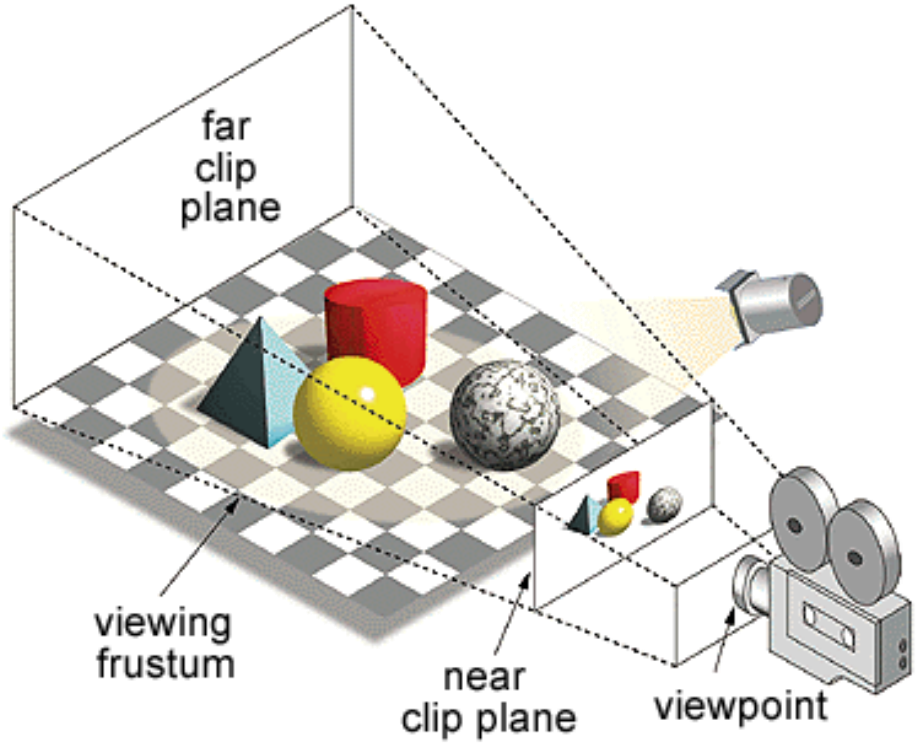
# Image and Color

## Introduction to Computer Graphics

Yu-Ting Wu

# Recap.

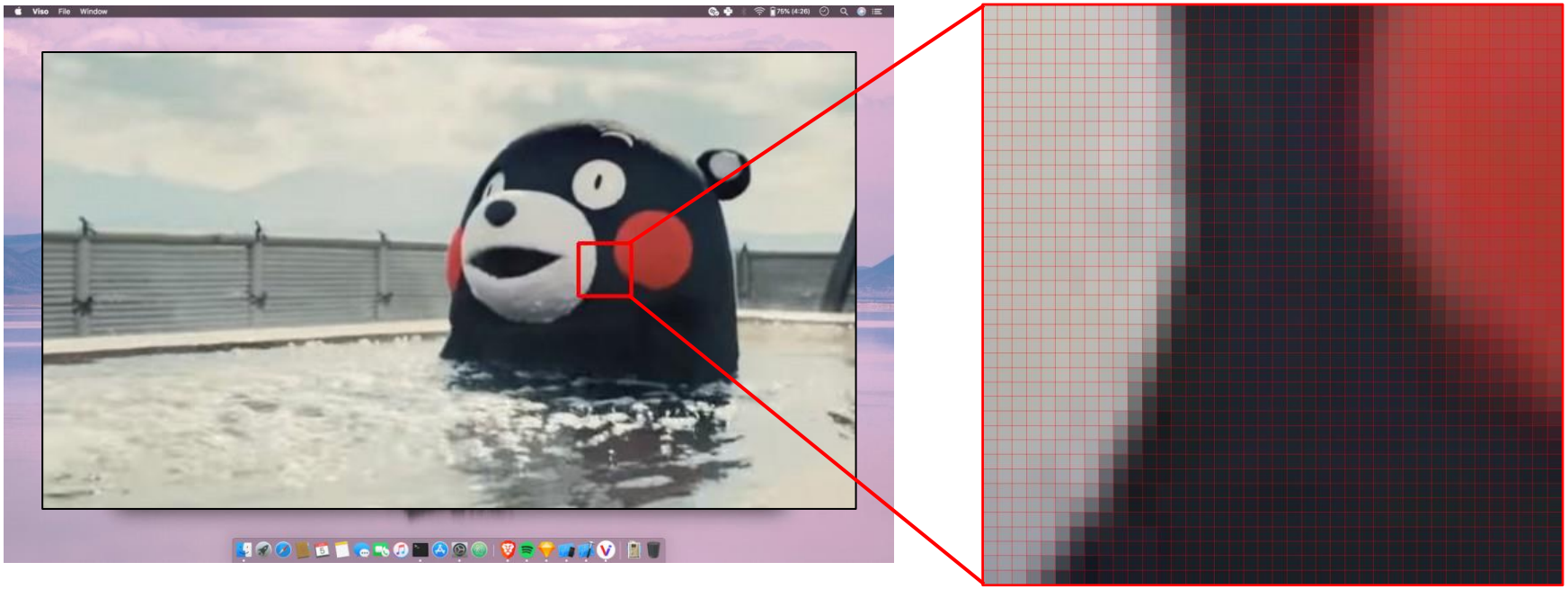
- In computer graphics, we generate an **image** from a **virtual 3D world**
  - We are going to introduce the representation of an image



# Image

# Image Display

- Monitor display pictures as a **rectangular array of pixels** (small, usually square, dots of color)
  - Merge optically when viewed at a suitable distance to produce the impression of continuous tones



# Image Display (cont.)

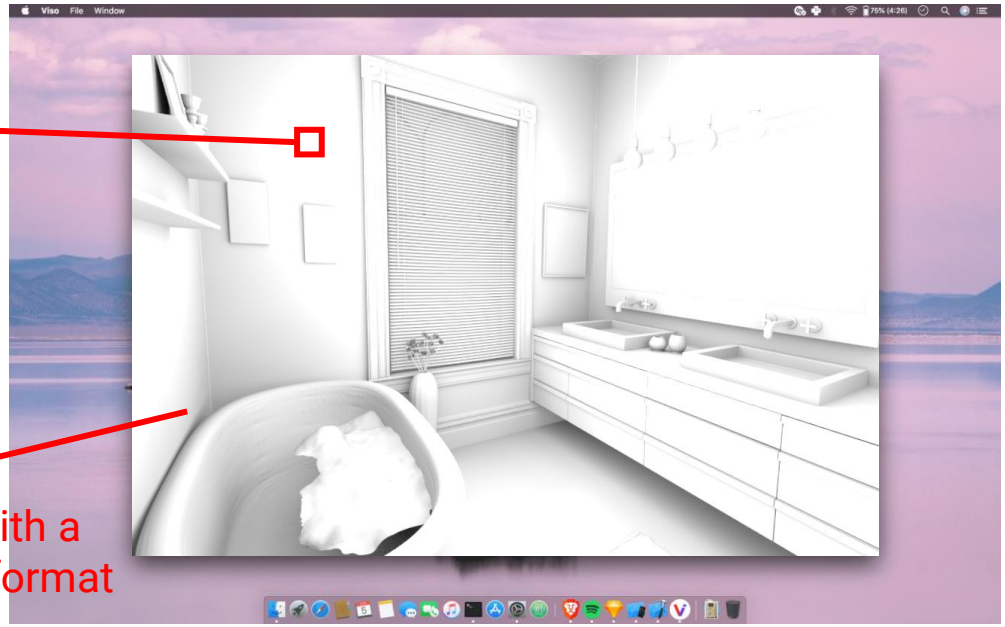
- Monitor display pictures as a **rectangular array of pixels** (small, usually square, dots of color)
  - Merge optically when viewed at a suitable distance to produce the impression of continuous tones

