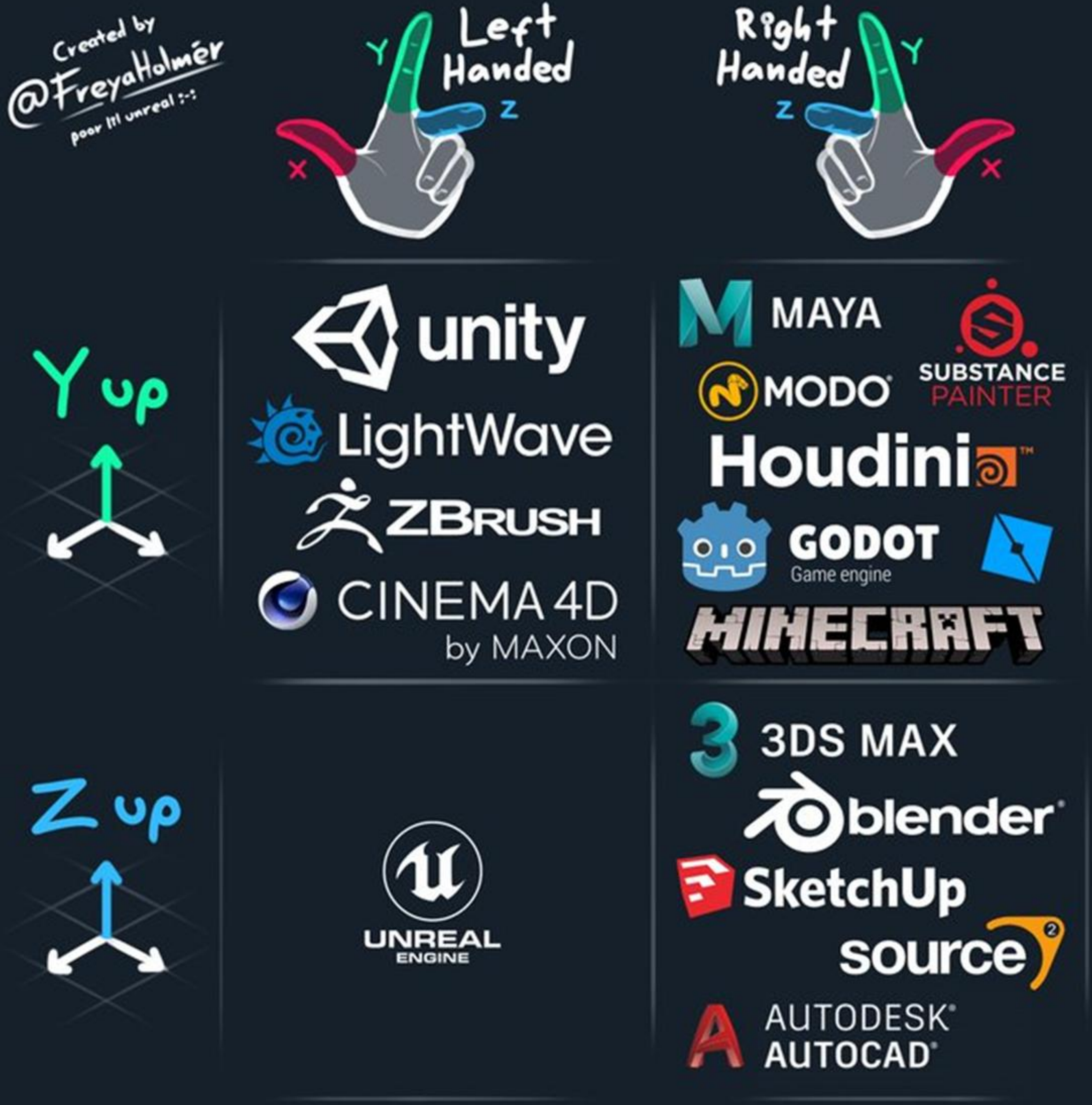
Linear Algebra, Optics and Color Spaces

Computer Graphics
Fall Semester 2025

S. Felix

University of Applied Sciences and Arts Northwestern Switzerland
School of Computer Science


Created by
@FreyaHolmér
poor lil unreal :-:
Left Handed
Right Handed
Y
Z
X
Y up
Z up
unity
LightWave
ZBrush
CINEMA 4D
by MAXON
MAYA
SUBSTANCE PAINTER
MODO
Houdini
GODOT
Game engine
MINECRAFT
UNREAL ENGINE
3DS MAX
blender
SketchUp
source
AUTODESK
AUTOCAD

