

# Visión por Computadora I

Ing. Maxim Dorogov  
(mdorogov@fi.uba.ar)

Laboratorio de Sistemas Embebidos -FIUBA

# PROGRAMA SUGERIDO

- **Clase 1:** Introducción a imágenes, sistemas de visión y OpenCV
- **Clase 2:** Op. de píxel, histogramas, binarización, coord. cromáticas
- **Clase 3:** Filtros: Lineales, separables, padding, DoG, Fourier, Bordes (Canny)
- **Clase 4:** Detección de objetos. Algoritmo de Harris. Shi-Tomasi. Hough. Pirámides.
- **Clase 5:** Extracción de características. SIFT, SURF, ORB, FAST, HoG, LBP
- **Clase 6:** Segmentación: k-means, watersheed, mean-shift. Procesamiento morfológico.
- **Clase 7:** Procesamiento de video, gstreamer, ffmpeg, optical flow, sustracción de fondo tracking.
- **Clase 8:** Examen + Teórica opcional.



# RÉGIMEN DE APROBACIÓN

- Trabajos prácticos correspondientes a cada unidad temática.
- Examen teórico.

Dinámica esperada para las clases:

- 90 minutos de teoría
- 10 minutos de descanso
- 80 minutos de práctica




# HERRAMIENTAS PARA LA CURSADA

- Lenguaje de programación
  - Python > 3.6
- Bibliotecas de código
  - Numpy
  - OpenCV/OpenCV-contrib > 3.4
  - Matplotlib
  - Entorno de programación:
    - Jupyter notebook/Google Colab
  - Gestión de entornos: Conda/Miniconda

<https://www.anaconda.com/distribution/>



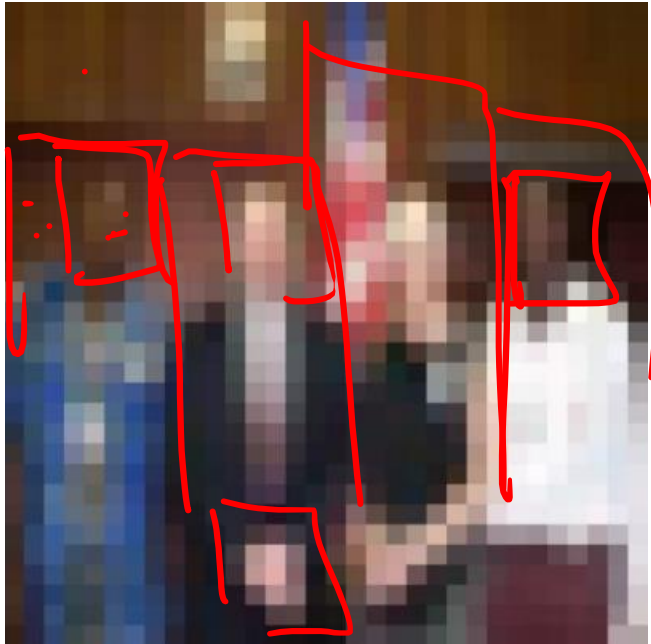
# BIBLIOGRAFÍA SUGERIDA

- La bibliografía es de referencia y no será obligatorio el uso de la misma.
-  ▪ Computer Vision: Algorithms and Applications | Szeliski | Springer
  - <http://szeliski.org/Book>
- Computer Vision: A Modern Approach | Forsyth, Ponce | Pearson
- Computer Vision | Shapiro | Pearson
- Learning OpenCV | Bradski, Kaehler | O'Reilly



# DESAFÍOS

- ¿Por qué puede la visión por computadora resultar compleja?
- Hay tareas complejas que nuestros ojos y cerebro realizan con facilidad pero **no son triviales** para sistemas de visión artificial.



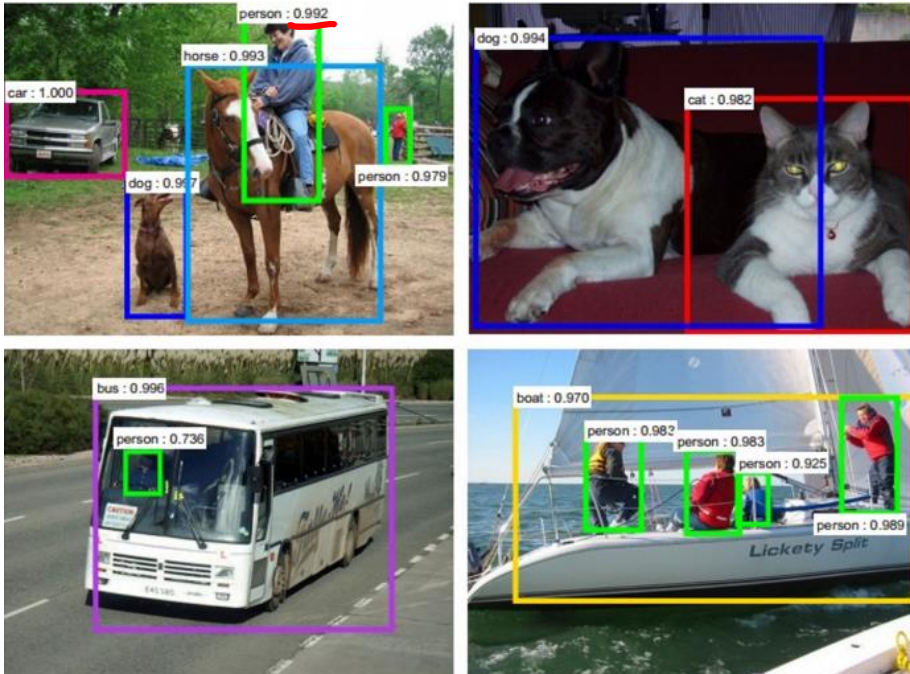
Podemos percibir y entender estas dos imágenes sin mayores inconvenientes...



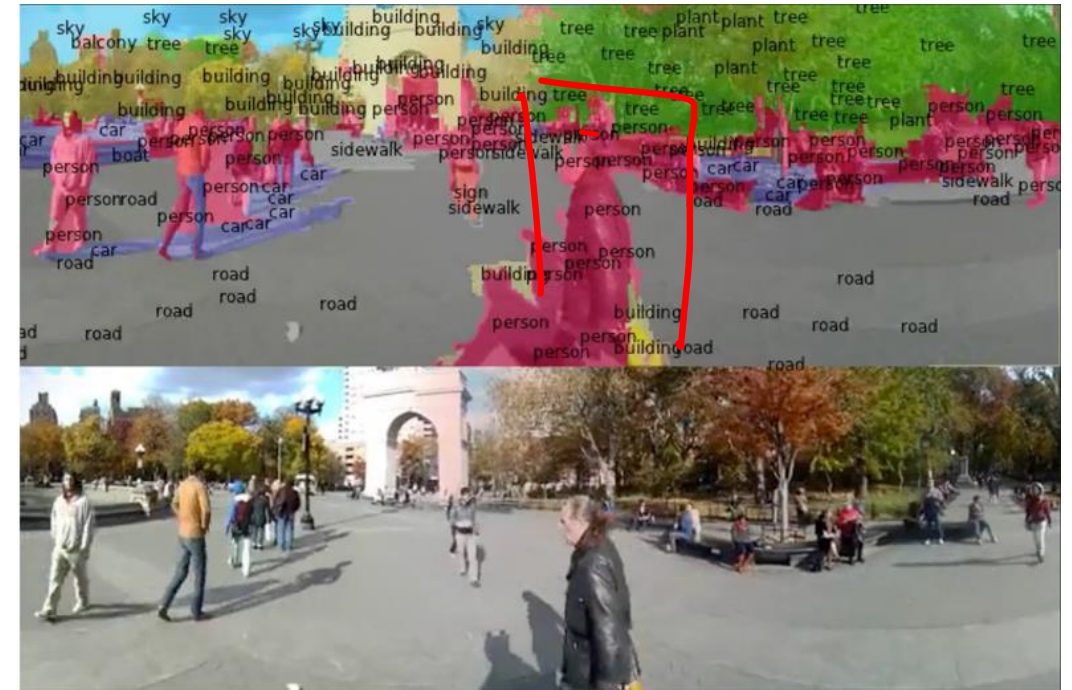


# DESAFÍOS

- ¿Qué podemos resolver con visión por computadora?
- Ejemplos:



Reconocimiento y detección

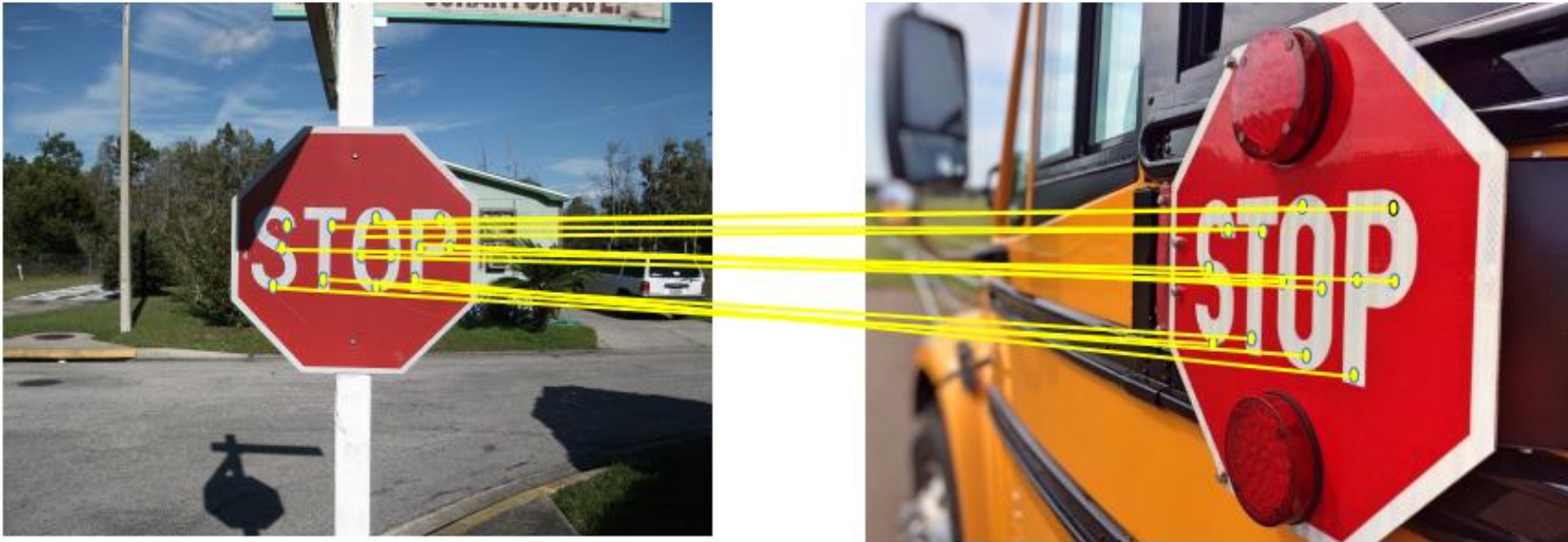


Segmentación de objetos



# DESAFÍOS

- Mas ejemplos:



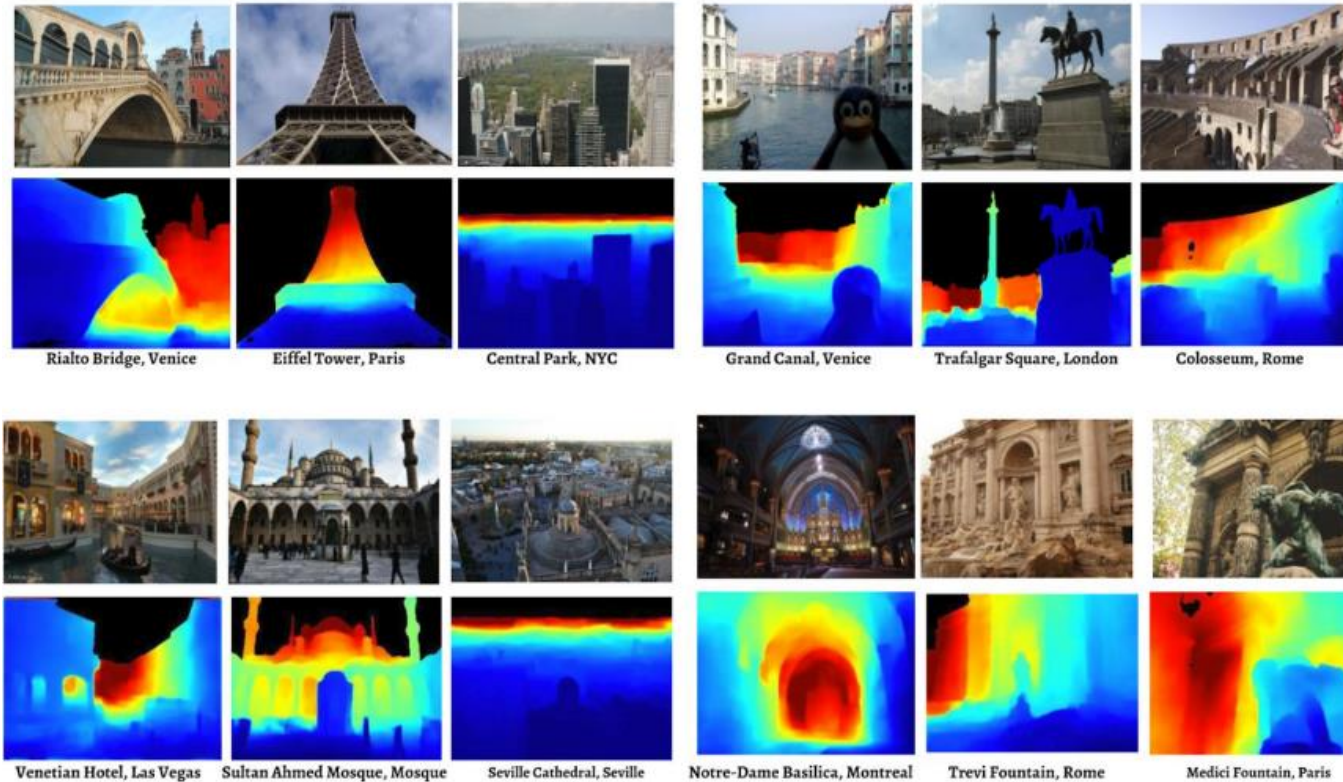
Extracción y detección de características o keypoints. (SIFT, David Lowe, 1999 )





# DESAFÍOS

- Mas ejemplos:



Estimación de profundidad



# DESAFÍOS

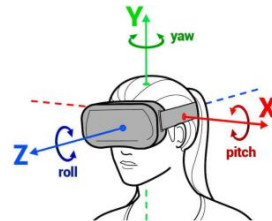
- Hoy en día es uno de los campos interdisciplinarios mas estudiados y con mayores aplicaciones en la vida cotidiana.