Programs set the **shade of grey or color (rendering)**

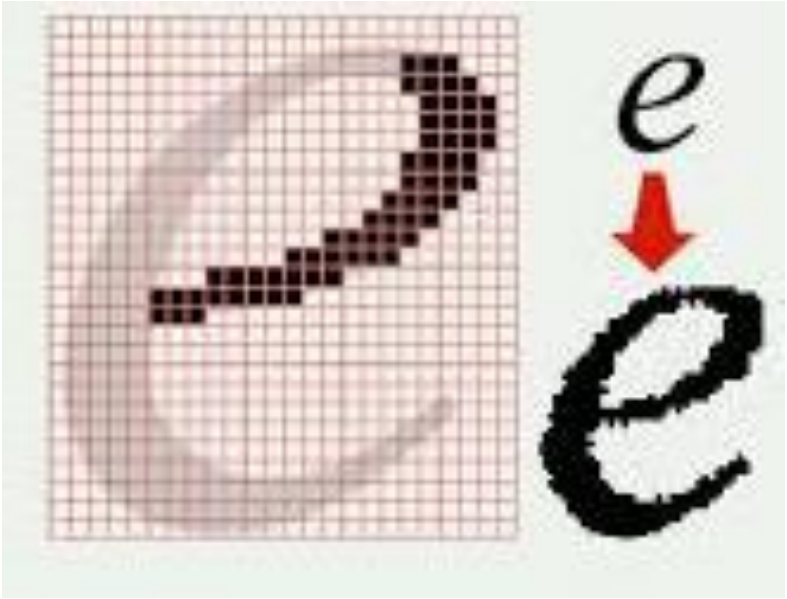
reconstruct pixel data from the format

Graphical Modeling

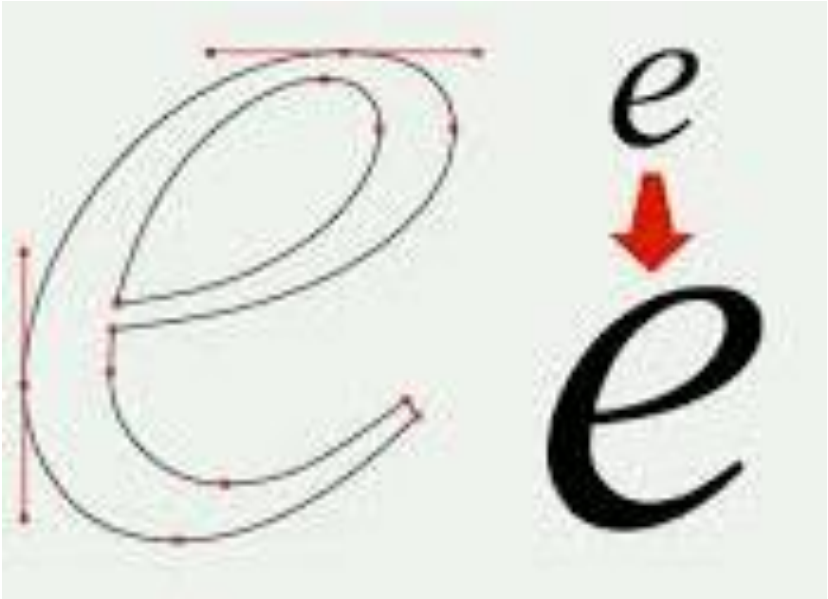

store with a specific format



# Two Approaches for Graphical Modeling



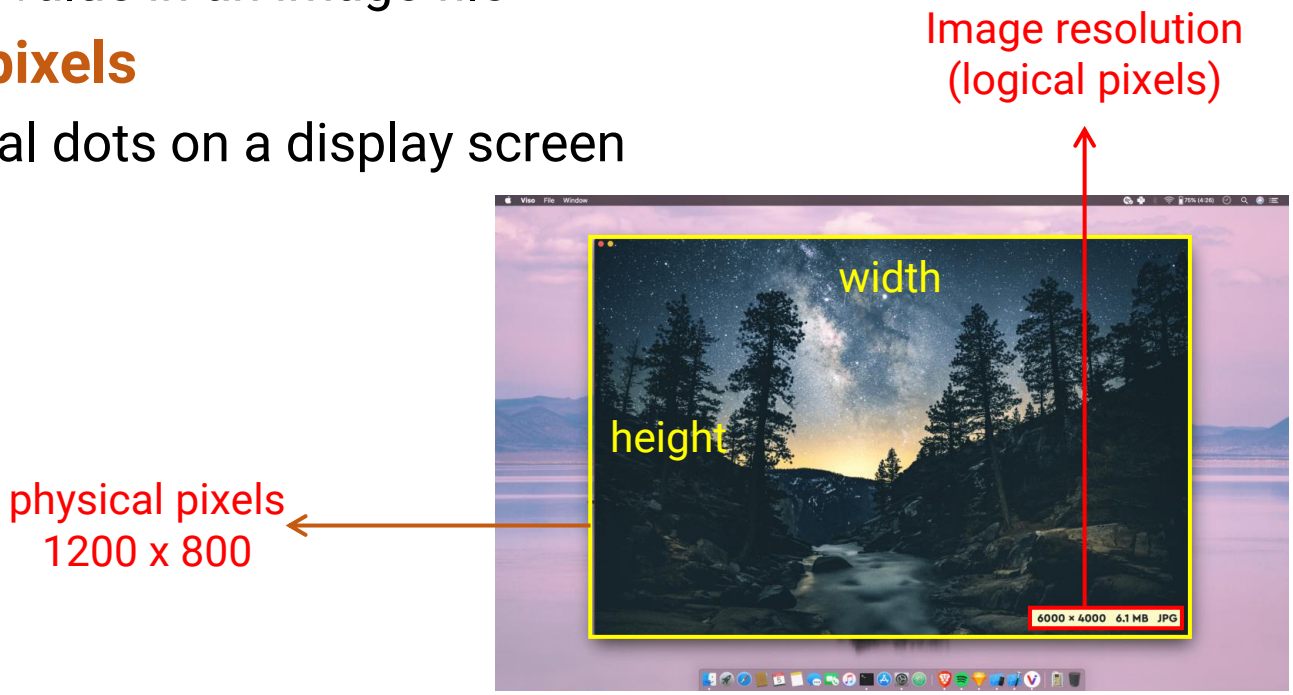
bitmapped images



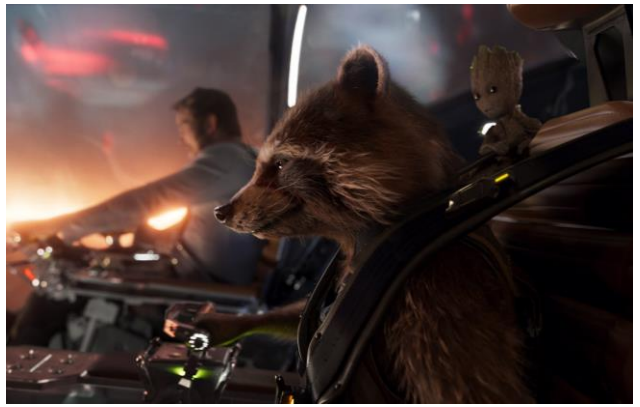
vector graphics

# Bitmapped Images

- An image is modeled by an array of pixel values
- Distinction between
  - **Logical pixels**
    - Stored value in an image file
  - **Physical pixels**
    - Physical dots on a display screen

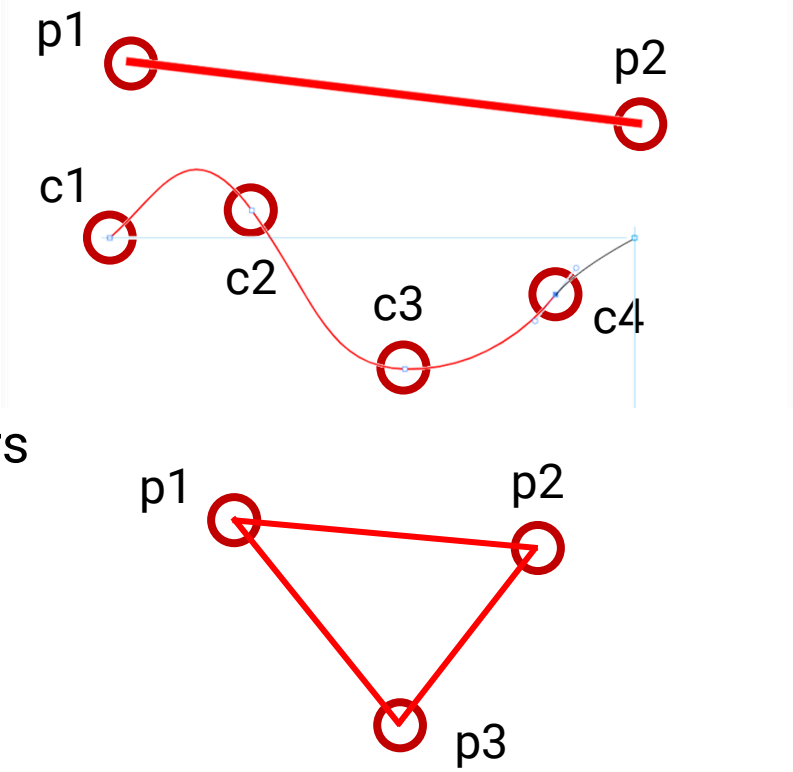


# Bitmapped Images Examples

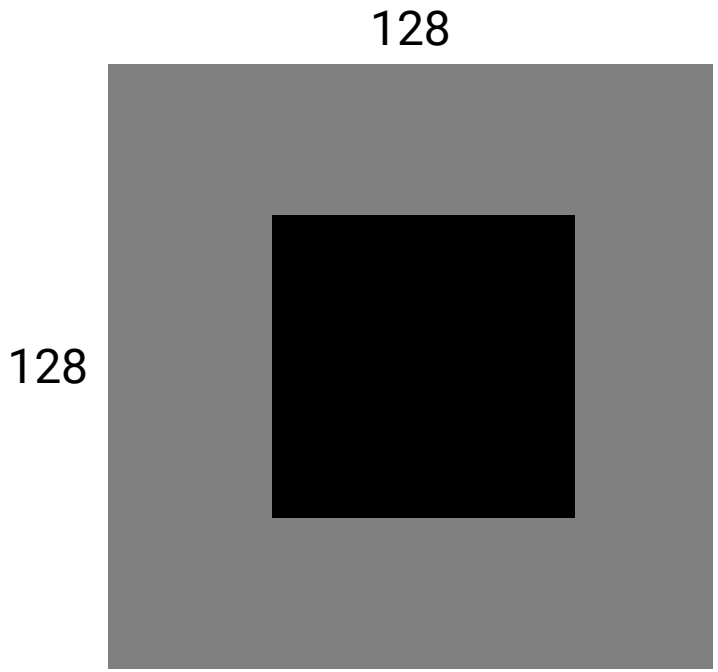



# Vector Graphics

- An image is modelled by the mathematical description of a collection of individual objects making up the image
  - **Lines**
    - End points
  - **Curves**
    - Control points
  - **Shapes**
    - Shape-dependent parameters



