab

- La respuesta de la consulta de información es:
  - Filename: '/Users/flopez/tortuga.avi'
  - FileSize: 80506184
  - FileModDate: '07-Dec-2015 09:57:51'
  - NumFrames: 586
  - FramesPerSecond: 120
  - Width: 640
  - Height: 320
  - ImageType: 'truecolor'
  - VideoCompression: 'none'
  - Quality: 4.2950e+07
  - NumColormapEntries: 0

- La lectura de estos campos puede guardarse en una variable (info) y después leer los campos por separado:
  - > info = aviinfo('tortuga.avi');
  - > info.FramesPerSecond
  - ans = 120
- Para cargar el fichero de vídeo en una variable: V = **aviread**('tortuga.avi');

11.7. Vídeo en Octave o Matlab

- **Visualizar los *frames* leídos**.

- En la variable V se habrá creado un vector fila con tantas columnas como *frames* tenga el vídeo:
  - size(V) ⇒ ans = 1 586 %Una fila y 586 columnas.

- Cada elemento del vector es una estructura con dos campos:
  - cdata: contiene la imagen del *frame*.
  - colormap: tabla de color (este campo está vacío si se usa color real RGB)

- Haciendo V(1) ⇒ ans = cdata: [320x640x3 uint8]
  colormap: []

- En V(1), en el campo cdata, hay almacenados 320x640x3 bytes. Hay 320x640 píxeles y cada píxel son 3 bytes según el formato RGB. El campo colormap está vacío.

11.7. Vídeo en Octave o Matlab

- **Visualizar los *frames* leídos**.
- Ahora se puede reproducir el primer *frame* usando imshow sobre cdata:
  - imshow(V(1).cdata)
- En este caso hemos accedido directamente a cdata porque sabemos que los *frames* usan color RGB.
- Si el frame estuviera representado en otro formato, debemos convertir el *frame* a imagen con el siguiente comando: I = frame2im(V(1));
- frame2im solo opera sobre un frame. Si queremos convertir todos los frames del vídeo tenemos que hacer un bucle.
- Análogamente, si quisiéramos convertir una imagen a *frame* debemos hacer el proceso opuesto con el comando im2frame.

11.7. Vídeo en Octave o Matlab

- **Escribir un fichero de vídeo**.

- Una vez procesado el vídeo podemos escribirlo en un fichero con el comando **movie2avi** $\Rightarrow$ movie2avi(V,'tortuga_borrosa.avi');

11.7. Vídeo en Octave o Matlab

- **Leer otros formatos**.

- Aunque Octave no tiene esta opción, MatLab incluye y recomienda usar la operación **mmreader** que es más general y funciona con más tipos de ficheros.

- Esta función crea un **reader** multimedia que luego puede pasarse a la operación **read**:
  - reader = mmreader('tortuga.mov');
  - V = read(reader, [1 10]); Metemos en V 10 *frames*.
  - size(V)
  - ans = 1080 1920 3 10

- La operación **V = read(reader, [1 10])** carga una matriz 4D en V donde cada dimensión contiene: Número de líneas, número de columnas, número de canales RGB y número de *frames*.

- **V(1,1,1,1)** corresponderá a un byte RGB del frame1 y dentro del frame1, de la fila 1 y columna 1.

11.7. Vídeo en Octave o Matlab

- **Leer otros formatos**.
- Para reproducir un determinado *frame*: **imshow(V(:,:,:,1))**;
- Para reproducir todos los *frames*: **implay(V,reader.FrameRate)**;

www.unir.net