NASA's Mars Curiosity Rover



Amazon Scout



6DoF head tracking



Hand & body tracking



3D scene understanding



3D-360 video capture



Vehículos asistidos



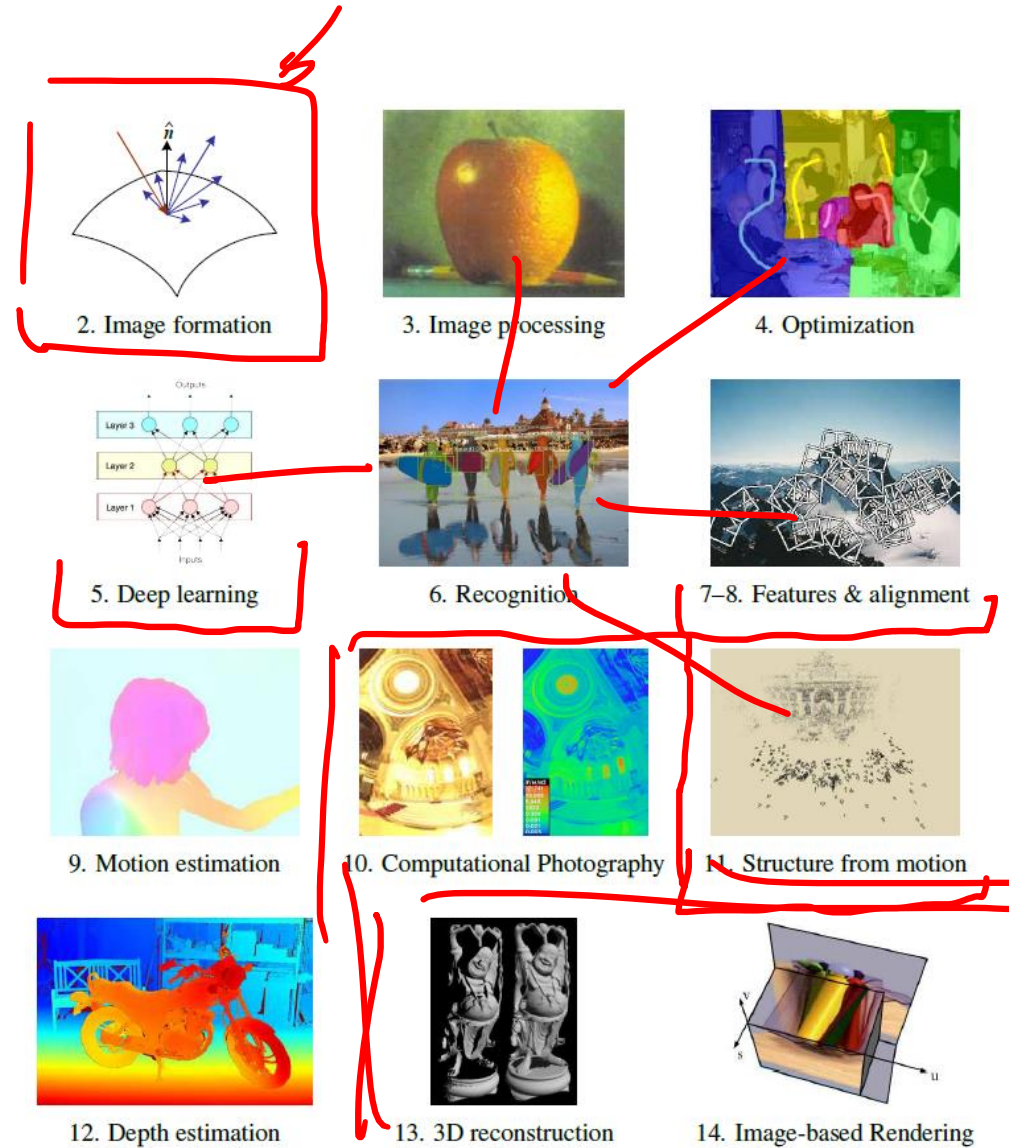
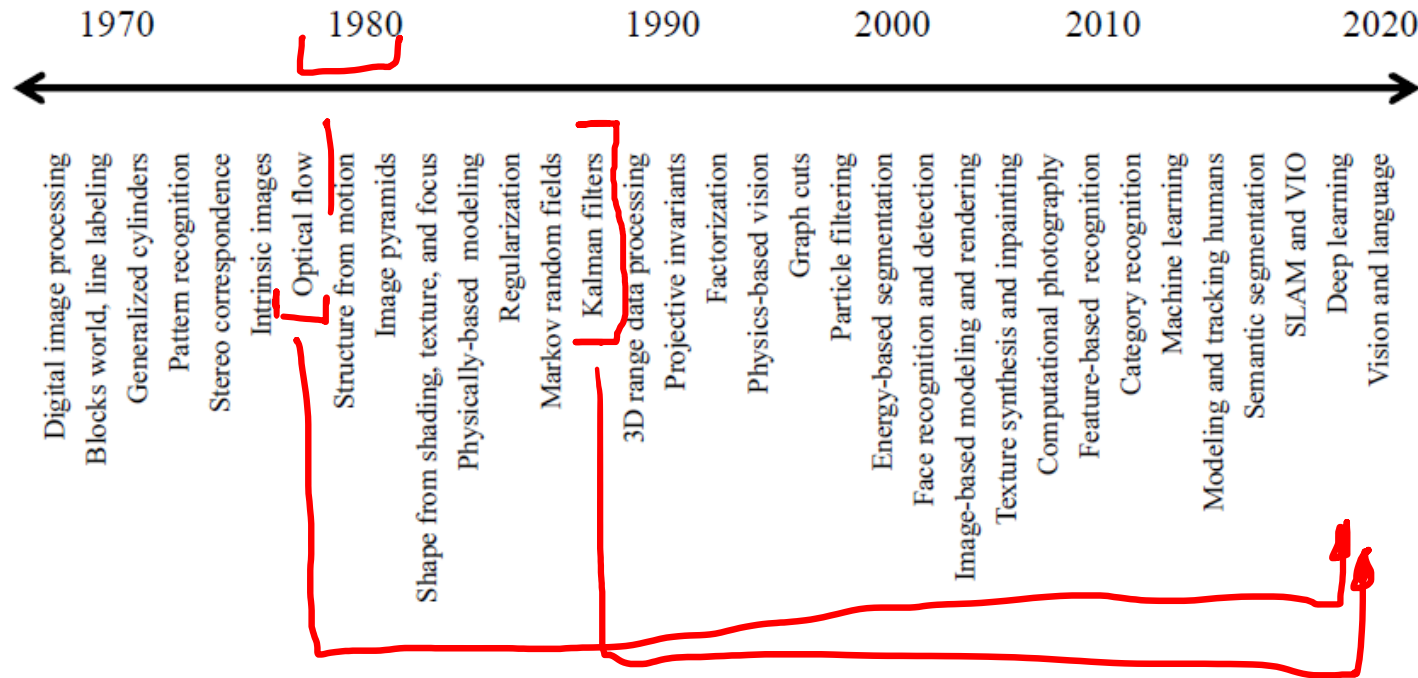
# DESAFÍOS