# An Simple Vector Graphics Example



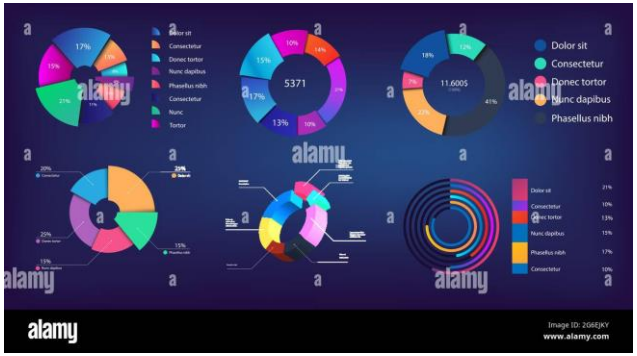

0.5 1.5 0.5 **setrgbcolor**

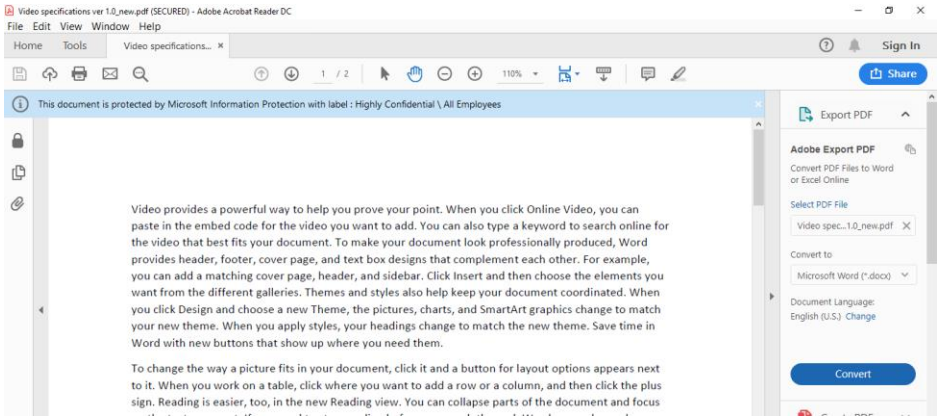
0 0 128 128 **rectfill**

0 0 0 **setrgbcolor**

32 32 64 64 **rectfill**

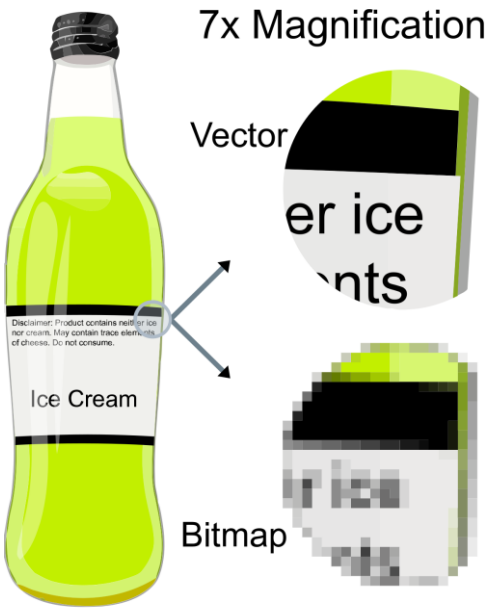
# Vector Graphics Examples




# Bitmapped v.s. Vector Graphics

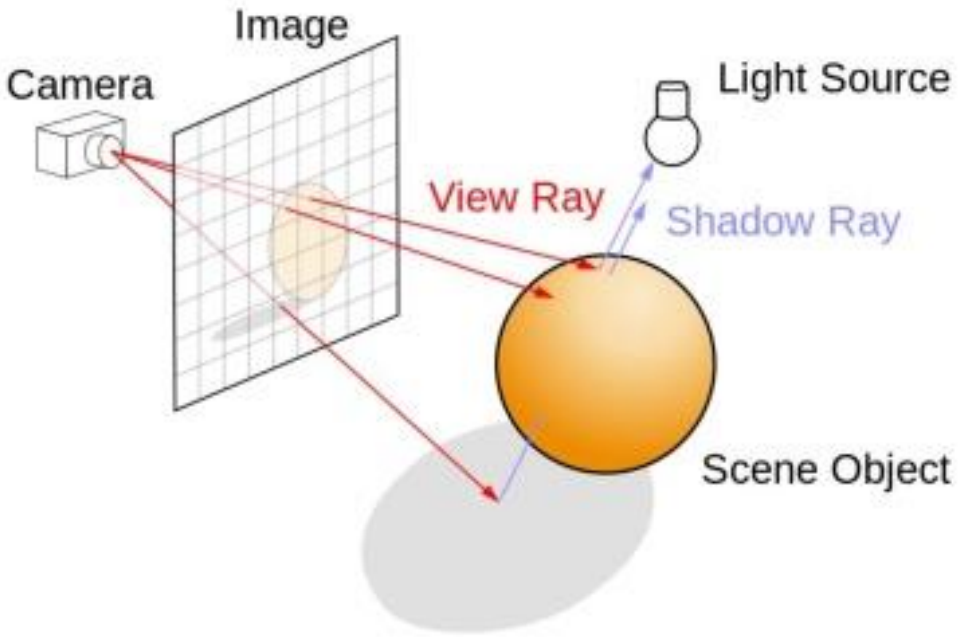
- Bitmapped images provide **better control of pixel values**, thus being more suitable for natural images
- Vector graphics are **resolution independent**, thus being more suitable for texts and icons





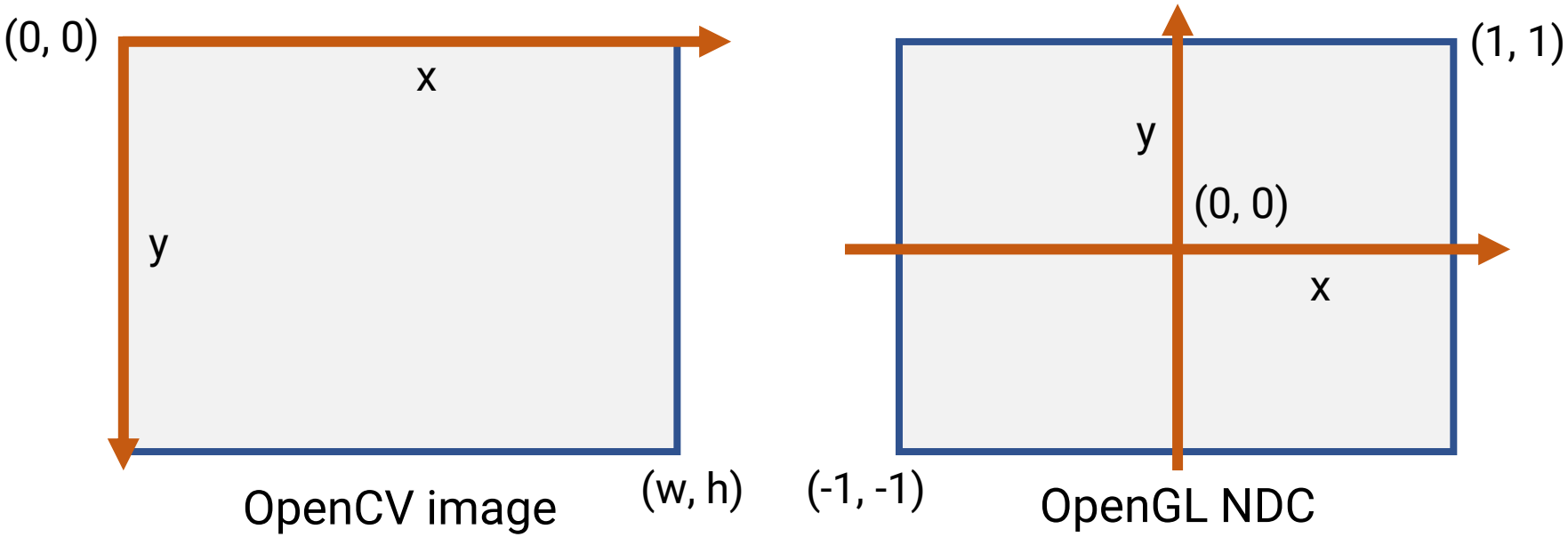
# 3D Graphics

- A **combination** of vector and bitmapped graphics
- Shapes are defined in the virtual 3D space and projected (rasterized) to the 2D image plane



