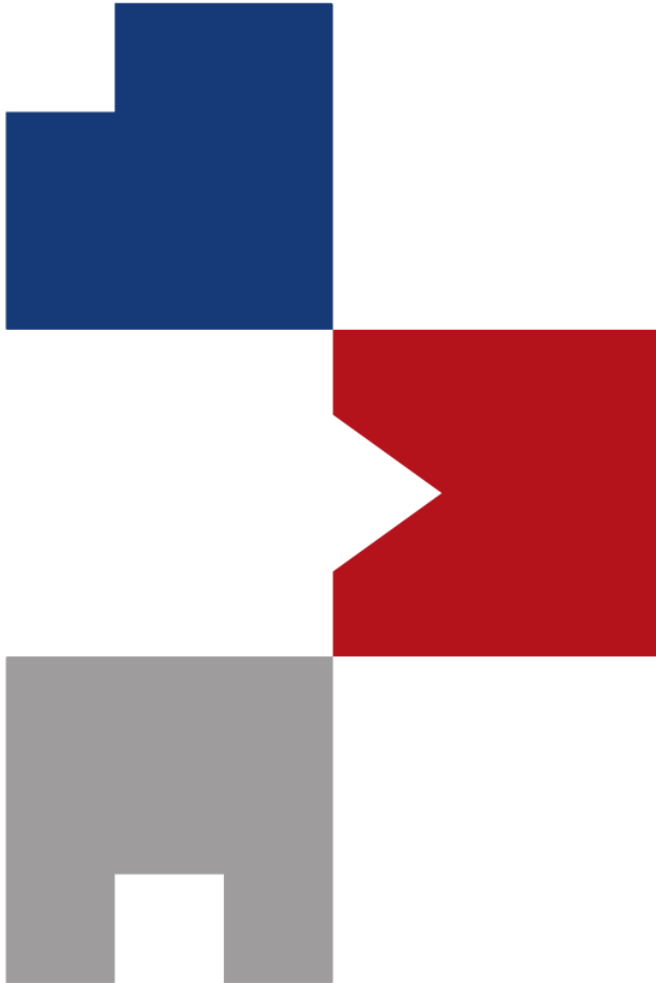


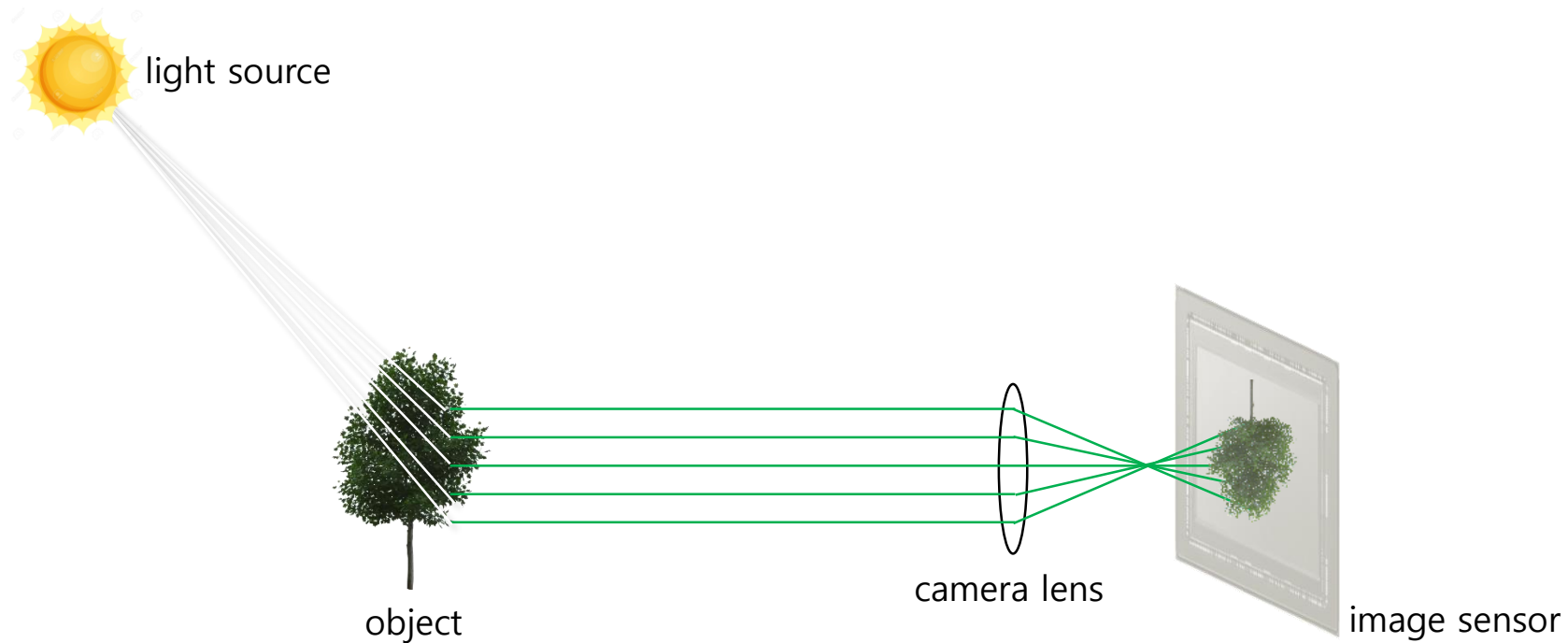
# Color



Sunglok Choi, Assistant Professor, Ph.D.  
Computer Science and Engineering Department, SEOULTECH  
[sunglok@seoultech.ac.kr](mailto:sunglok@seoultech.ac.kr) | <https://mint-lab.github.io/>

