



# VISIÓN ARTIFICIAL



## LABORATORIO SESIÓN 3

# LABORATORIO SESIÓN 3.ÍNDICE

---

- 3.1 Sensor RGB Bayer
- 3.2 Repaso espacios de color
- 3.3 Objetivo práctico

# LABORATORIO SESIÓN 3.ÍNDICE

---

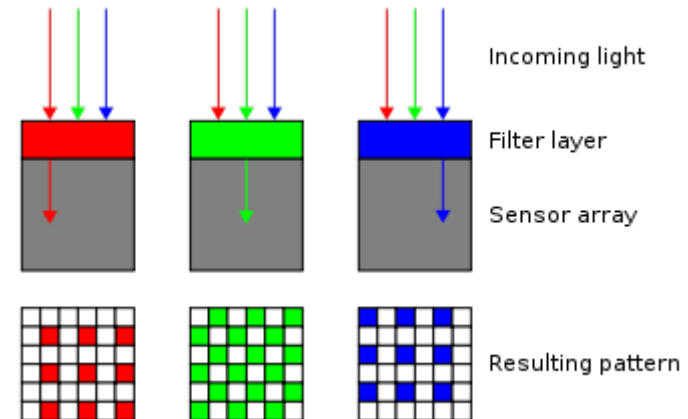
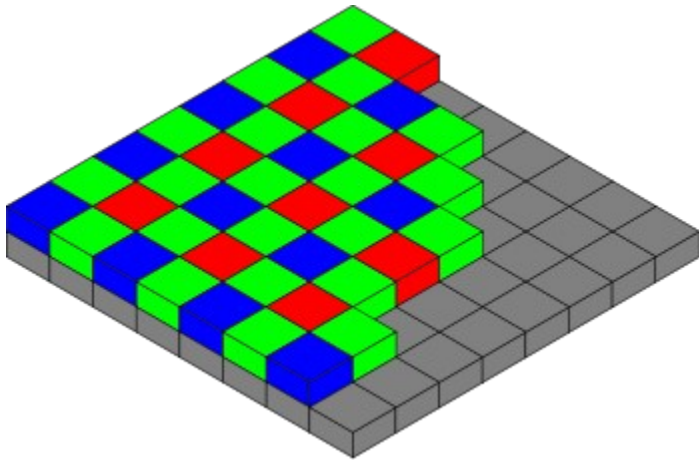
- **3.1 Sensor RGB Bayer**
- 3.2 Repaso espacios de color
- 3.3 Objetivo práctico

# 3.1 Sensor RGB de Bayer

## ■ ¿Cómo se obtiene el color en una cámara?

□ Se suele utilizar el **filtro o mosaico de Bayer**

- **50% verde** (componente principal de luminancia; el ojo humano es más sensible al verde)
- **25% azul** (crominancia)
- **25% rojo** (crominancia)



# 3.1 Sensor RGB de Bayer

- ¿Cómo obtener un valor de Rojo, de Verde y de Azul en cada píxel? Interpolación cromática