Light, simplified

Light sources emit photons

Photons move in **straight lines**,
until they hit something

Each photon is monochromatic
(of a single **color**/wavelength)

We see photons when
they reach our eyes

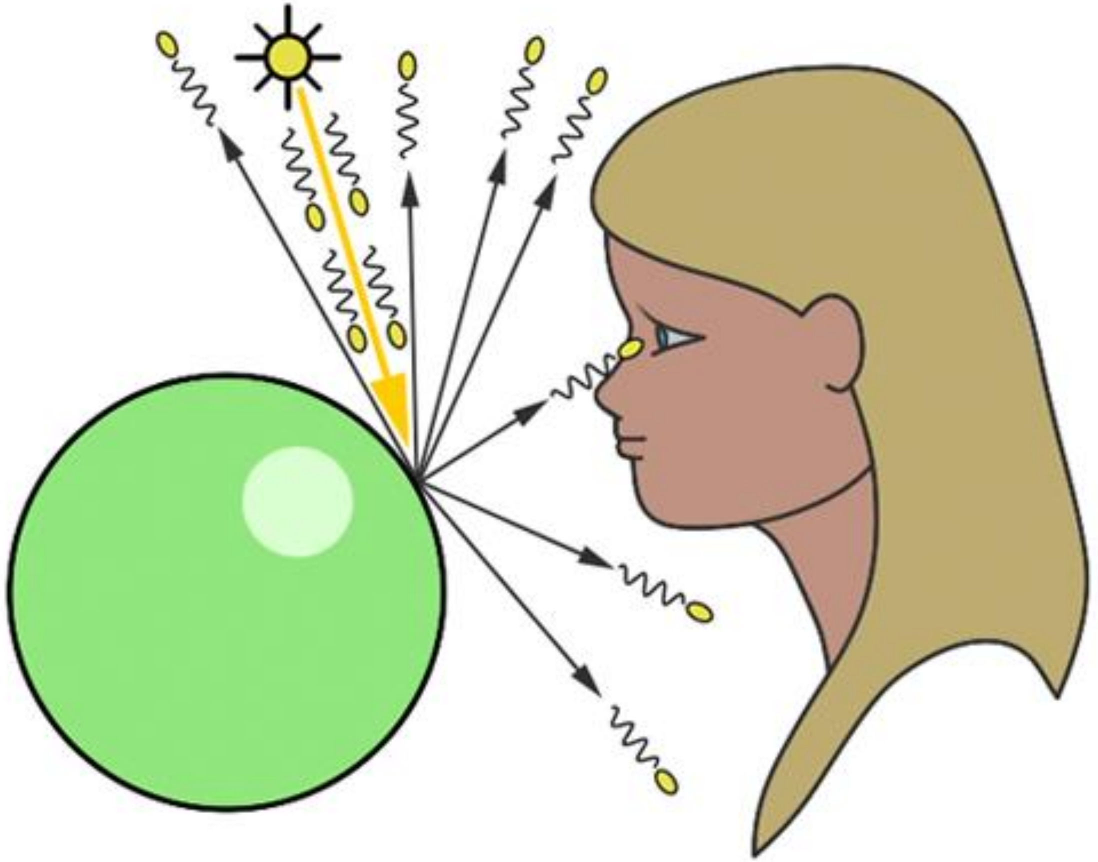
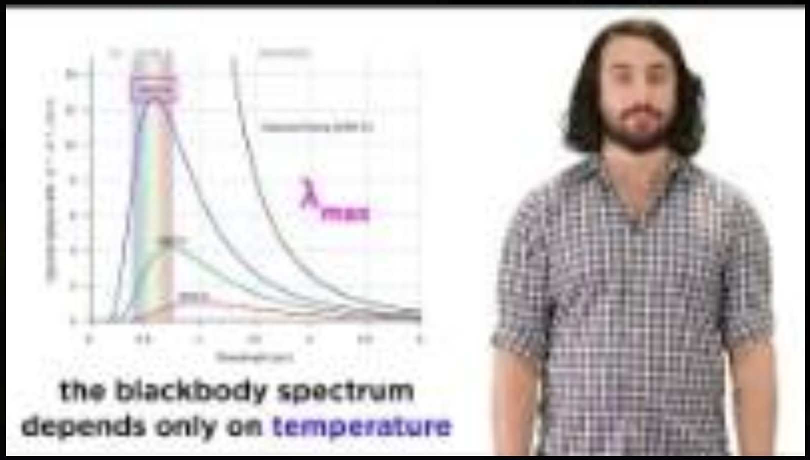