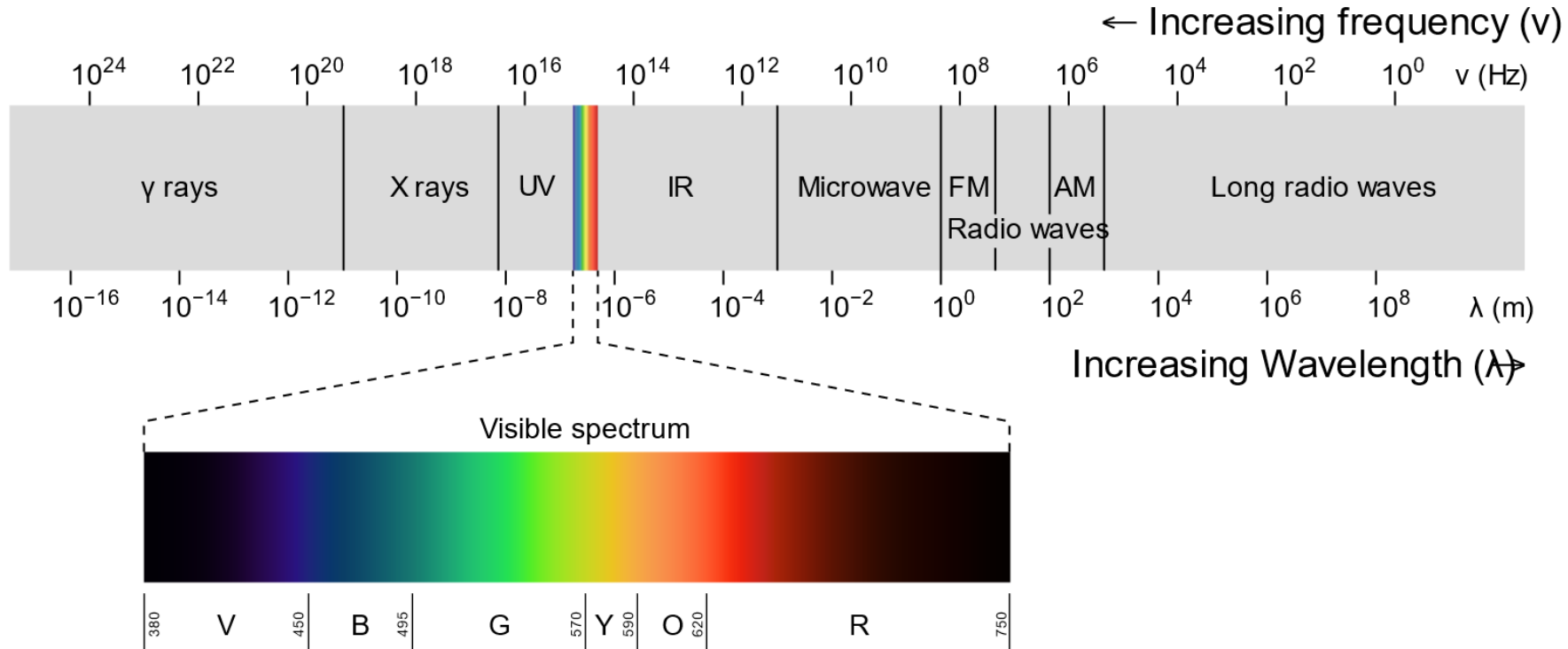
# Photometric Image Formation

- **Camera:** A light measuring device
  1. A **light source** emits *light*.
  2. The *light* is reflected or absorbed or passes through (e.g. transparent) an **object**.
  3. The *light* goes through the **camera lens** and hits the **image sensor**.



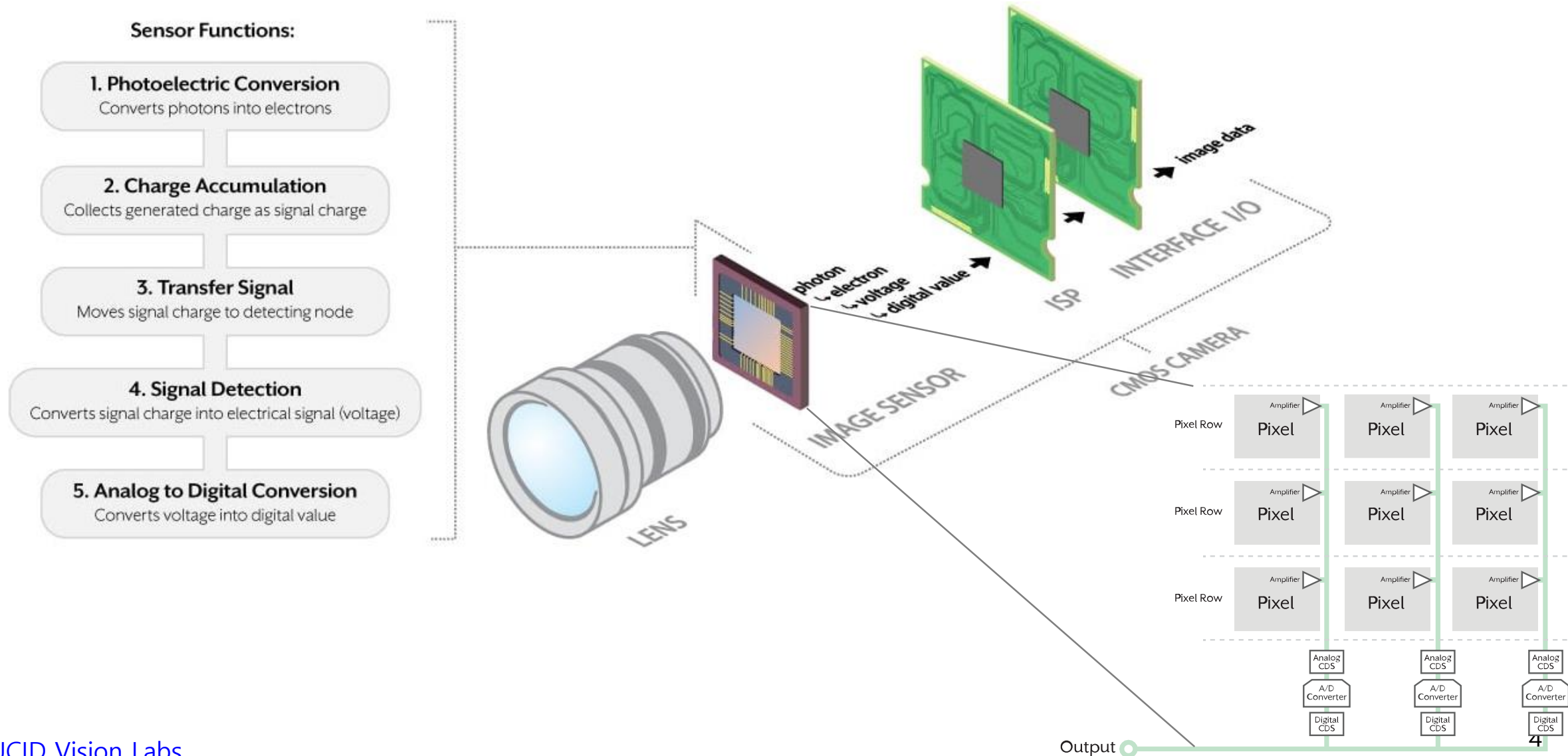
# Light

- The **intensity** of light is the **amount of light energy** per unit area per unit time. (unit:  $\text{W}/\text{m}^2$  or lux)
- The **color** of light is characterized by the **frequency** (unit: Hz) or **wave length** (unit: m).



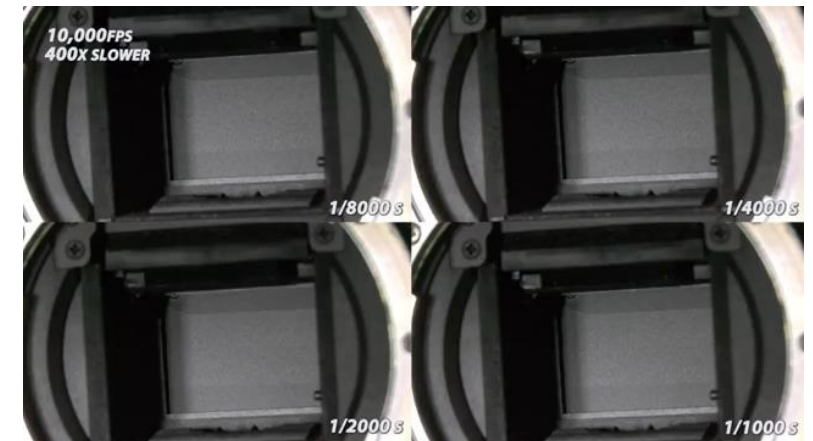
# Photometric Image Formation

- [CMOS sensors](#) (or CCD sensors) are an array of [photodetectors](#) that can detect the intensity of light.