- Desde chicos ya estábamos familiarizados con sistemas de visión artificial:





# HISTORIA Y APLICACIONES



**Visión por computadora: Historia y campos de aplicación**



# PERCEPCIÓN DE LA LUZ POR EL OJO

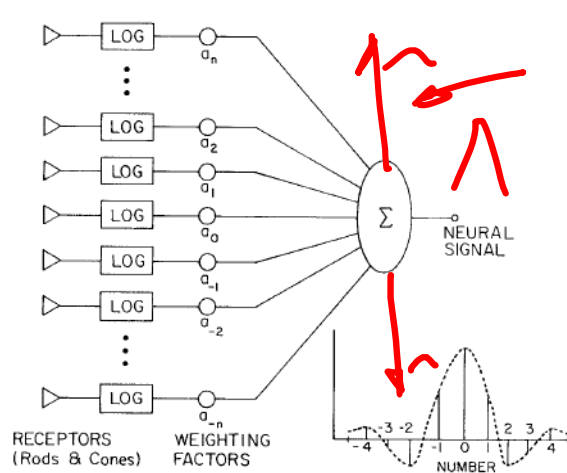
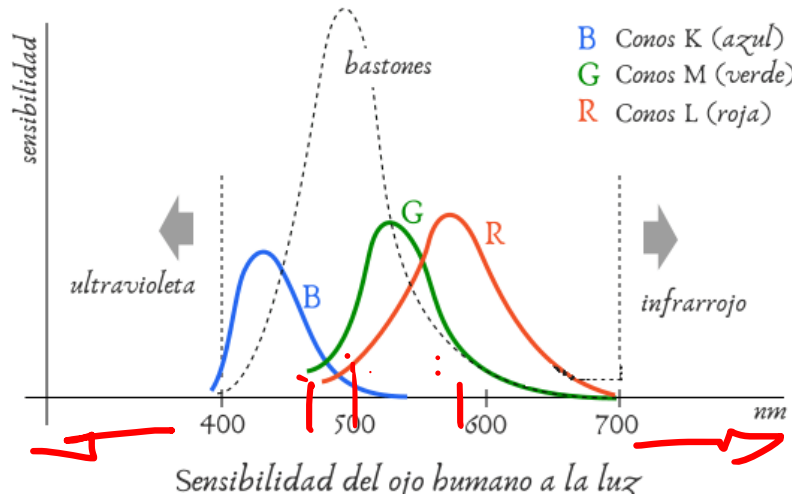
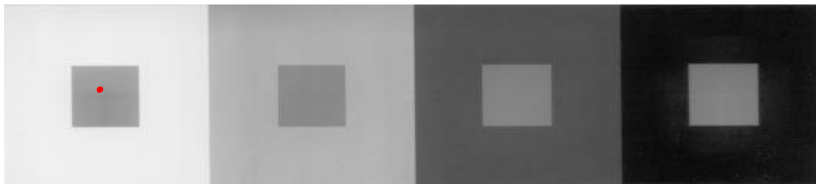


FIGURE 2.4-6. Lateral inhibition effect.



- Bastones: Intensidad (responden poco al rojo). 120 millones
- Conos: Color (concentrados en la mácula). 6~7 millones
  - Rojos – 64%
  - Verdes – 32 %
  - Azules – 2 %
- Respuesta logarítmica. Fracción de Weber (0.02)
- Daltonismo (8% hombres / 1% mujeres)





# MODELOS DE COLOR

- RGB: Commission Internationale d'Eclairage (CIE) en 1930

- Rojo: 700 nm
- Verde: 546,1 nm
- Azul: 435,8 nm

- XYZ: Resuelve el color negativo

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{0,17697} \begin{bmatrix} 0,49 & 0,31 & 0,20 \\ 0,17697 & 0,81240 & 0,01063 \\ 0,00 & 0,01 & 0,99 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Además, permite separar crominancia de luminancia

$$x = \frac{X}{X+Y+Z} \quad y = \frac{Y}{X+Y+Z} \quad z = \frac{Z}{X+Y+Z}$$

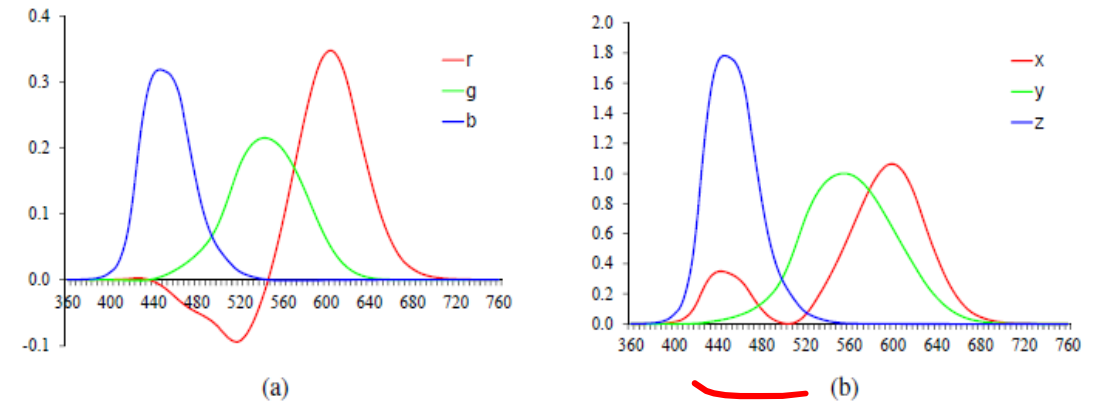
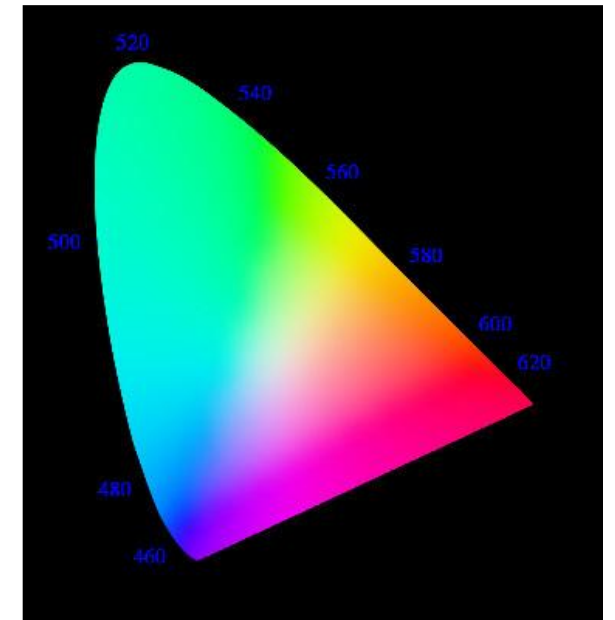
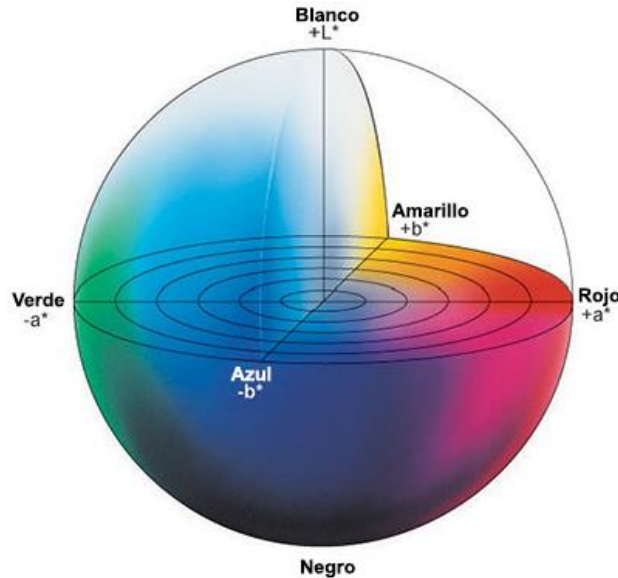