Image: Scratchapixel

Light, not simplified

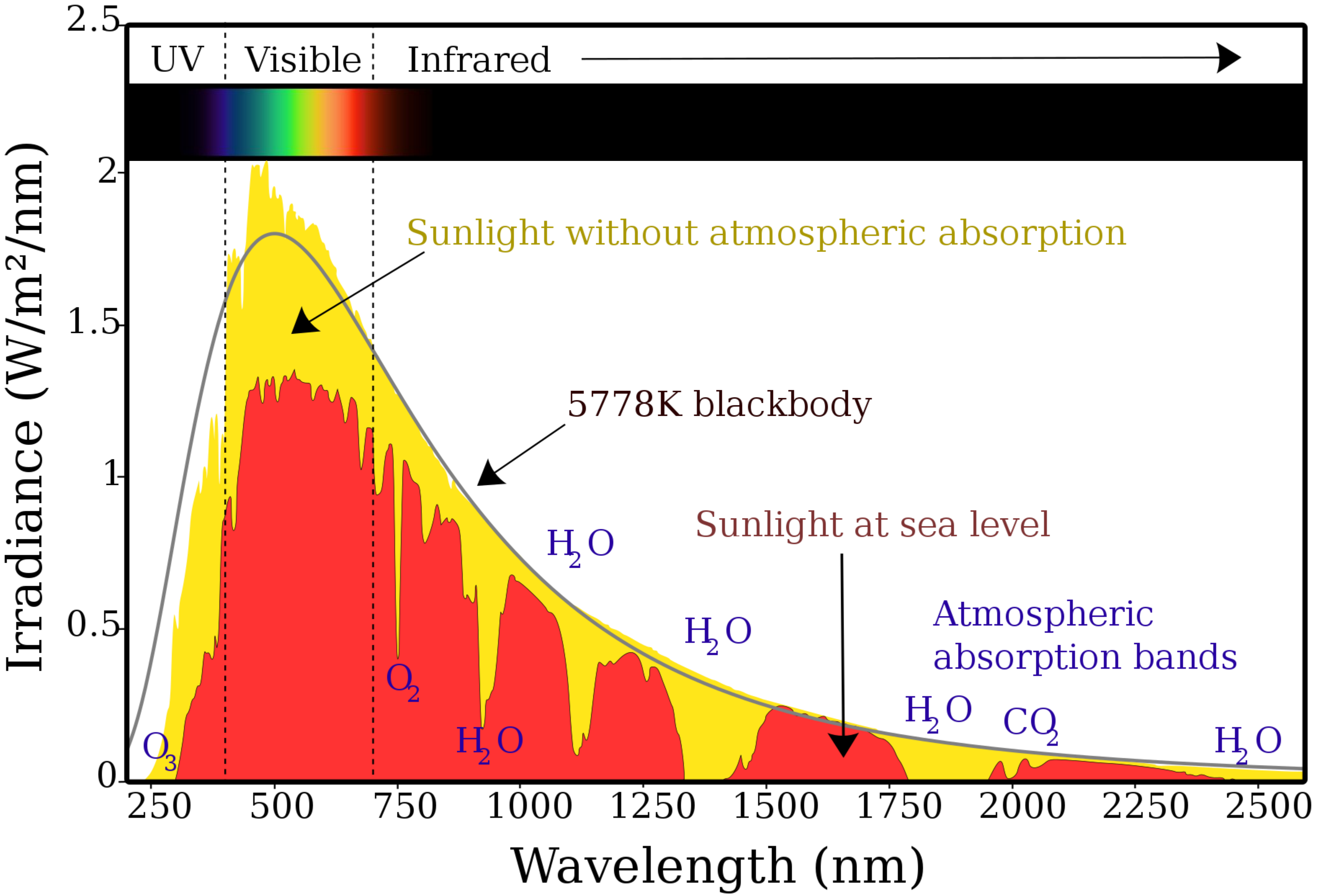
https://www.youtube.com/watch?v=QCX62YJCmGk