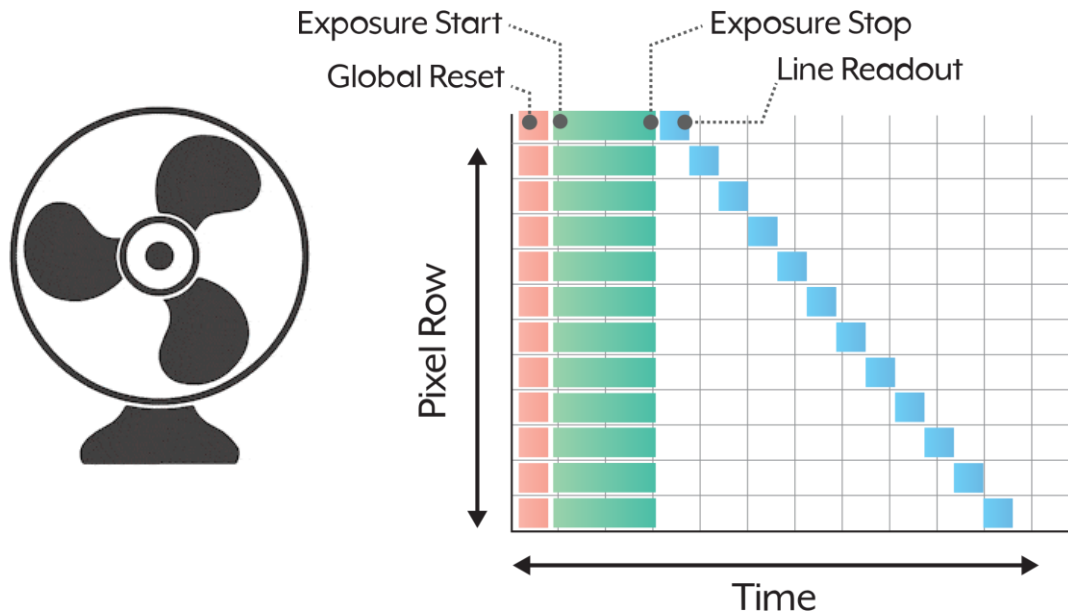
# Photometric Image Formation

- [CMOS sensors](#) (or CCD sensors) are an array of [photodetectors](#) that can detect the intensity of light.
- Q) How does a camera control the intensity of light?
  - [Aperture](#) (조리개 in Korean)
    - A hole of a camera
    - Unit: [f-number](#) (f/N; N is f-number)
  - [Shutter speed](#) (a.k.a. exposure time; 노출시간 in Korean)
    - The length of time that an image sensor is exposed to light
    - Unit: Second
    - Note) Global shutter vs. Rolling shutter
  - [ISO sensitivity](#) (a.k.a. exposure index)
  - Additional light sources (e.g. [flash](#))

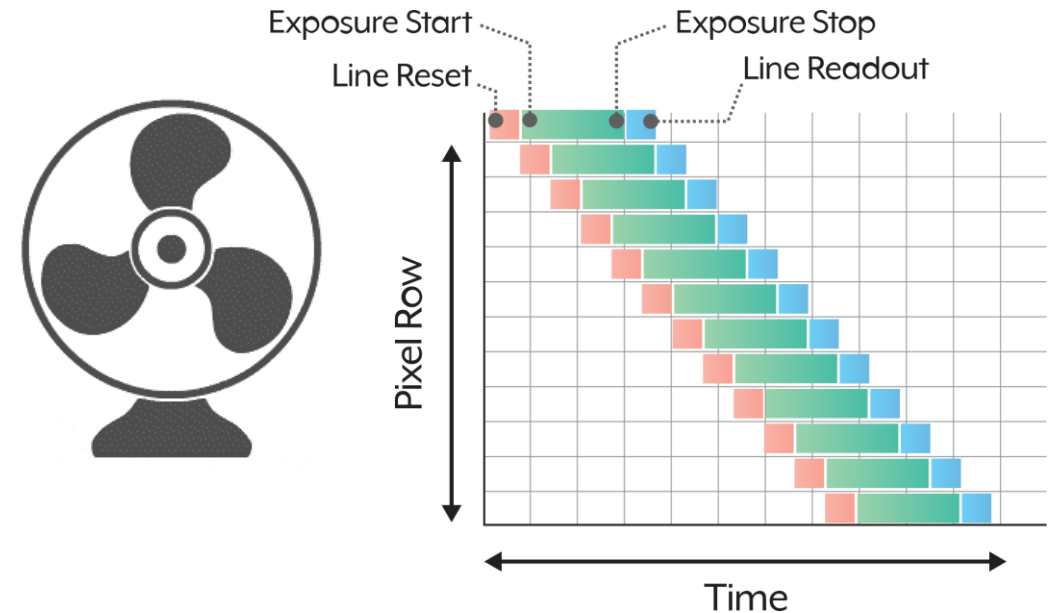


# Photometric Image Formation

- [CMOS sensors](#) (or CCD sensors) are an array of [photodetectors](#) that can detect the intensity of light.
- Q) How does a camera control the intensity of light?
  - [Shutter speed](#) (a.k.a. exposure time; 노출시간 in Korean)
    - The length of time that an image sensor is exposed to light
    - Unit: Second
    - Note) *Global* shutter vs. *Rolling* shutter