# Image Coordinate

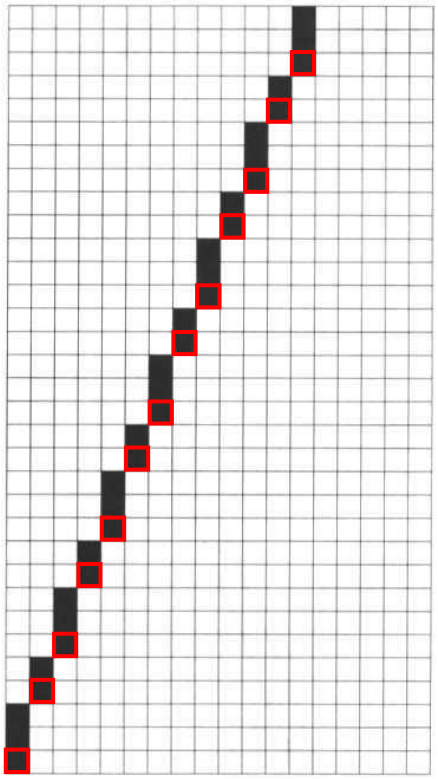
- The coordinate of a 2D image depends on libraries



# Rendering of Math

- When it becomes necessary to render a vector drawing, the **stored values** (e.g., endpoints of a line) are used in conjunction with the **general form** of the description of each class of object
  - Can be considered as **sampling**
- Example: y = 5x/2 + 1

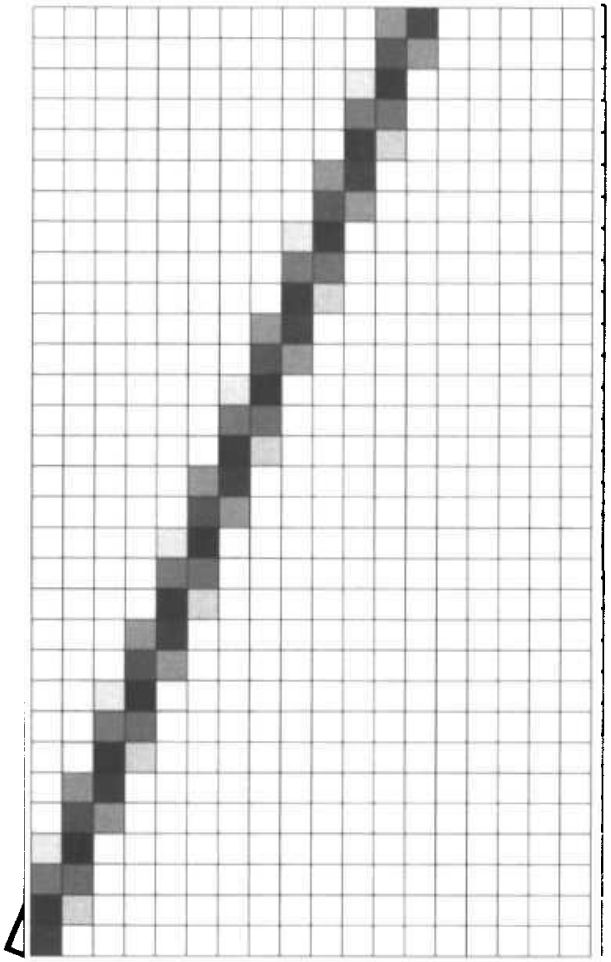
  pass through (0, 1), (1, 4), (2, 6), (3, 9) …



- Jaggedness is inevitable!
  - Due to the use of a grid of discrete pixels

# Anti-aliasing

- Anti-aliasing is a **practical** technique to reduce the jaggies
- Use intermediate grey values
  - In the frequency domain, it relates to reducing the frequency of the signal
- Coloring each pixel in a shade of grey whose **brightness is proportional to the area** of the intersection between the pixels and a "**one-pixel-wide**" line



# Anti-aliasing (cont.)

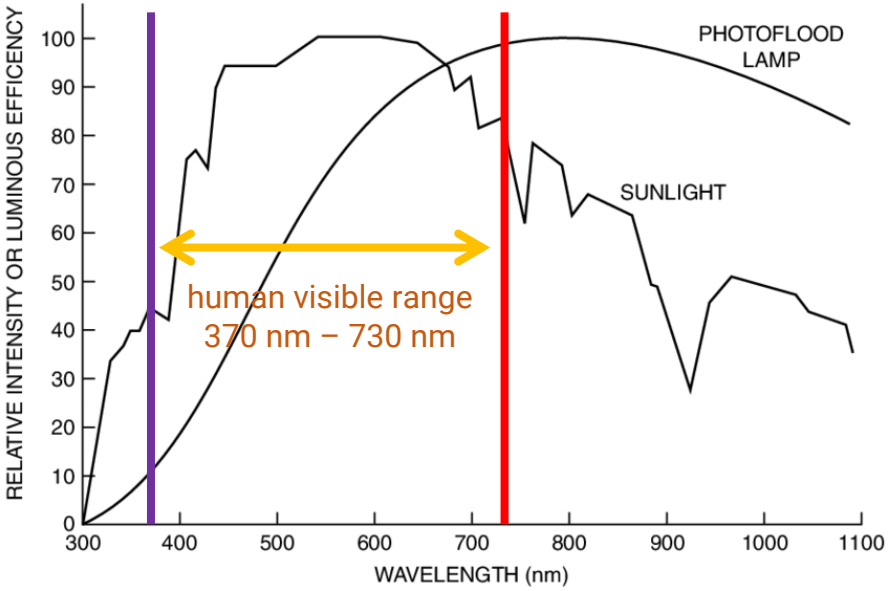
# Color

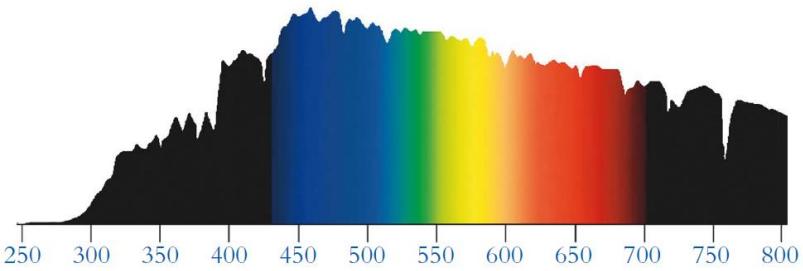
# Color Science

- Color is a common experience for humans, but being a rather complex phenomenon
- Color science is a topic that attempts to relate the **subjective sensation** of color to **measurable** and **reproducible** physical phenomena