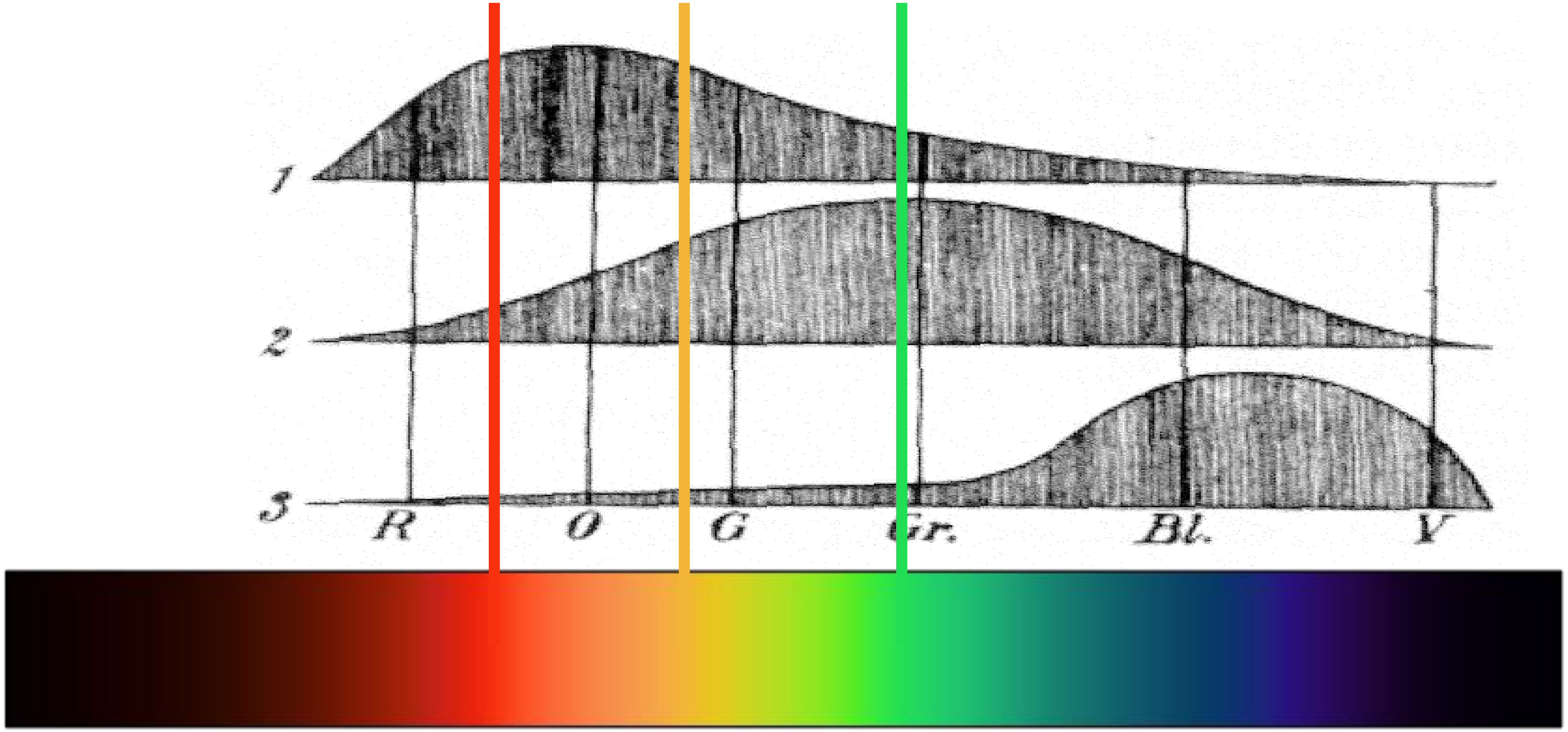
https://www.youtube.com/watch?v=7BXvc9W97iU

https://www.youtube.com/watch?v=Iuv6hY6zsd0

Spectrum of Solar Radiation (Earth)



Eyes contain four different kinds of photoreceptors, **three** of which are sensitive to colors.