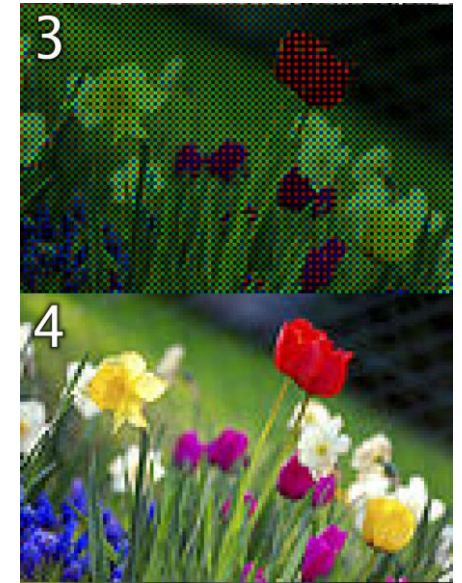
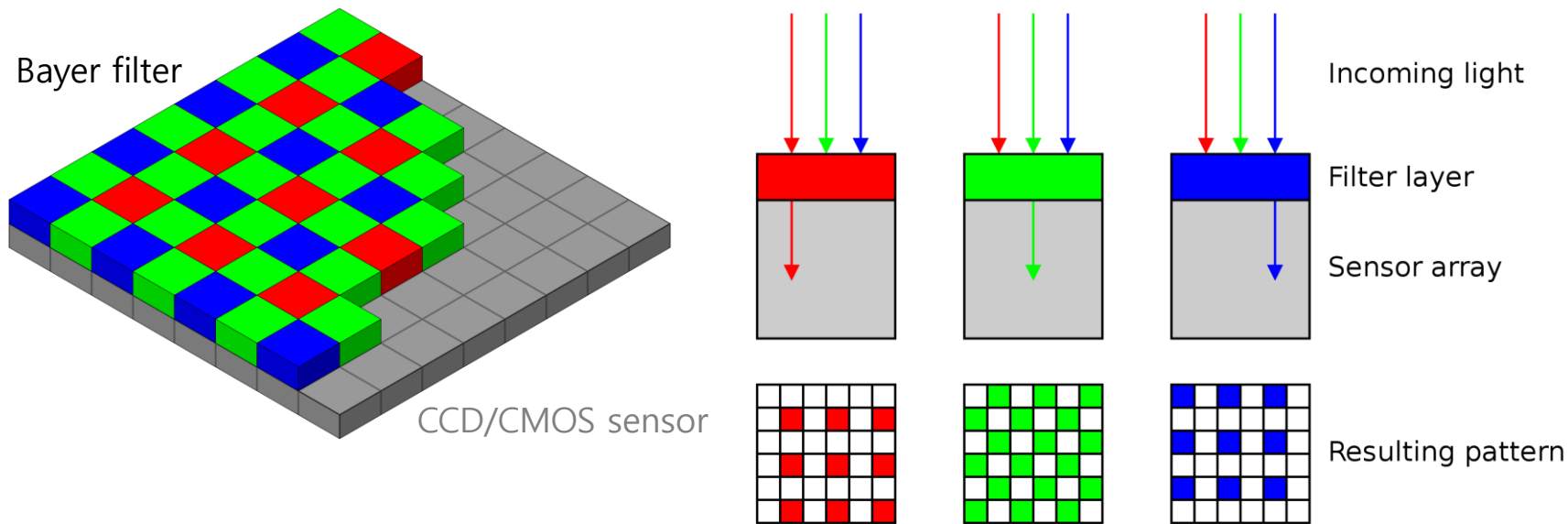


vs.



# Photometric Image Formation

- [CMOS sensors](#) (or CCD sensors) are an array of [photodetectors](#) that can detect the intensity of light.
- Q) How does a camera capture (or distinguish) the color of light?
  - [Color filter array](#) for RGB separation
    - e.g. [Bayer filter](#) is a the most common for RGB.
  - A [demosaicing](#) algorithm (inside of a camera) reconstructs RGB images (4) from raw Bayer images (3).



# Table of Contents

- **Photometric Image Formation**

- Light
- Q) How does a camera control the intensity of light?
- Q) How does a camera capture (or distinguish) the color of light?

- **Color Spaces**

- Color spaces for **media**
- Color spaces for **human-friendly color selection**
- Color spaces for **human visual perception**

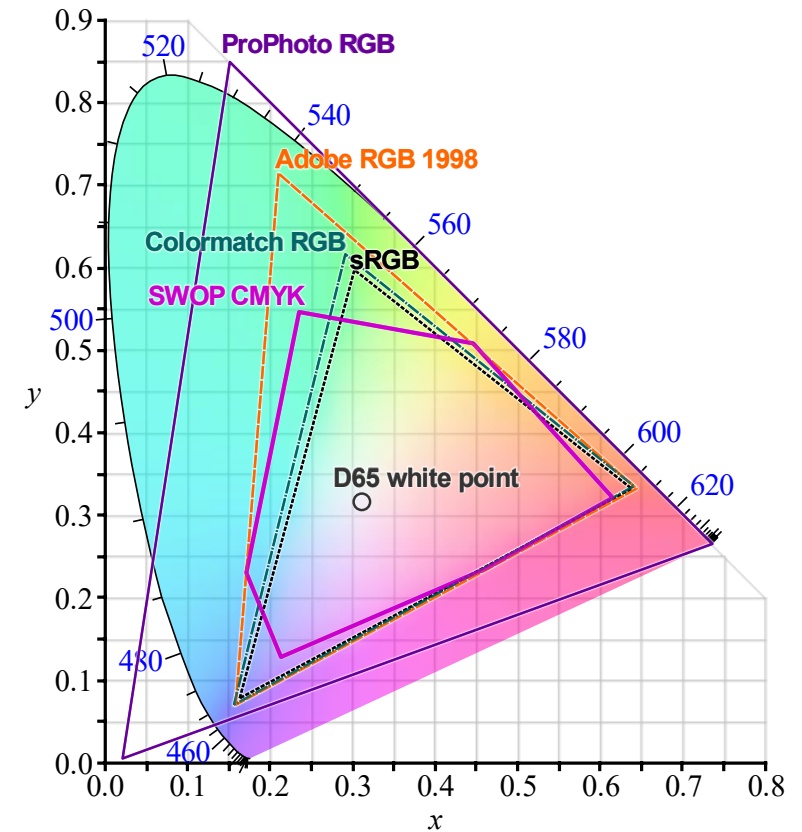
- **Color Image Processing**

- Color balancing
- Color segmentation
- Color histogram
- How to use algorithms only support gray-scale images
  - Example) Color histogram equalization



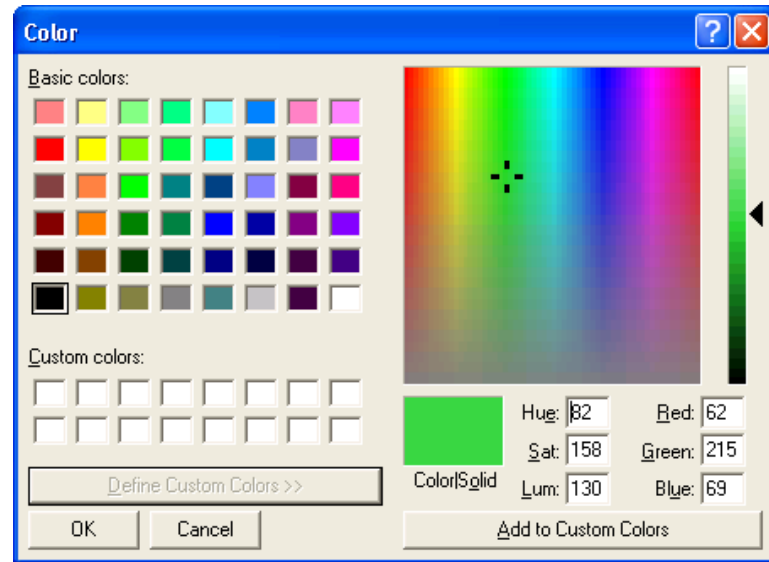
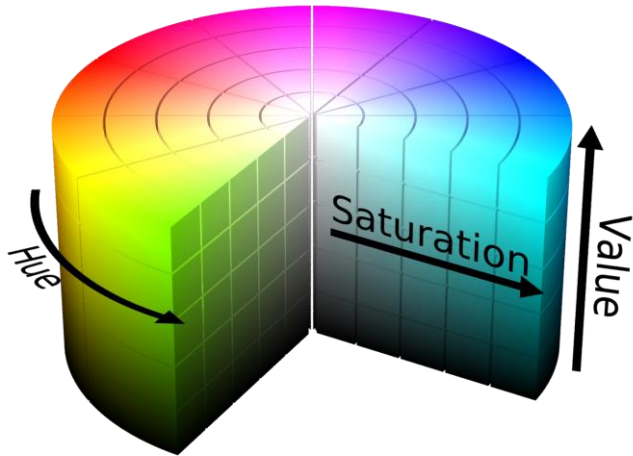
# Color Spaces