# Spectral Power Distribution

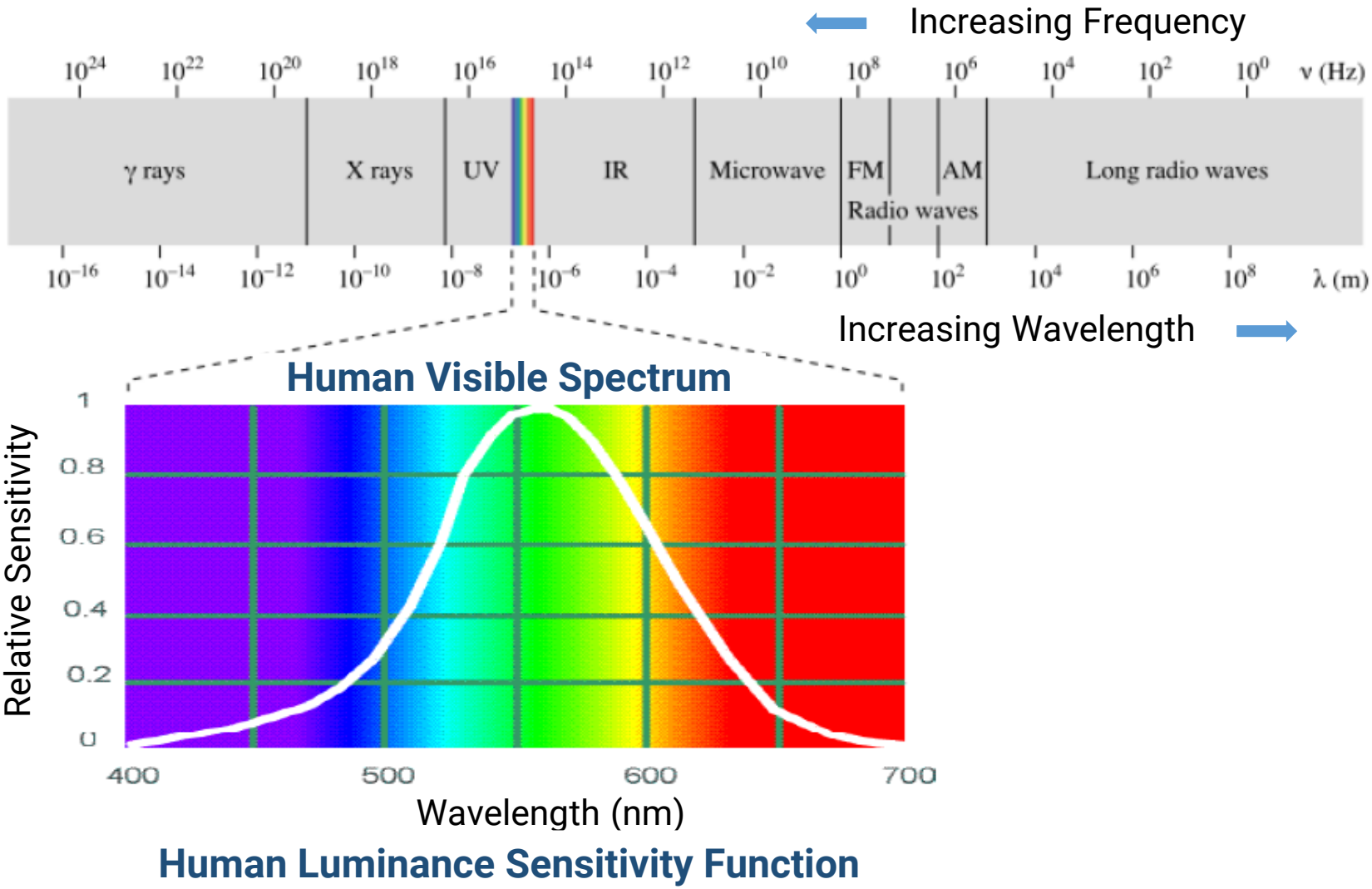
- Light is an electromagnetic wave, and we can measure its wavelength and intensity
- **Spectral power distribution (SPD)** is a description of how the intensity of light varies with its wavelength



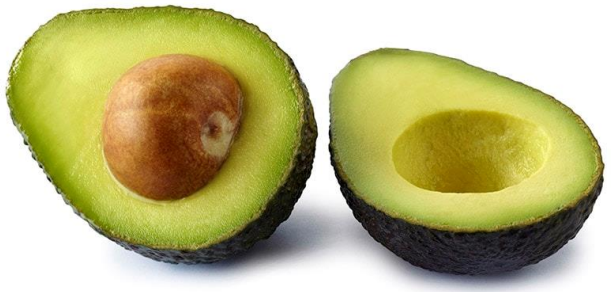

SPD of daylight

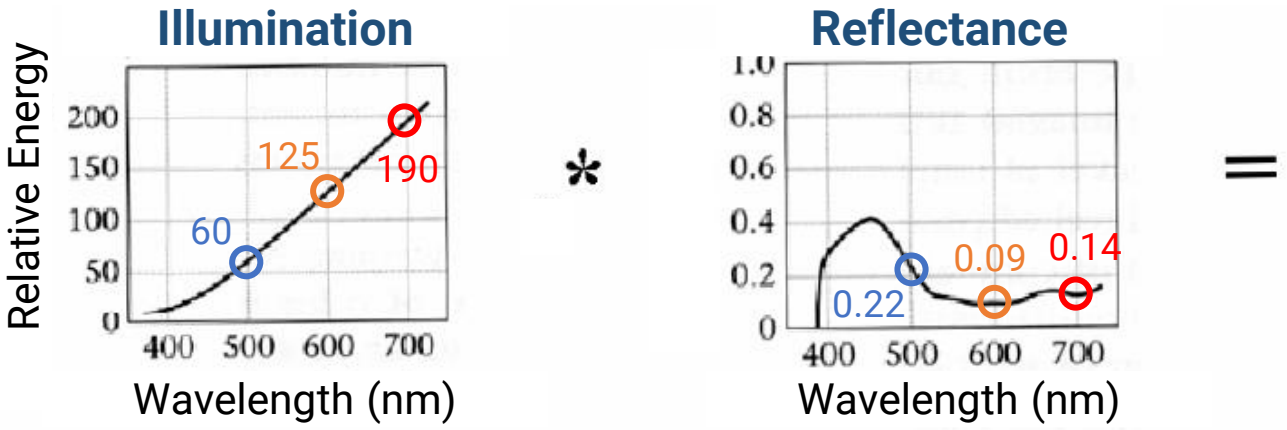
# Spectral Power Distribution (cont.)



# Color

- Reflected color is the result of interaction of **light source spectrum** with **surface reflectance**



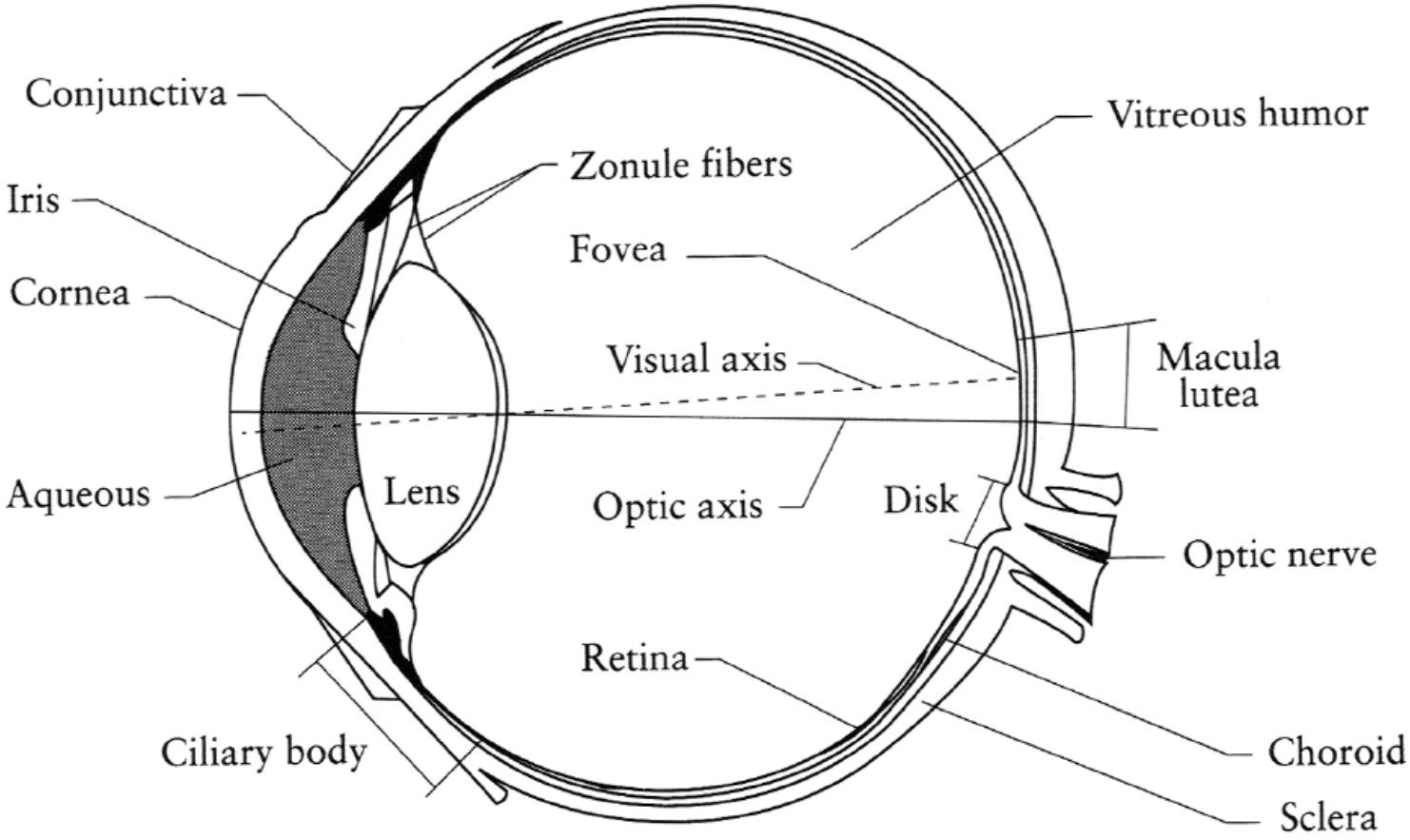


# Tristimulus Theory

- SPDs are too cumbersome for representing the color in computer graphics
- Need a more compact, efficient, and accurate way to represent color signals
  - Find proper basis functions to map the infinite-dimensional space of all possible SPDs to the **low-dimensional space of coefficients**
- We use the **tristimulus theory**
  - All visible SPDs can be accurately represented with **three values**