Figure 2.28 Standard CIE color matching functions: (a)  $\bar{r}(\lambda)$ ,  $\bar{g}(\lambda)$ ,  $\bar{b}(\lambda)$  color spectra obtained from matching pure colors to the R=700.0nm, G=546.1nm, and B=435.8nm primaries; (b)  $\bar{x}(\lambda)$ ,  $\bar{y}(\lambda)$ ,  $\bar{z}(\lambda)$  color matching functions, which are linear combinations of the  $(\bar{r}(\lambda), \bar{g}(\lambda), \bar{b}(\lambda))$  spectra.



# OTROS ESPACIOS DE COLOR

## ▪ CIELAB (L\*a\*b)



$$L^* = 116 f\left(\frac{Y}{Y_n}\right); \quad a^* = 500 \left[ f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right]; \quad b^* = 200 \left[ f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right]$$

→ Y V V

$$f(t) = \begin{cases} t^{1/3} & \text{si } t > \delta^3 \\ \frac{t}{3\delta^2} + \frac{2\delta}{3} & \text{otro caso} \end{cases}$$

0 - 1.2  
0 - π

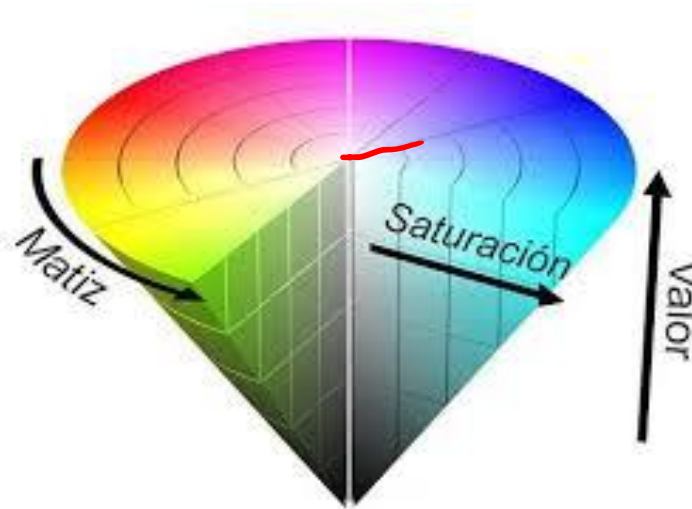
## ▪ HSV

- **Hue:** Dirección alrededor de la rueda de color, en grados  $\in [0, 360]$
- **Saturation:** Distancia escalada desde la diagonal  $\in [0, 1]$
- **Value:** Promedio o máximo valor de color  $\in [0, 1]$

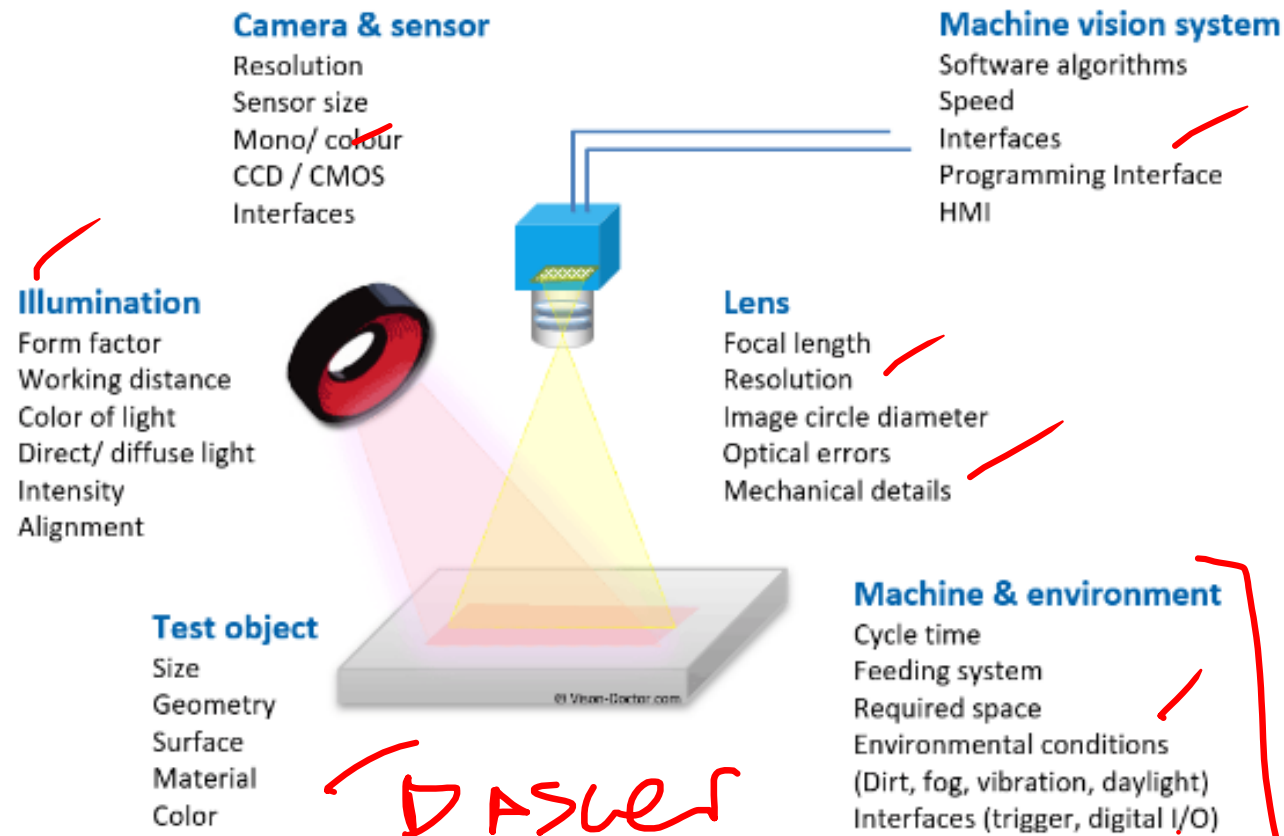
Conversión RGB → HSV

- $V = M = \max(R, G, B); m = \min(R, G, B)$
- $S = (M - m)/M$  ( $S = 0$ , si  $V = 0$ )
- $H = 60 \times \begin{cases} 0 & , \text{si } (M - m) = 0 \\ 0 + (G - B)/(M - m), & \text{si } \max = R \\ 2 + (B - R)/(M - m), & \text{si } \max = G \\ 4 + (R - G)/(M - m), & \text{si } \max = B \end{cases}$