- A [color space](#) is a specific system for representing color numerically and graphically. [\[show its list\]](#)
- Color spaces for **media (e.g. display and printing)**
  - [RGB](#): An additive color space for displays (emitting light; e.g. television)
    - [sRGB](#): A standard RGB color (IEC 61966-2-1:1999)  
used in monitors, printers, and WWW
  - [CMY\(K\)](#): An subtractive color space for printing on white papers
  - Note) Chromaticity diagram (→)
    - sRGB: A monitor
    - [SWOP CMYK](#): A printer
    - ProPhoto RGB contains infeasible ranges of color.
  - [YCbCr](#) (digital) and [YUV](#) (analog): Color spaces for video transmission and compression with *economic* bandwidth
    - How? Human eyes has less resolution in color perception than intensity perception. → e.g. Y: 8-bit, Cb/Cr: 4-bit

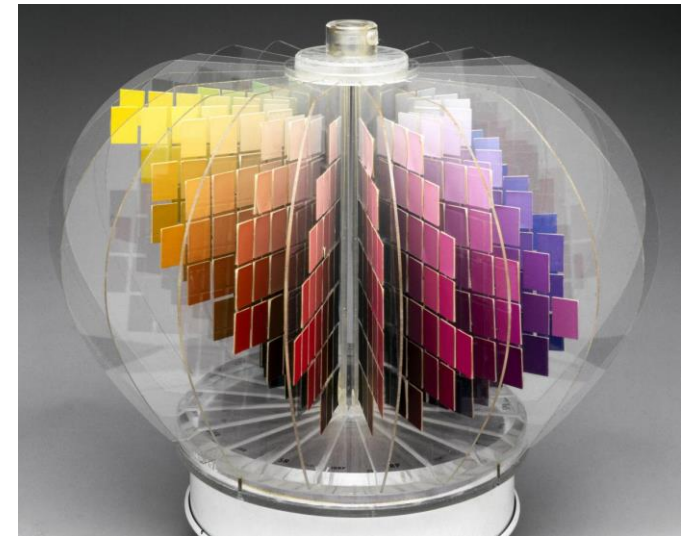


# Color Spaces

- A [color space](#) is a specific system for representing color numerically and graphically. [\[show its list\]](#)
- Cylindrical-coordinate color spaces for **human-friendly color selection**
  - [HSV \(HSL\)](#): Hue-Saturation-Value (색상-채도-명도 in Korean) designed by computer graphics researchers (1970s)
  - [Munsell](#): Hue-Chroma-Value (색상-채도-명도 in Korean) used for paints, crayons, papers, soils, wires, ... (since 1905)



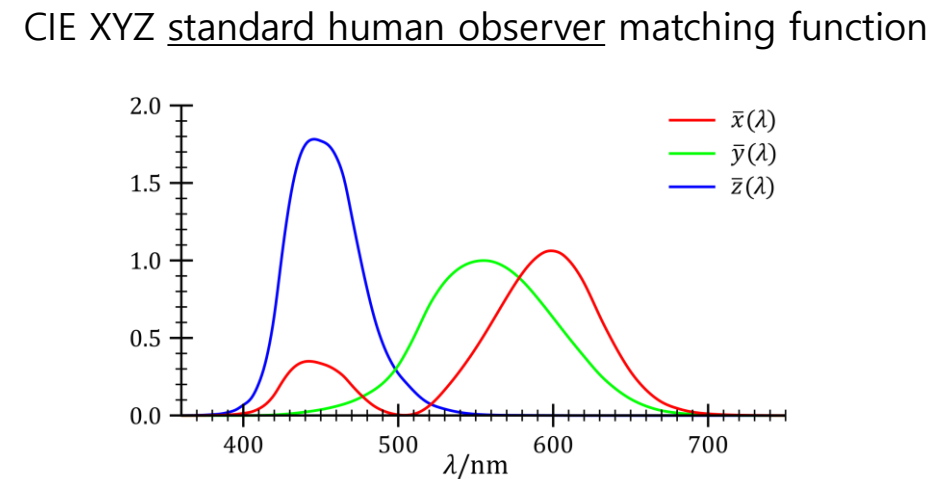
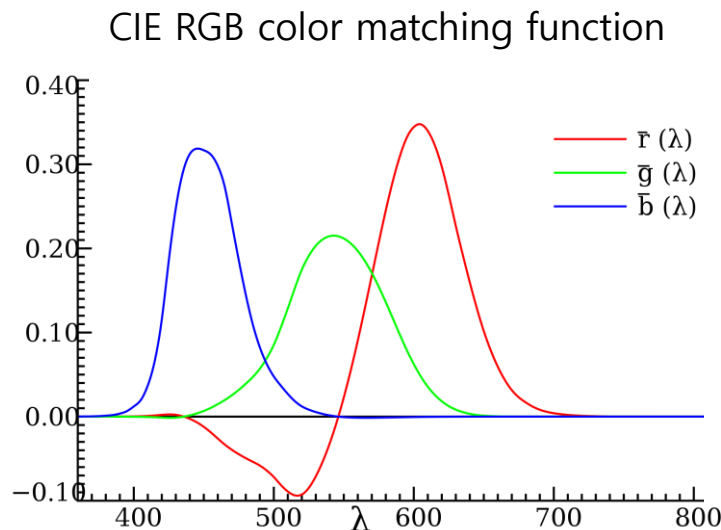
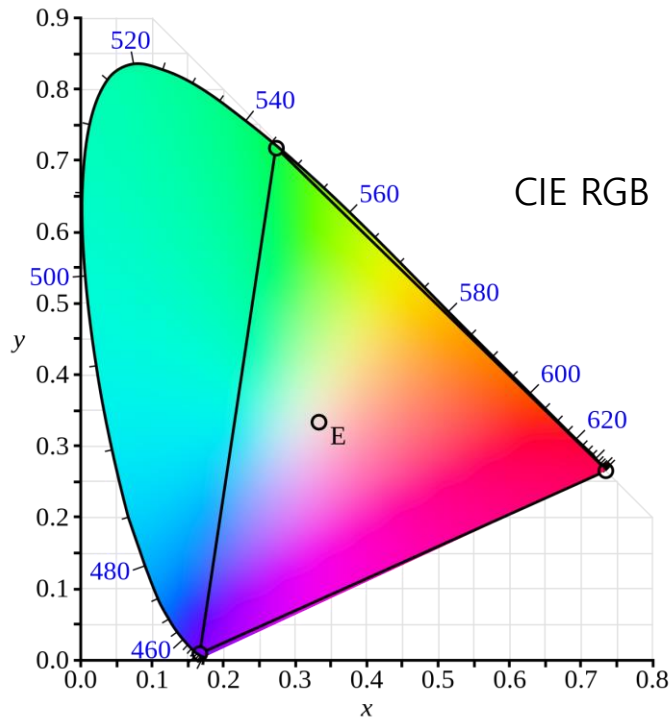
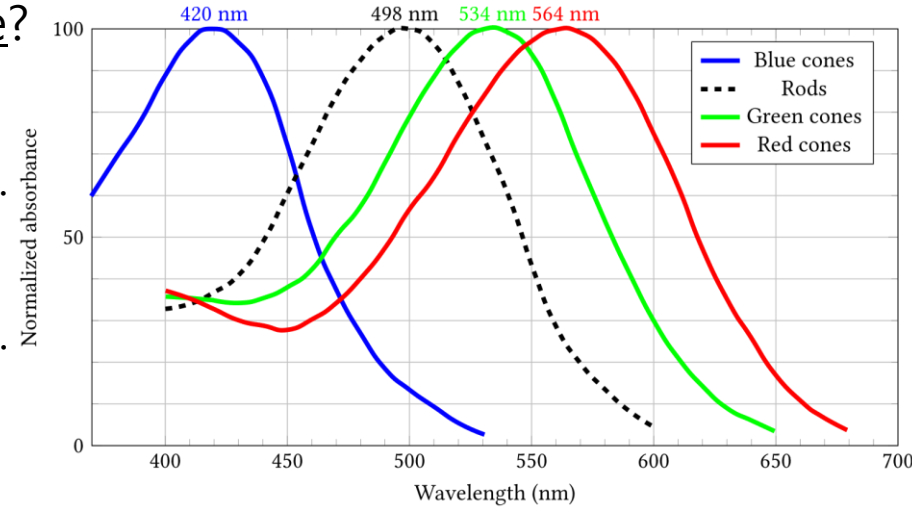
Color Dialog Box