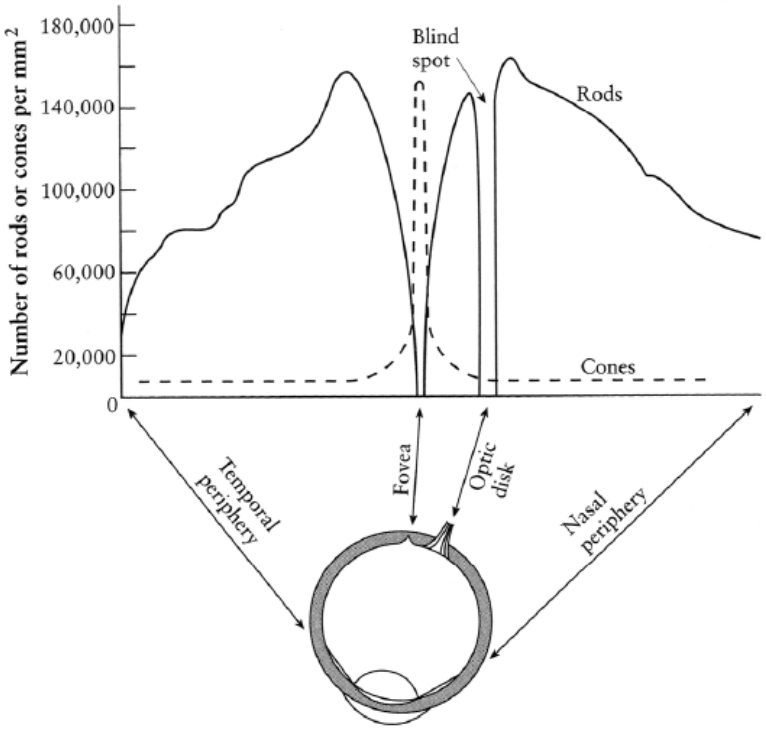
**= Any color can be specified by just three values, giving the weights of each of the three components**

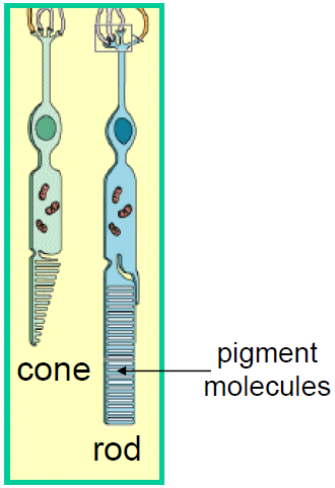
# Human Eye

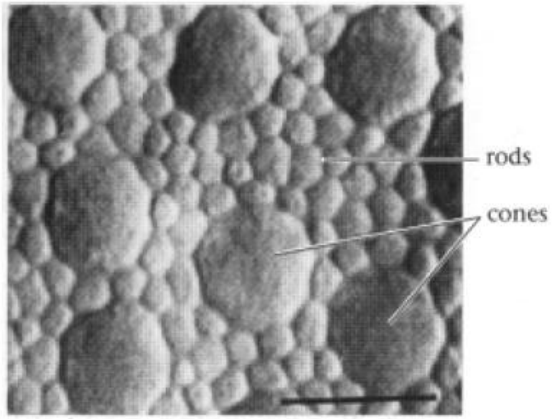

Conjunctiva
Iris
Cornea
Aqueous
Ciliary body
Zonule fibers
Lens
Fovea
Visual axis
Optic axis
Retina
Vitreous humor
Macula lutea
Disk
Optic nerve
Choroid
Sclera

# Rods and Cones

- Two types of cells on the retina: rods and cones
  - **Rods:** responsible for **intensity (125M)**
  - **Cones:** responsible for **color (6M~7M)**

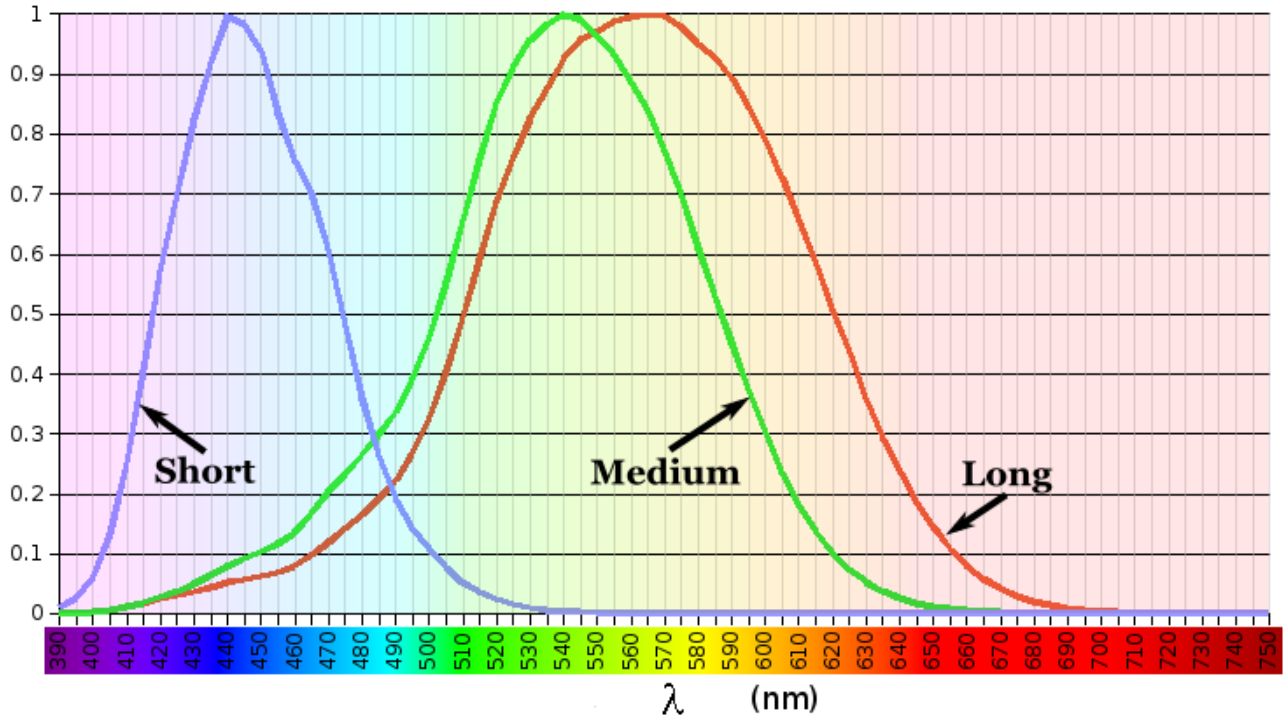


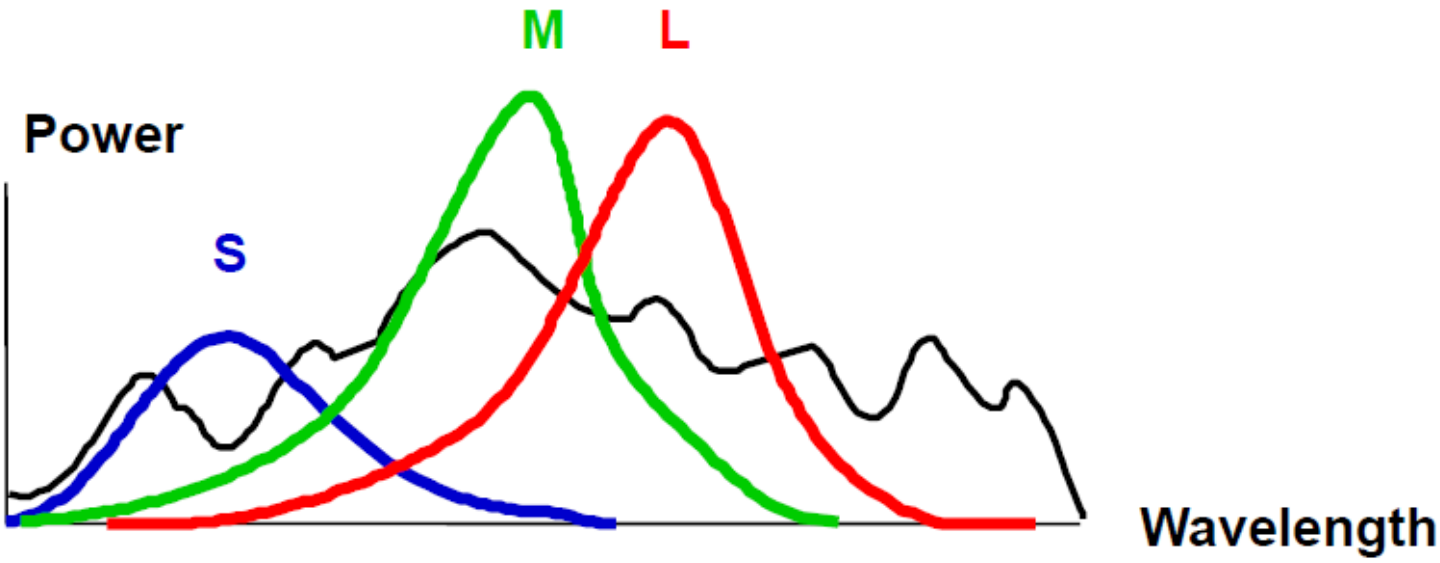
# Three Types of Cone Cells

- L-cones: 564 nm (Long)
- M-cones: 534 nm (Medium)
- S-cones: 420 nm (Short)