	1	2	3
	0.7	0.6	0.1

	1	2	3
	0.9	0.2	0.1

	1	2	3
	0.5	1.0	0.1

Image: Young & Helmholz

Color Spectrum of an LED light bulb

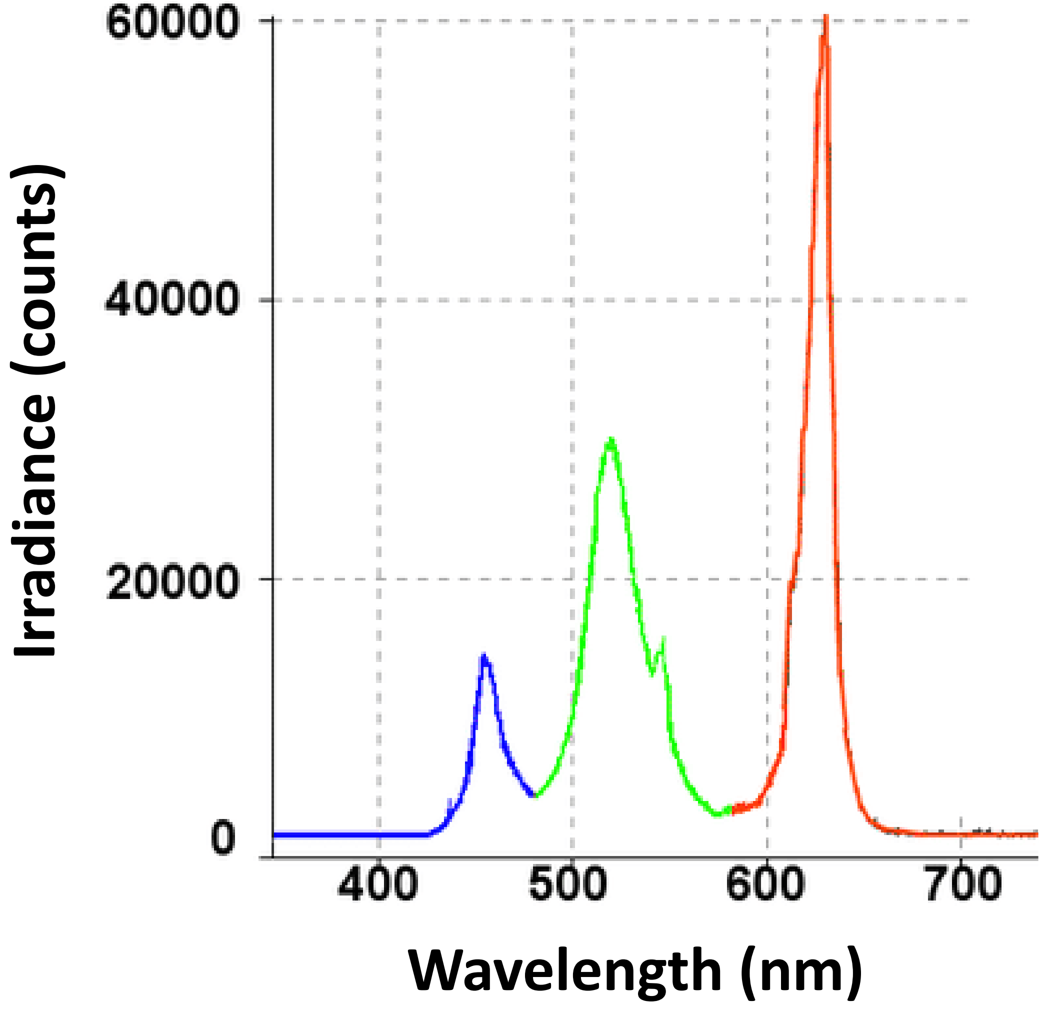
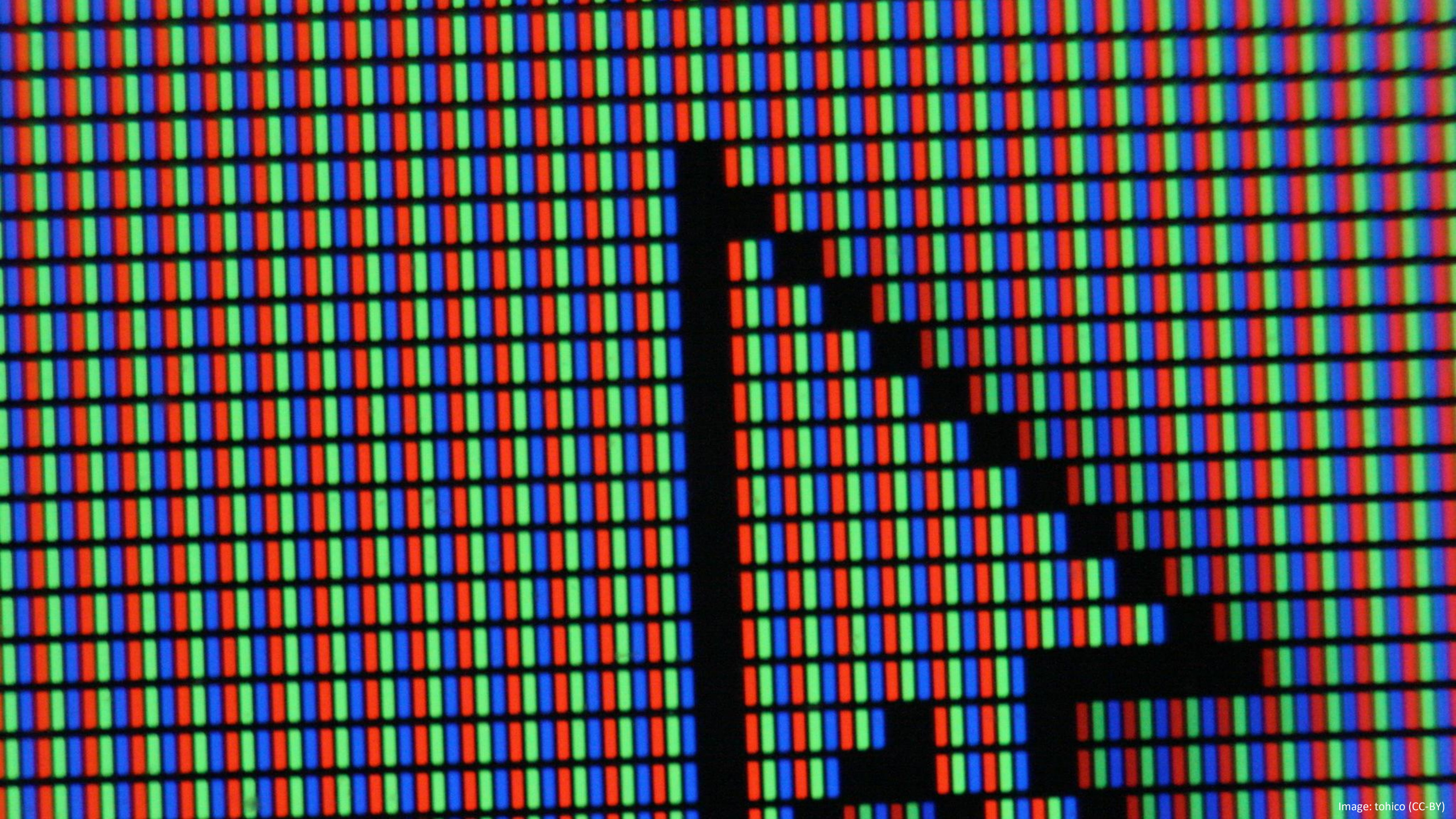