Munsell color tree

# Color Spaces

- A [color space](#) is a specific system for representing color numerically and graphically. [\[show its list\]](#)
- Q) Why are color spaces represented in the three-dimensional space?
  - Human eyes have three different cone cells (원추세포 in Korean).
    - They response **long(L)**, **medium(M)**, and **short(S)** wavelength of light.
  - Human eyes are tristimulus and [trichromacy](#).
  - Its color space is also defined as [LMS \(long, medium, short\) color space](#).



# Color Spaces

- A [color space](#) is a specific system for representing color numerically and graphically. [\[show its list\]](#)
- Color spaces for **human visual perception**
  - [XYZ](#) (a.k.a. CIE 1931 XYZ): A standard color space based on physiologically perceived colors by human observers
    - Note) Y ~ [luminance](#) (명암 in Korean), Z ~ blue of CIE RGB, X ~ mixture
    - The Hunt-Pointer-Estevéz matrix (1980)

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 1.91020 & -1.11212 & 0.20191 \\ 0.37095 & 0.62905 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix}$$

- [xyY](#) (a.k.a. CIE xyY): A standard color space to represent the quality of color (regardless of its [luminance](#))

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

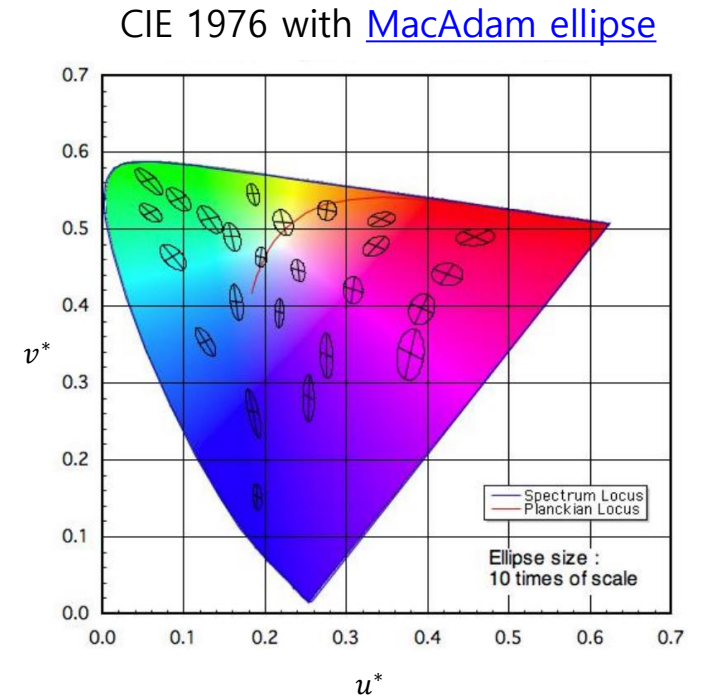
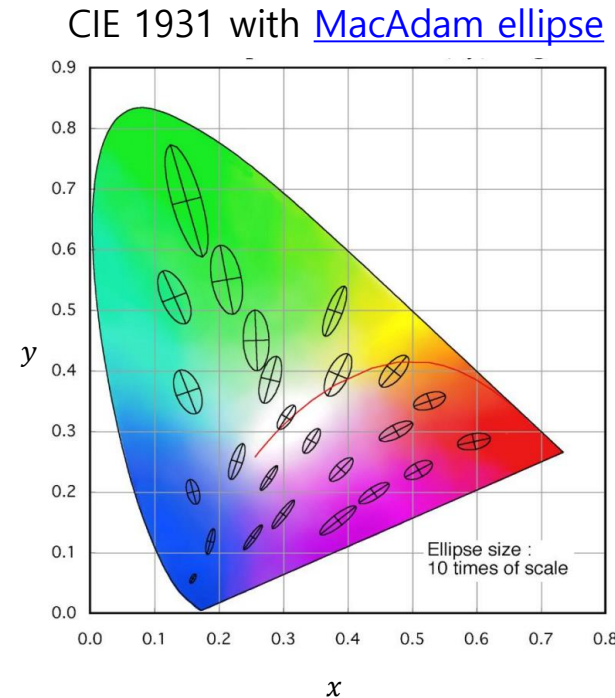
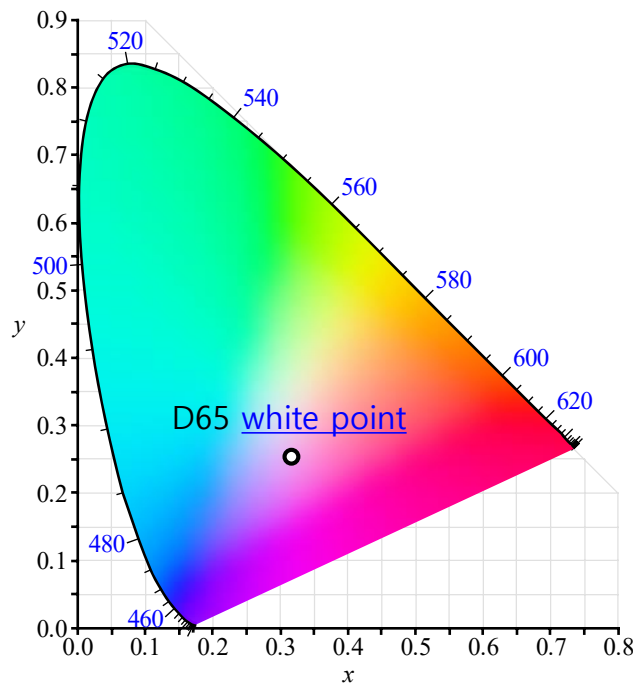
- [LAB](#) (a.k.a. CIE L\*a\*b\*) and [LUV](#) (a.k.a. CIE 1976 L\*u\*v\*): Standard color spaces for better representation