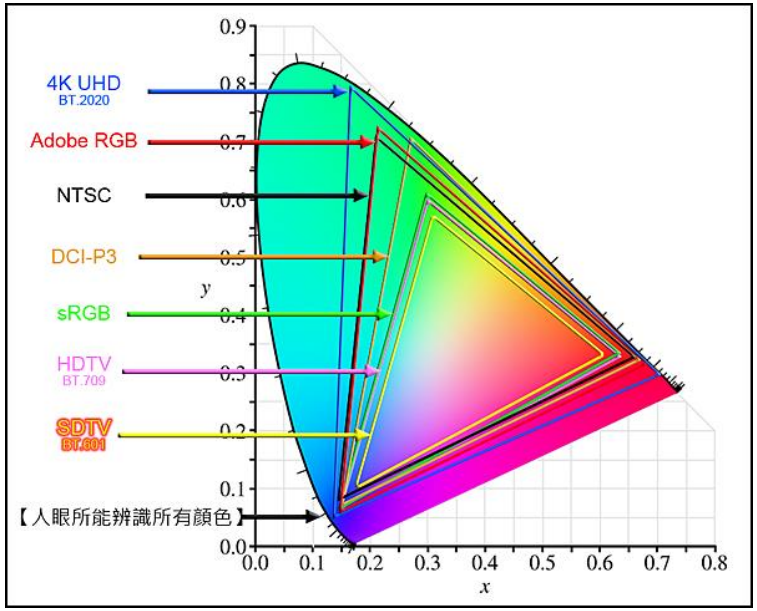
# Color Perception

- Rods and cones act as **filters** on the spectrum
  - To get the output of a filter, multiply its response curve by the spectrum, integrate over all wavelengths
  - Each cone yields one number and we just got three numbers in total!



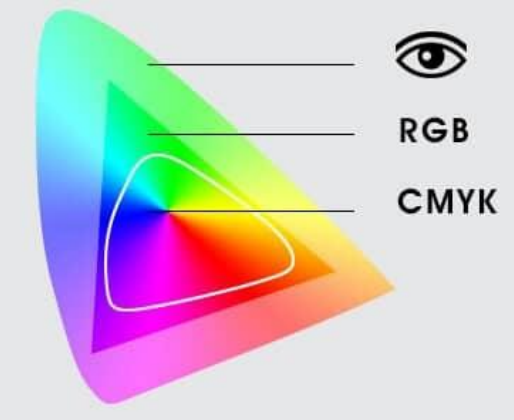
# RGB Color Model

- The **tristimulus theory** and the **response curves of LMS cones** lead to the RGB model
  - Any color can be represented by three values, giving the proportions of red (R), green (G), and blue (B) light
  - However, no standard SPDs are defined for R, G, and B



# RGB Color Gamut

- Although the RGB model provides a good representation of color, it cannot represent all visible colors of the human eye
- RGB primaries do produce the **largest** gamut from the simple addition of three primaries
- Red, green, and blue are called the **primary color** of the light (additive mixing)

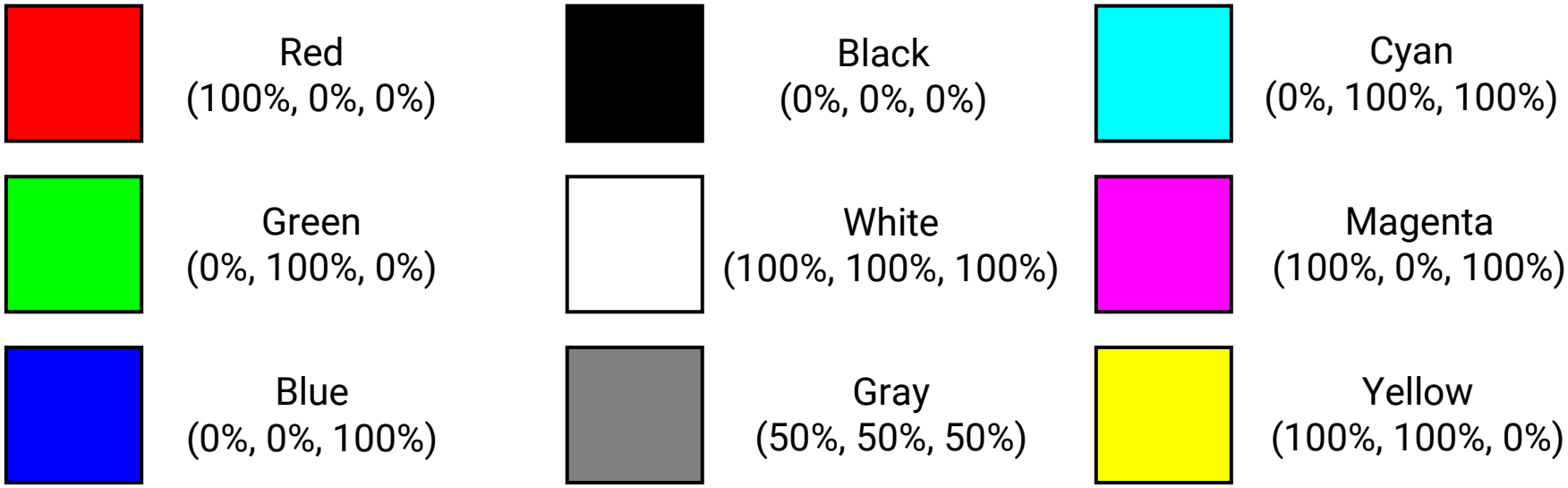


# RGB Color Model Representation

- We can write a color with the RGB model in the form of

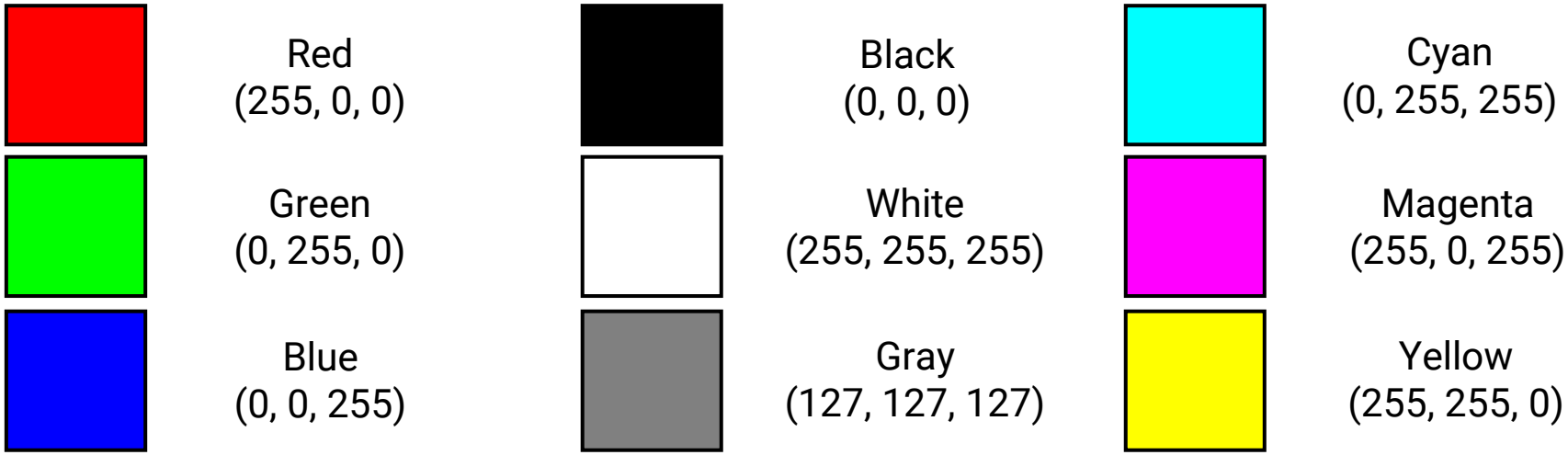
  **(r, g, b),**

Where r, g, b are the **amounts (proportion of the pure light)** of red, green, and blue light making up the color