- Interpolación de píxeles verdes

$$G8 = (G3 + G7 + G9 + G13) / 4$$

- Interpolación de píxeles rojos/azules en posiciones de verde

$$B7 = (B2 + B12) / 2$$

$$R7 = (R6 + R8) / 2$$

- Interpolación de píxeles rojos/azules en posiciones de azul/rojo

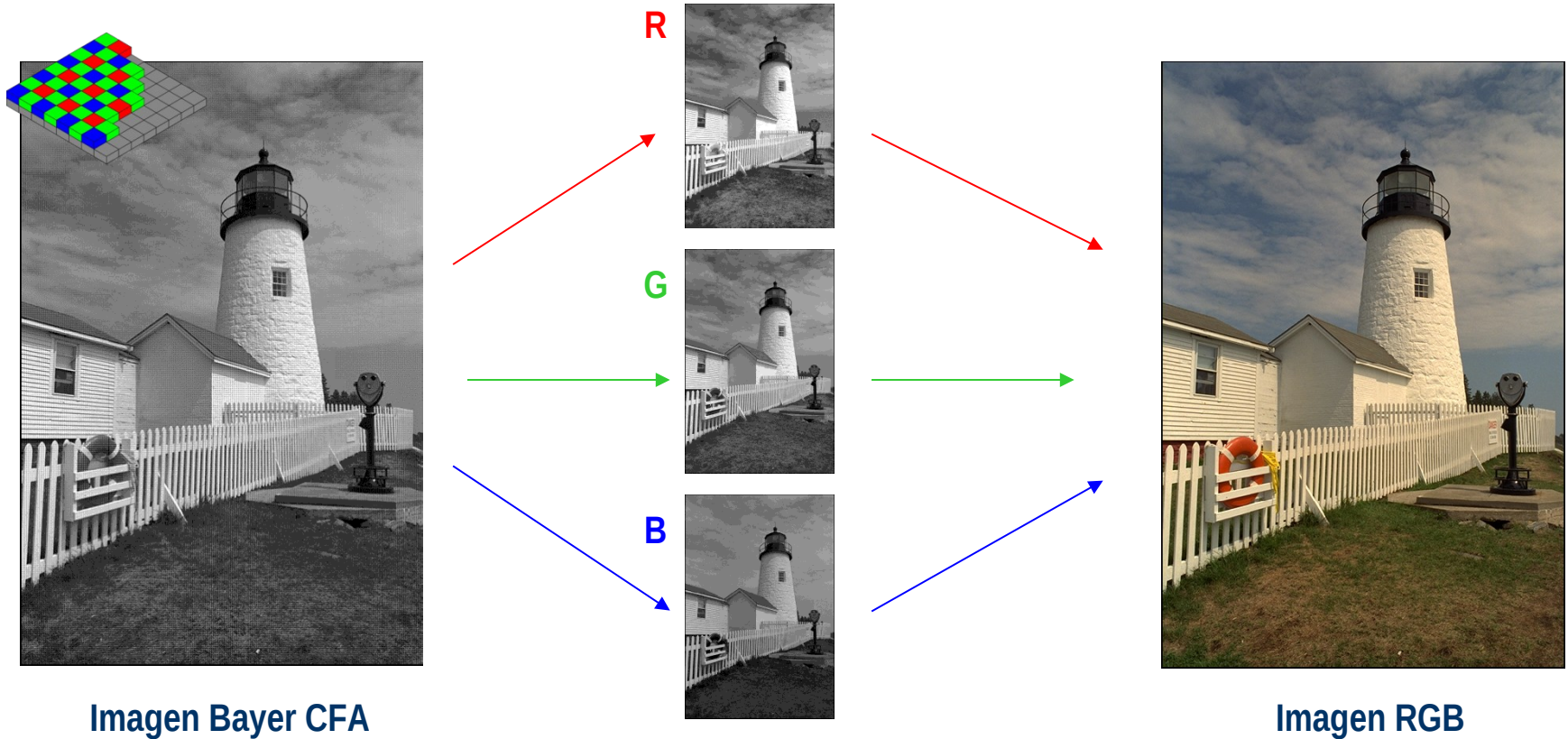
$$B8 = (B2 + B4 + B12 + B14) / 4$$

$$R12 = (R6 + R8 + R16 + R18) / 4$$

G1	B2	G3	B4	G5
R6	G7	R8	G9	R10
G11	B12	G13	B14	G15
R16	G17	R18	G19	R20
G21	B22	G23	B24	G25

# 3.1 Sensor RGB de Bayer

## ■ Conversión Bayer a RGB



# LABORATORIO SESIÓN 3.ÍNDICE

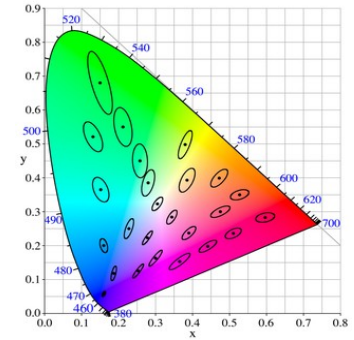
---

- 3.1 Sensor RGB Bayer
- **3.2 Repaso espacios de color**
- 3.3 Objetivo práctico

# 3.2 Repaso espacios de color

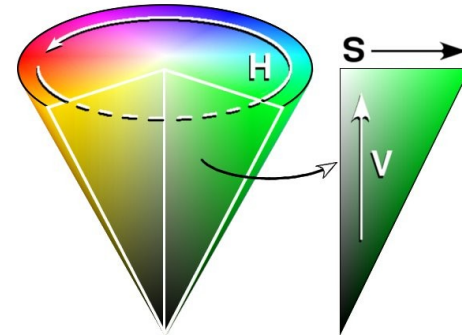
## ■ CIE XYZ y XYZ normalizado:

- Transformación lineal de RGB a XYZ (siendo **Y la luminancia**, esto es, el brillo relativo)
- Perceptivamente no lineal, canales RGB muy correlados, etc.



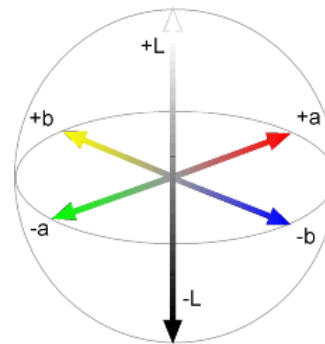
## ■ Hue, Saturation, Value

- Si suponemos distribución normal del espectro
  - Hue (media), Saturation (1/varianza), Value (área)



## ■ Lab

- Mapeado no lineal del espacio XYZ
  - $L \rightarrow$  Luminancia
  - $a$  y  $b \rightarrow$  crominancia perceptivamente lineal