$$H = H + 360, \text{ si } H < 0$$



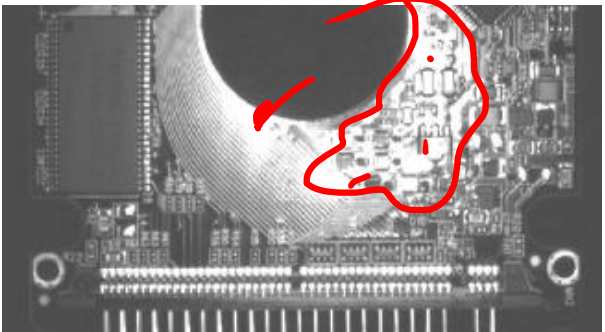
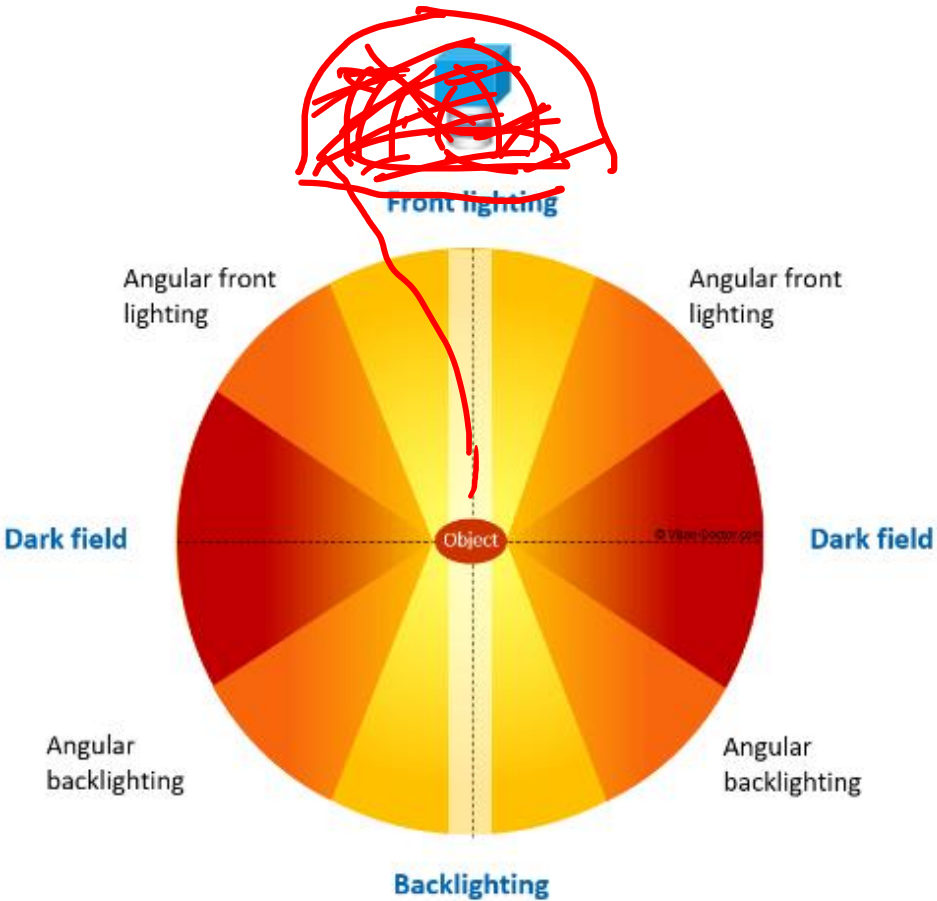
# SISTEMAS DE VISIÓN



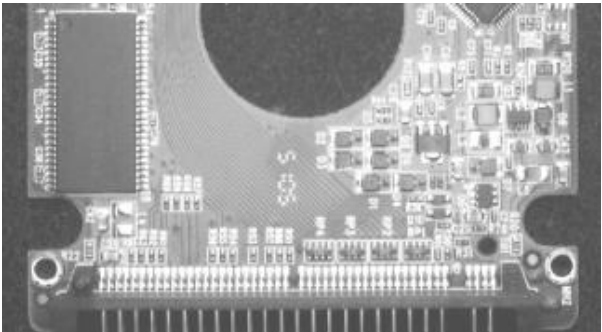
- **Cámara:**
  - Sensores CMOS o CCD.
  - Tipo de shutter
  - Tipo de interfaz (USB, Ethernet, etc...)
  - LUTs
  - Depth Camera
  - Smart
  - Etc...
- **Lente:**
  - Telecéntrica
  - Entocéntrica
- **Iluminación:**
  - Campo oscuro (Dark field)
  - Luz directa
  - Luz difusa (Domo)
  - Backlight
  - Color
- **Procesamiento**
  - Cloud
  - Edge



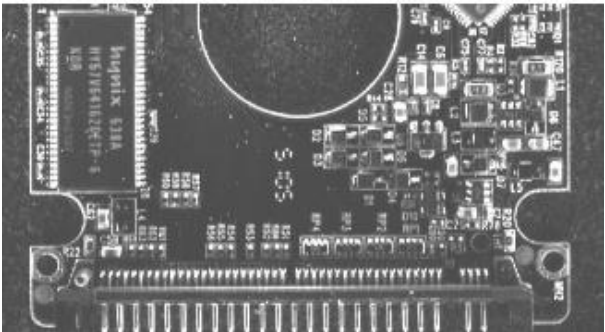
# ILUMINACIÓN



Luz directa



Luz con difusor



Dark Field ✓

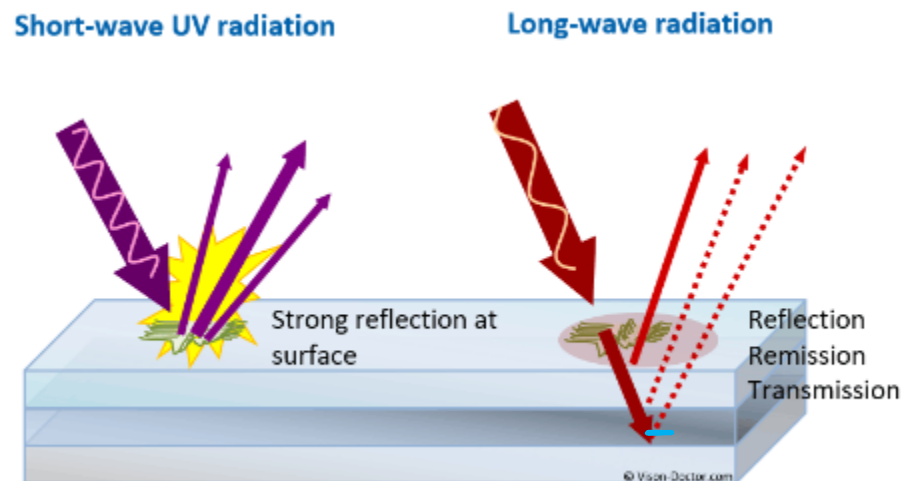
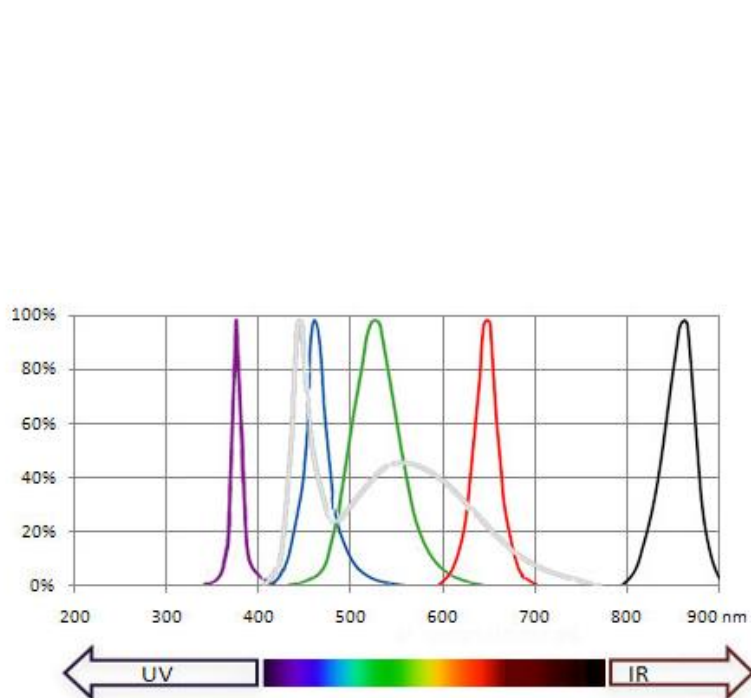