# Color Depth

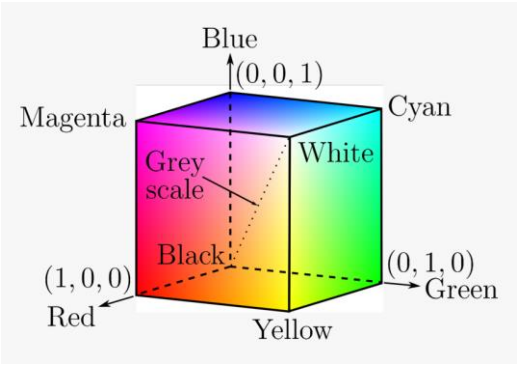
- In digital representation, we must choose the **number of bits** used for a color
- The most common choice is **8 bits (1 byte)** for each primary color, making 24 bits (3 bytes) in total
  - The range of value falls within [0, 255], making a total 256 x 256 x 256 = 16777216 different colors **(24 bit color depth)**



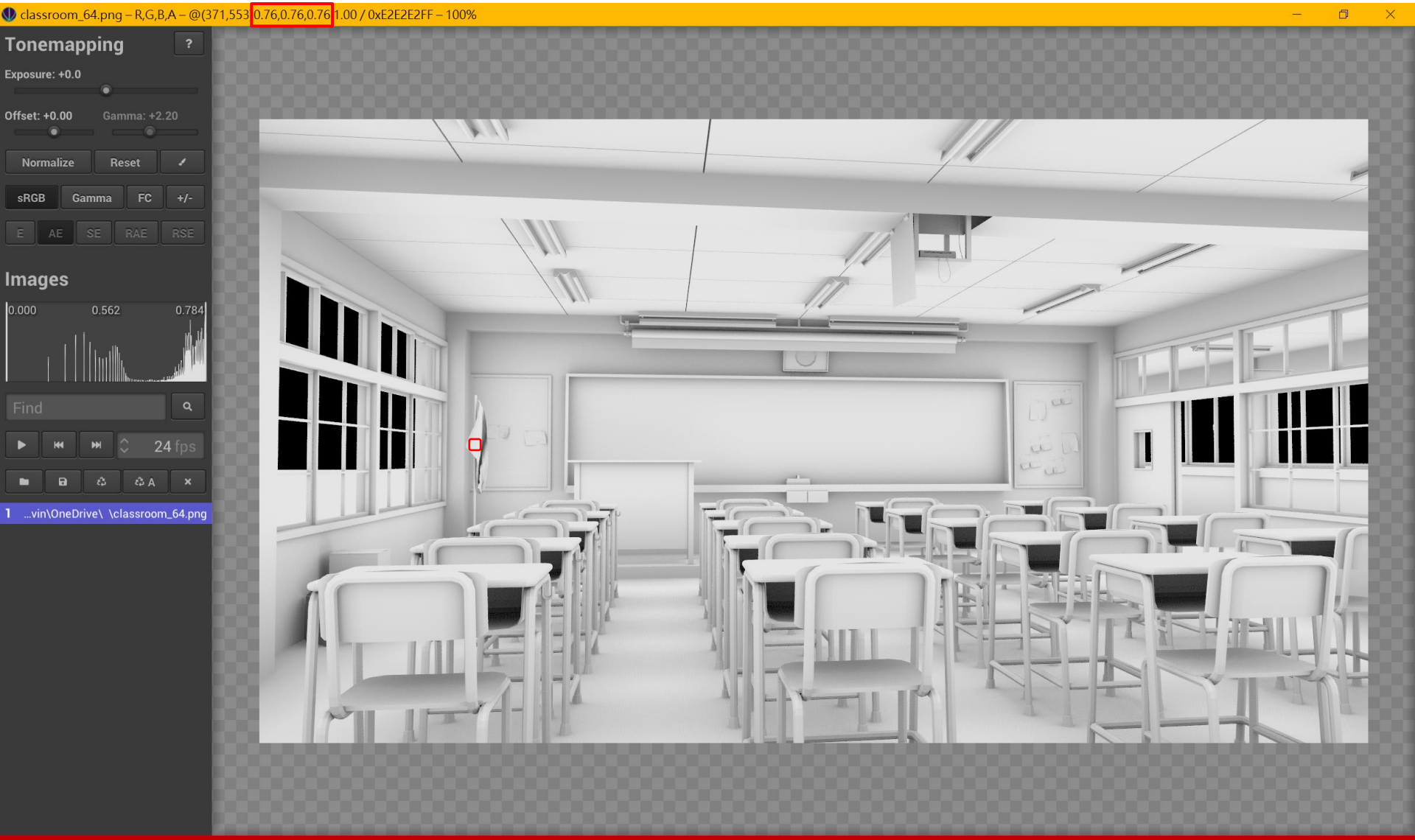
# Color Representation

- In **interactive** computer graphics, we usually normalize the range of color to [0.0, 1.0]

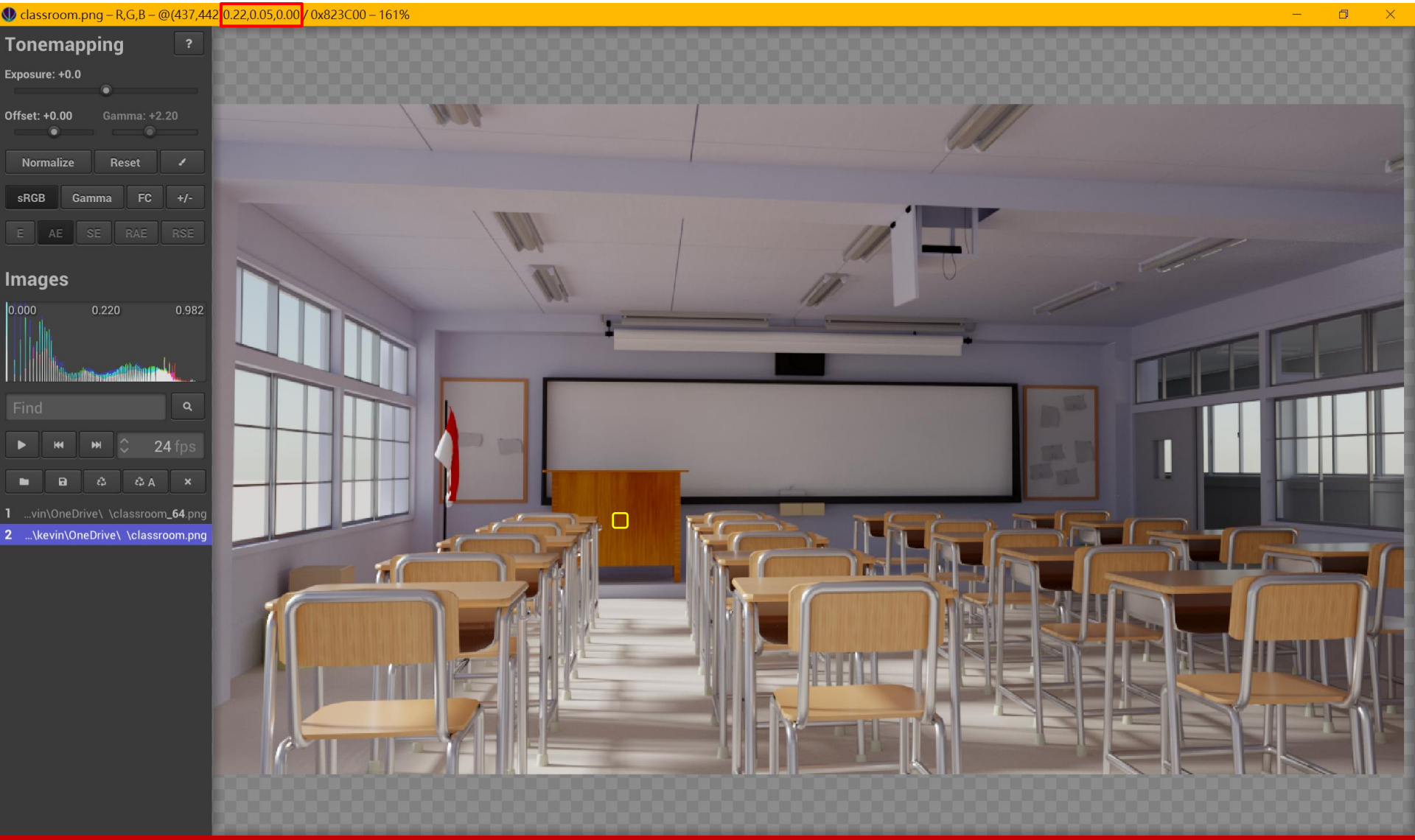
| | | |
|---|---|---|
| Red (1, 0, 0) | Black (0, 0, 0) | Cyan (0, 1, 1) |
| Green (0, 1, 0) | White (1, 1, 1) | Magenta (1, 0, 1) |
| Blue (0, 0, 1) | Gray (0.5, 0.5, 0.5) | Yellow (1, 1, 0) |



# The Rendered Images (Gray Scale)



# The Rendered Images (Color)



# Any Questions?