## ■ Yuv/YCrCb

- Luminancia + 2 canales de crominancia perceptivamente no lineales



## 3.2 Repaso espacios de color



RGB



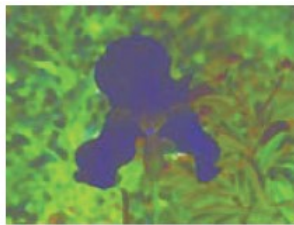
R



G



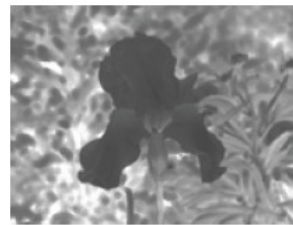
B



rgb



r



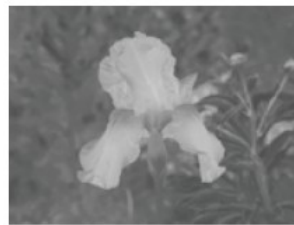
g



b



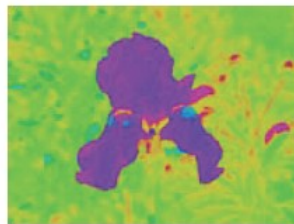
L\*



a\*



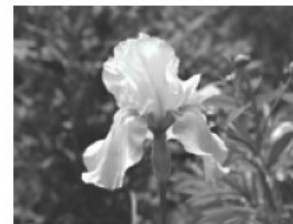
b\*



H



S



V

# LABORATORIO SESIÓN 3.ÍNDICE

---

- 3.1 Sensor RGB Bayer
- 3.2 Repaso espacios de color
- **3.3 Objetivo práctico**

## 3.3 Objetivo práctico

- Partimos del SW de captura y grabación de la Sesión 2
  - Podemos usar `CapturaVideo` y procesar en modo “on-line” (se obtiene una imagen, se procesa, se obtiene la siguiente, se procesa, etc.)
    - **Esta opción solo se puede usar en el laboratorio**
  - Podemos usar la versión modificada en la sesión 2 para grabar imágenes y vídeo
    - Grabamos vídeos en formato RAW (usando `rawImage`)

```
cv::Mat imgRead = cv::Mat(rawImage.GetRows(), rawImage.GetCols(), CV_8UC1, rawImage.GetData());
```
    - Usamos el proyecto `Lee.zip` para leer del vídeo
    - **Esta opción permite grabar vídeos en el laboratorio y usarlos en cualquier PC que tenga instaladas las OpenCV**
- Por si acaso, se proporciona un video de ejemplo:  
`dat/video00000.raw`

## 3.3 Objetivo práctico

- Se proporciona el archivo **main.cpp**
  - Lee un vídeo en formato RAW
    - Parámetros de entrada: nombreVideo.raw ancho alto
  - Entra en un bucle esperando a que se pulse una tecla
  - En función de la tecla se debe transformar la imagen Bayer capturada por la cámara
    - **Llamar a la función correspondiente**
    - **Guardar resultado con `cv::imwrite`**

## 3.3 Objetivo práctico

- Crear una librería que se llame **colors.cpp** y **colors.h** que contenga las siguientes funciones:

- ☐ **OBLIGATORIAS**

- ☐ `int cvt_Bayer_RGB (cv::Mat imgBayer, cv::Mat *imgRGB);`
- ☐ `int cvt_Bayer_HSV (cv::Mat imgBayer, cv::Mat *imgHSV);`

- ☐ **OPCIONALES**

- ☐ `int cvt_Bayer_XYZ (cv::Mat imgBayer, cv::Mat *imgXYZ);`
- ☐ `int cvt_Bayer_Lab (cv::Mat imgBayer, cv::Mat *imgLAB);`
- ☐ `int cvt_Bayer_Luv (cv::Mat imgBayer, cv::Mat *imgLUV);`
- ☐ `int cvt_Bayer_YCrCb (cv::Mat imgBayer, cv::Mat *imgYCrCb);`

## 3.3 Objetivo práctico

- Cada función deberá llevar a cabo las siguientes tareas:
  - Transformar del espacio Bayer al espacio RGB
    - `cv::cvtColor()`
  - Transformar del espacio RGB al espacio correspondiente (HSV, XYZ, Lab, Luv, YCrCb) en cada función
    - `cv::cvtColor()`
  - Separar en 3 canales la función transformada en 3 imágenes diferentes
    - `vector<cv::Mat> canales(3);`
    - `cv::split(imgColor, canales);`
    - `cv::imshow("B", canales.at(0));`
    - `cv::imshow("G", canales.at(1));`
    - `cv::imshow("R", canales.at(2));`
  - Visualizar cada uno de los canales, así como la imagen transformada

## 3.3 Objetivo práctico

- Capturar y analizar los resultados de los distintos espacios de color, así como de cada uno de sus canales



Bayer



RGB



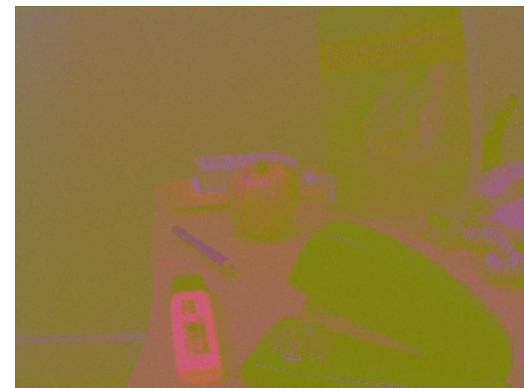
XYZ



HSV



Lab



YCrCb