# Color Spaces

- Color spaces for **human visual perception**

- [XYZ](#) (a.k.a. CIE 1931 XYZ): A standard color space based on physiologically perceived colors by human observers
- [xyY](#) (a.k.a. CIE xyY): A standard color space to represent the quality of color (regardless of its [luminance](#))
- [LAB](#) (a.k.a. CIE  $L^*a^*b^*$ ) and [LUV](#) (a.k.a. CIE 1976  $L^*u^*v^*$ ): Standard color spaces for better representation

- CIE xy [chromacity](#) diagram





# Color Spaces

- Example) Color space conversion (HSV)

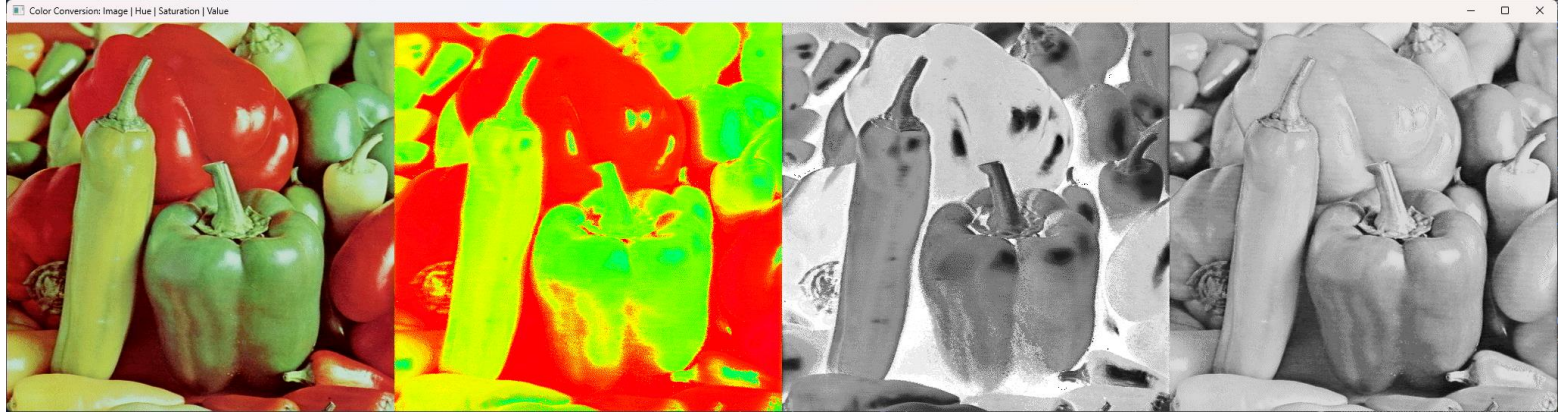
```
import numpy as np
import cv2 as cv
```

```
img = cv.imread('../data/peppers.tif')
assert img is not None, 'Cannot read the given image'
```

```
# Convert the BGR image to its HSV image
img_hsv = cv.cvtColor(img, cv.COLOR_BGR2HSV)
```

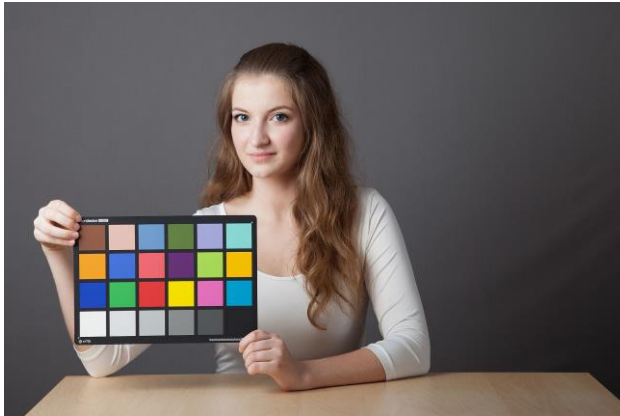
```
# Show hue, saturation, and value channels as color images
```

```
img_hue = np.dstack((img_hsv[:, :, 0],
                    np.full_like(img_hsv[:, :, 0], 255),
                    np.full_like(img_hsv[:, :, 0], 255)))
img_hue = cv.cvtColor(img_hue, cv.COLOR_HSV2BGR)
img_sat = np.dstack((img_hsv[:, :, 1], ) * 3)
img_val = np.dstack((img_hsv[:, :, 2], ) * 3)
merge = np.hstack((img, img_hue, img_sat, img_val))
cv.imshow('Color Conversion: Image | Hue | Saturation | Value', merge)
cv.waitKey()
cv.destroyAllWindows()
```

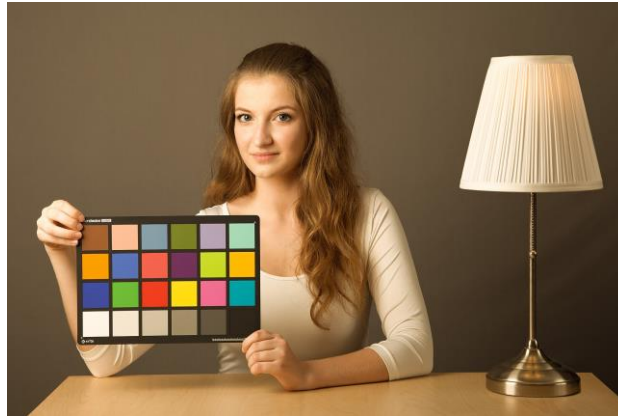


# Color Image Processing

- **Color balancing (or constancy)**: A global adjustment of the intensities of the color (typically RGB)
  - **White balance** (shortly WB): A most common color balancing which make a white object appears as white



Neutral light

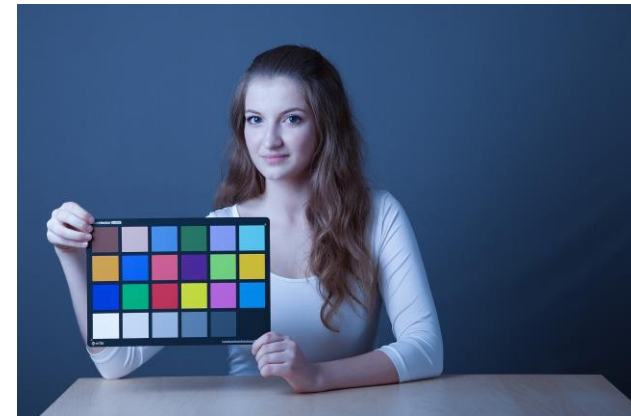
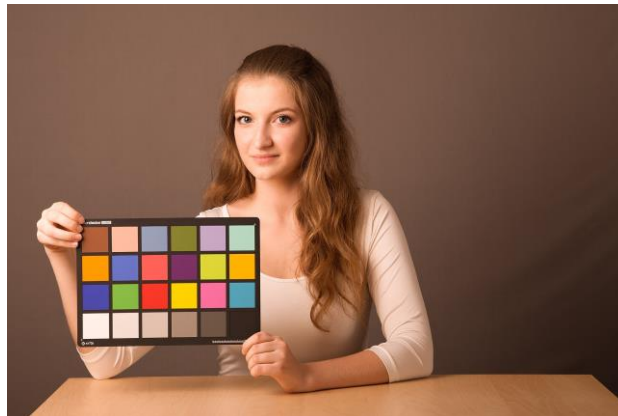
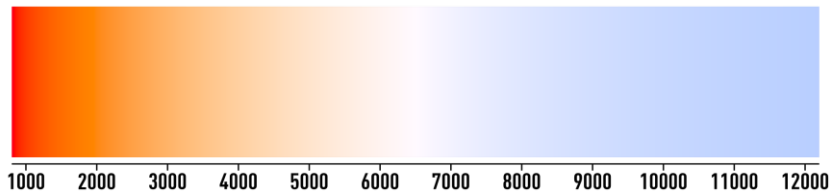


Warm light



Cold light

- **Color temperature**: A parameter describing the color of light source (as temperature; unit: [K])



# Color Image Processing

- **Color segmentation:** Extracting a specific color range

- e.g. Colored objects, road lanes, ..., skin



- Q) How to represent the color range? (What is a good color space?)
  - Thresholds (~ box), a ellipse, Gaussian mixture model (~ a set of ellipses), ...