Backlight



# ILUMINACIÓN

Iluminación por encima (y debajo) del espectro visible:



Luz ultravioleta: Permite ver capas internas del material a analizar



Algo similar sucede cuando se utiliza luz infrarroja





# LENTE

Principalmente se dividen en etnocéntrica y telecéntrica, las primeras producen un error de perspectiva y son las lentes mas comunes usadas en fotografía, cine, y aplicaciones de consumo masivo.

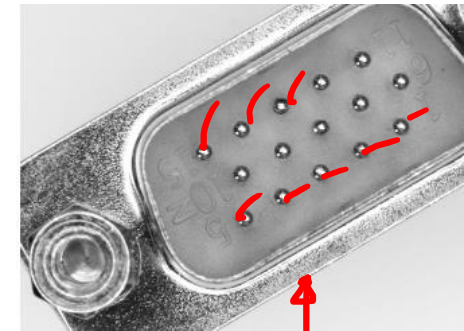


Lens class	Aperture angle
Tele lens	$< 20^\circ$
Long focal length lens	$20-40^\circ$
Normal lens	$40-55^\circ$
Wide angle lens	$> 55^\circ$
Super wide angle lens	$\sim 110^\circ$
Fisheye lens	$\sim 180^\circ$

Tipos de lente etnocéntrica



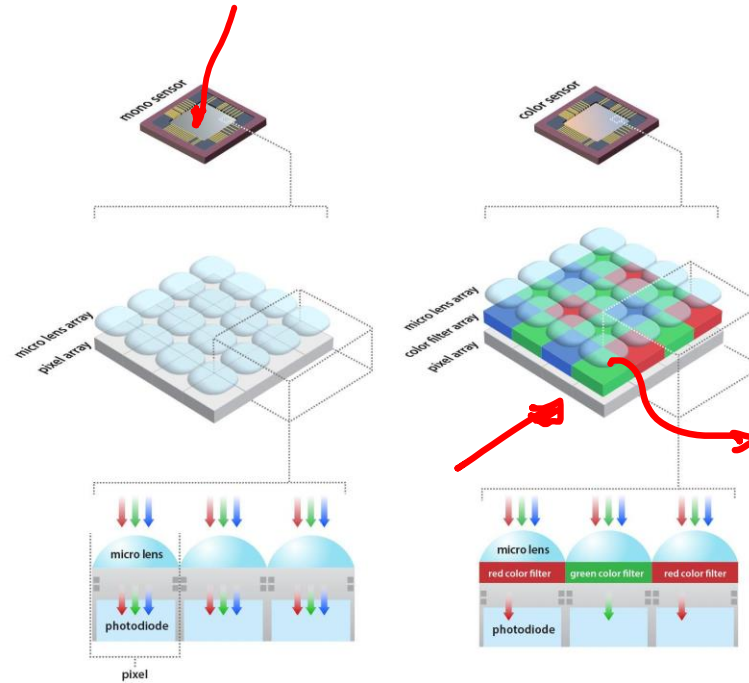
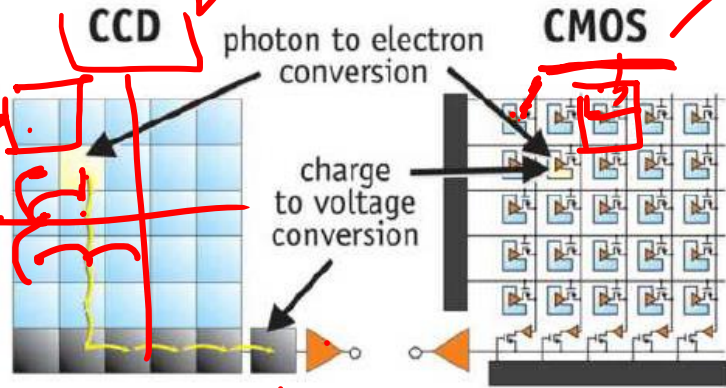
La lente telecéntrica corrige los errores de perspectiva ya que los rayos de luz inciden paralelos al eje óptico de la cámara. Son usadas principalmente en metrología y no tienen aplicaciones por fuera del ambiente industrial.



Lente etnocentrica, (izquierda) y telecéntrica (derecha)



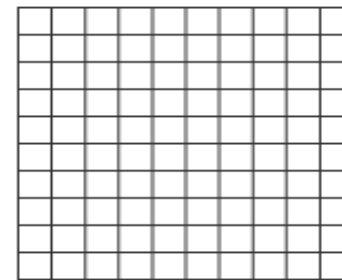
# TIPOS DE SENSORES



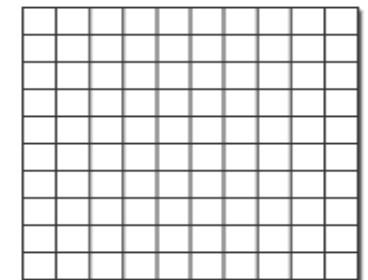
- **CCD** (charge-couple device):  
Willard Boyle y George E. Smith  
– Laboratorios Bell – 1969
  - A/D central 20 a 75MHz
  - Blooming/ Smearing ✓
  - Mayor sensibilidad
- **CMOS** (complementary metal oxide semiconductor)
  - Conversión en el fotosito
  - Rolling shutter
  - Microlentes