- **Color histogram** (as a descriptor)

- e.g. Person re-identification (ReID)





# Color Image Processing

- Some algorithms only support **gray-scale images**, not color images.

- e.g. Histogram equalization, edge detection, ...

```
# Read the given image as gray scale
```

```
img = cv.imread('../data/lena.tif', cv.IMREAD_GRAYSCALE)
```

- Why? (Difficulty in dealing with three-dimensional values)

- Three-dimensional comparison? (e.g. [0, 75, 0] vs. [25, 25, 25])
- Three-dimensional transformation → Natural color?

- Ad-hoc solutions to use the algorithms only for gray-scale images

- ~~• Idea #1) Apply the algorithm to each RGB channel~~
- Idea #2) Apply the algorithm to **the luminance channel**
  - e.g. RGB → **Y**CbCr → RGB
  - Note) The definitions of luminance are ambiguous.

L\*a\*b\*



I (Intensity)



HSV



HSL



Color

# Color Image Processing

- Example) Color histogram equalization

```
img_list = [...]
```

```
# Initialize a control parameter
```

```
img_select = 0
```

```
while True:
```

```
    # Read the given image
```

```
    img = cv.imread(img_list[img_select])
```

```
    # Apply histogram equalization to each channel
```

```
    img_hist1 = np.dstack((cv.equalizeHist(img[:, :, 0]),  
                           cv.equalizeHist(img[:, :, 1]),  
                           cv.equalizeHist(img[:, :, 2])))
```

```
    # Apply histogram equalization only to the luminance channel in YCbCr
```

```
    img_cvt = cv.cvtColor(img, cv.COLOR_BGR2YCrCb)
```

```
    img_hist2 = np.dstack((cv.equalizeHist(img_cvt[:, :, 0]),  
                           img_cvt[:, :, 1],  
                           img_cvt[:, :, 2]))
```

```
    img_hist2 = cv.cvtColor(img_hist2, cv.COLOR_YCrCb2BGR)
```

```
    # Show all images
```

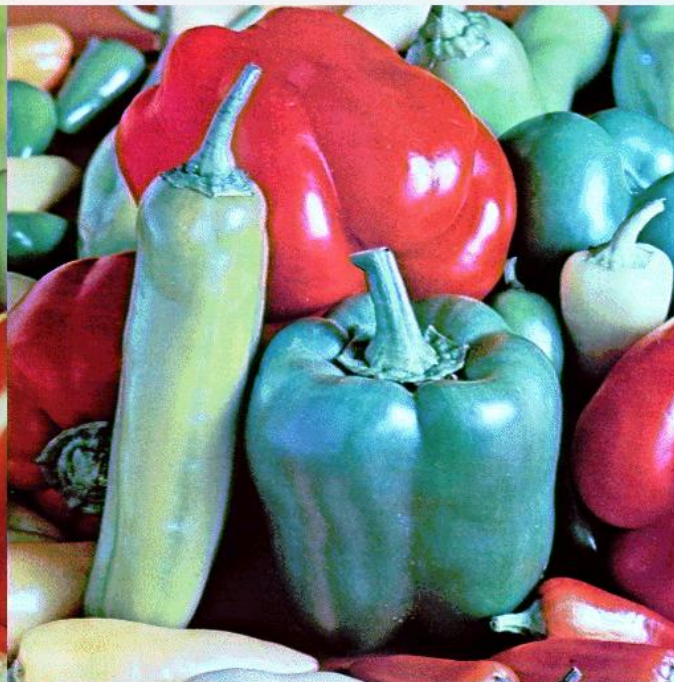
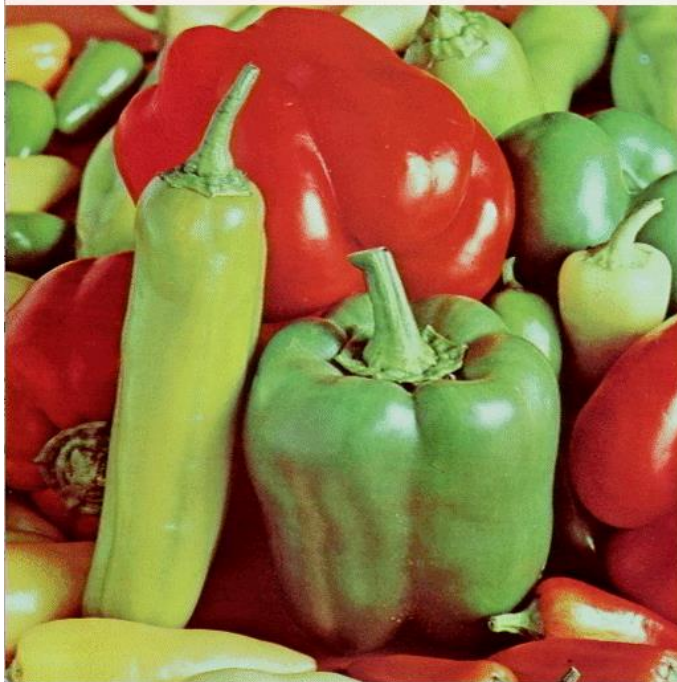
```
    merge = np.hstack((img, img_hist1, img_hist2))
```

```
    cv.imshow('Color Histogram Equalization: Image | Each Channel | Luminance Channel', merge)
```

```
    key = cv.waitKey()
```

```
    if key == 27: # ESC
```







# Summary

## ▪ Photometric Image Formation

- CMOS sensor: Light intensity sensor
- Color camera = **Bayer filter** + CCD/CMOS sensor

## ▪ Color Spaces

- Color spaces for **media**: RGB, CMY(K), YCbCr, and YUV
- Color spaces for **human-friendly color selection**: HSV, HSL, and Munsell
- Color spaces for **human visual perception**: XYZ, xzY, LAB ( $L^*a^*b^*$ ), and LUV ( $L^*u^*v^*$ )
- Example) Color space conversion

## ▪ Color Image Processing

- Color balancing (e.g. white balance, color temperature, ...)
- Color segmentation (e.g. colored objects, ...)
- Color histogram as a descriptor (e.g. person re-identification, ...)
- How to use algorithms only support gray-scale images
  - Example) Color histogram equalization