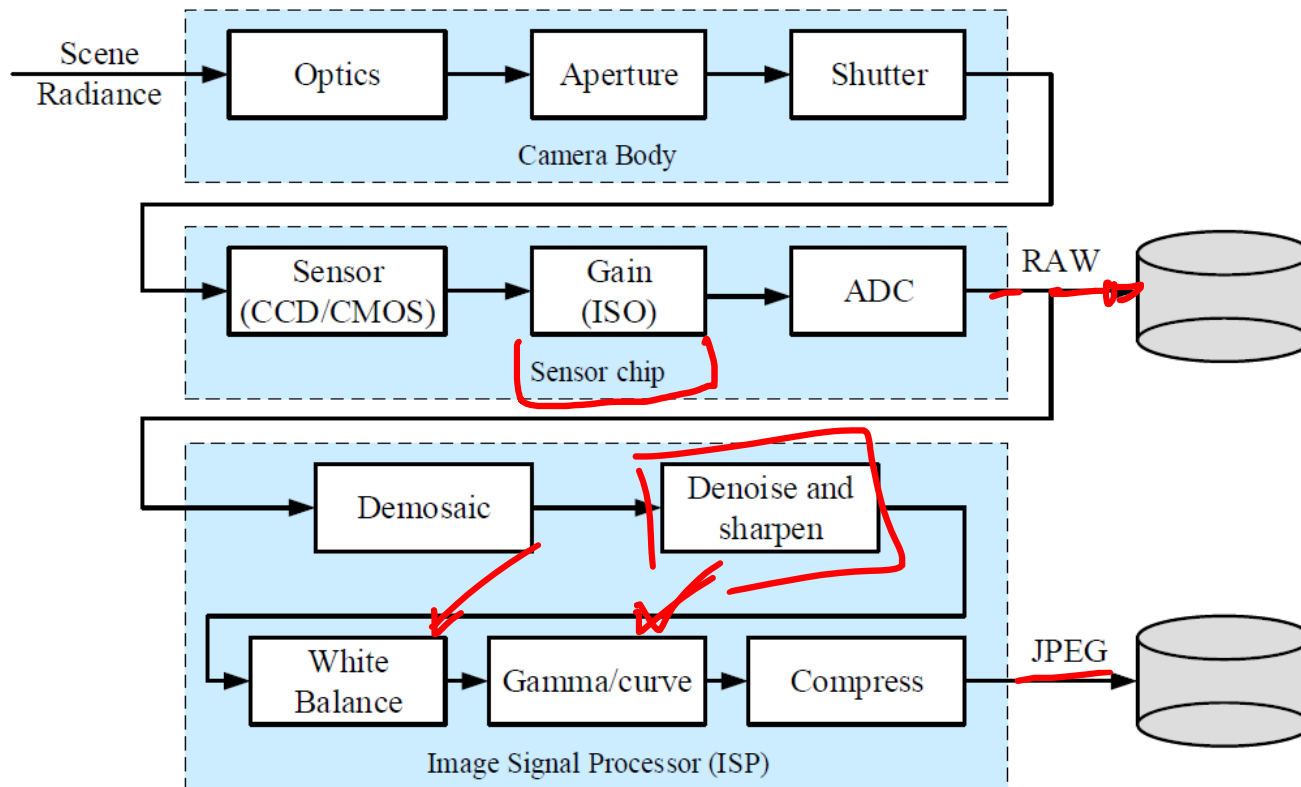
Rolling Shutter



Total Shutter ✓



# CÁMARA DIGITAL



Las cámaras profesionales y de aplicaciones específicas permiten, además de una imagen, obtener los datos en formato RAW. (Raster en aplicaciones espaciales)

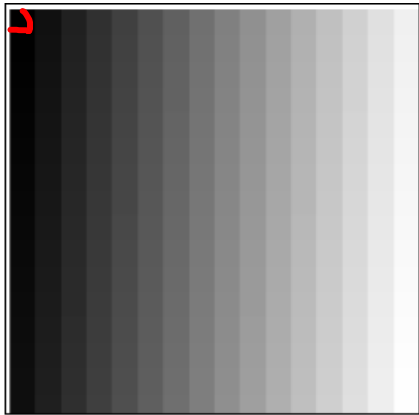
- No es un “formato” estandarizado
- Algunos formatos de datos RAW: TIFF, CR2, 3FR, NRW.
- Ocupan 3 veces (o mas!) de espacio que un archivo de imagen.

Muchas veces en la cámara se efectúa un post procesamiento digital para compensar las falencias del sistema óptico (lente-sensor), algo típico en teléfonos celulares o cámaras hogareñas.

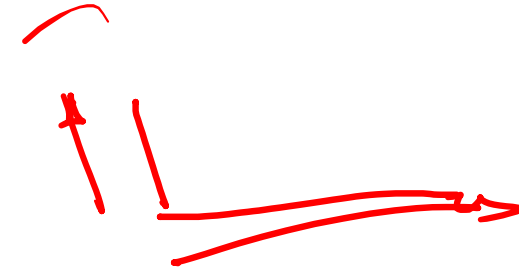


# REPRESENTACIÓN DE IMÁGENES

- $N \times M \times 1$  si la imagen esta en escala de grises
- $N \times M \times 3$  para imágenes a color
- Rango: 0 (pixel apagado) – 255 (max. Intensidad) con uint8 para indicar la intensidad de cada pixel
- También existen imágenes binarias, se utilizan como mascararas
- En OpenCV y Numpy se indexa por [N-fil, N-col, N-canal]

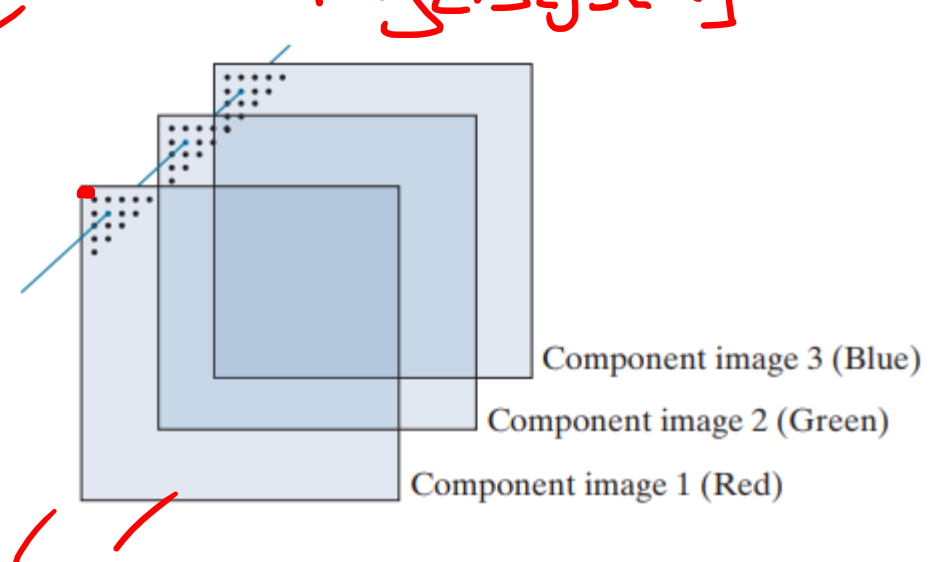


0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241
2	18	34	50	66	82	98	114	130	146	162	178	194	210	226	242
3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243
4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244
5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245
6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246
7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247
8	24	40	56	72	88	104	120	136	152	168	184	200	216	232	248
9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249
10	26	42	58	74	90	106	122	138	154	170	186	202	218	234	250
11	27	43	59	75	91	107	123	139	155	171	187	203	219	235	251
12	28	44	60	76	92	108	124	140	156	172	188	204	220	236	252
13	29	45	61	77	93	109	125	141	157	173	189	205	221	237	253
14	30	46	62	78	94	110	126	142	158	174	190	206	222	238	254
15	31	47	63	79	95	111	127	143	159	175	191	207	223	239	255



$[x, y]$

$img[i][j][k]$



Se toma la esquina superior izquierda como origen ( $y=0, x=0$ ) de coordenadas