Image: Hejduk, S., Witas, K., Latal, J., Vitasek, J. and Bocheza, J., 2014, July. Use of the RGBY model for indoor mobile visible light communication. In *2014 16th International Conference on Transparent Optical Networks (ICTON)* (pp. 1-4). IEEE.



Representing Colors

Colors can be represented as points or **vector** in a three-dimensional **color space**

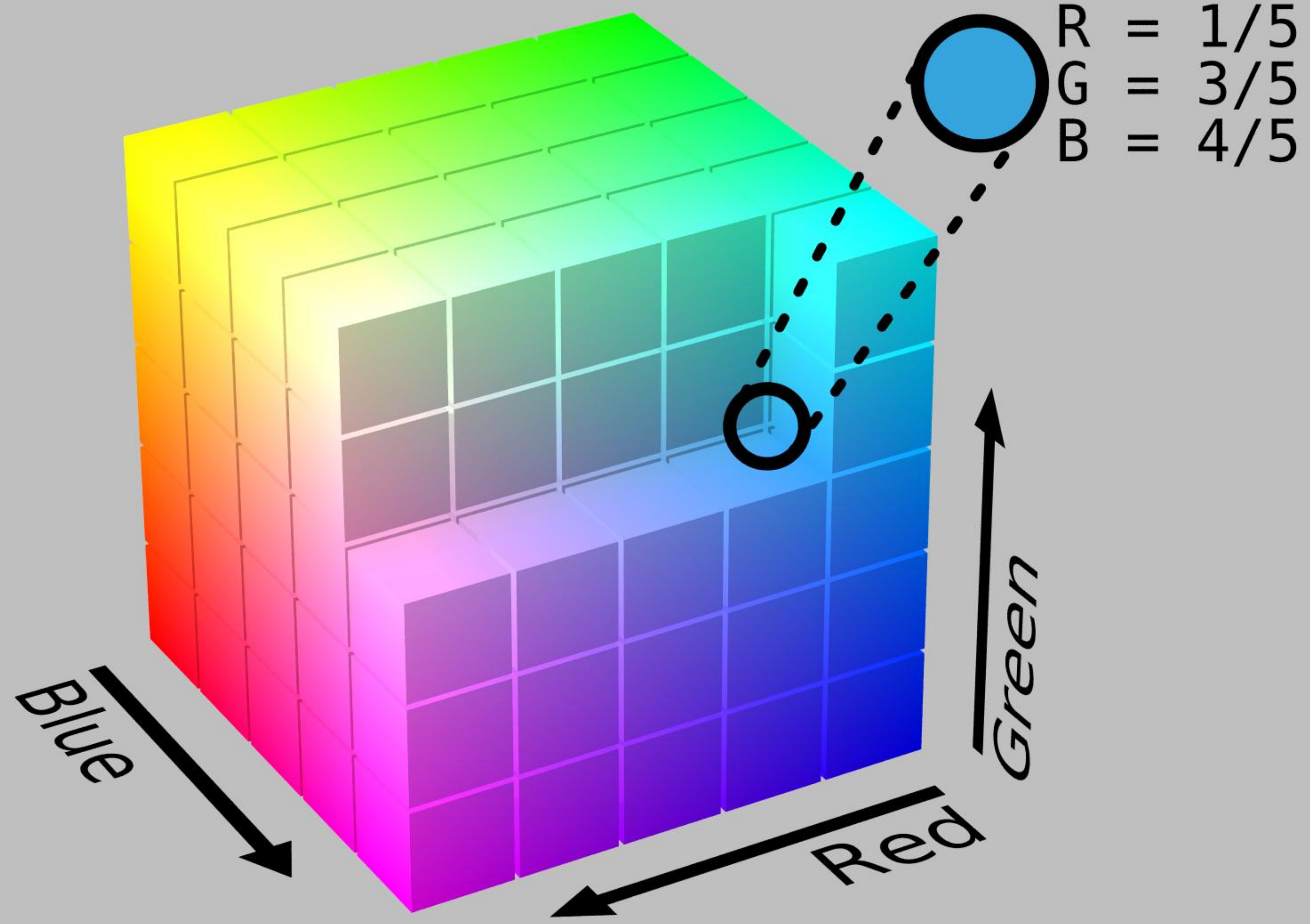
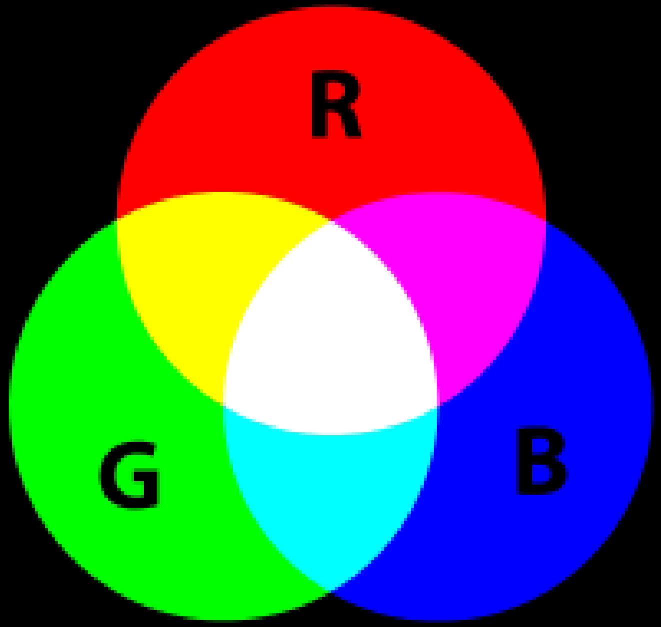
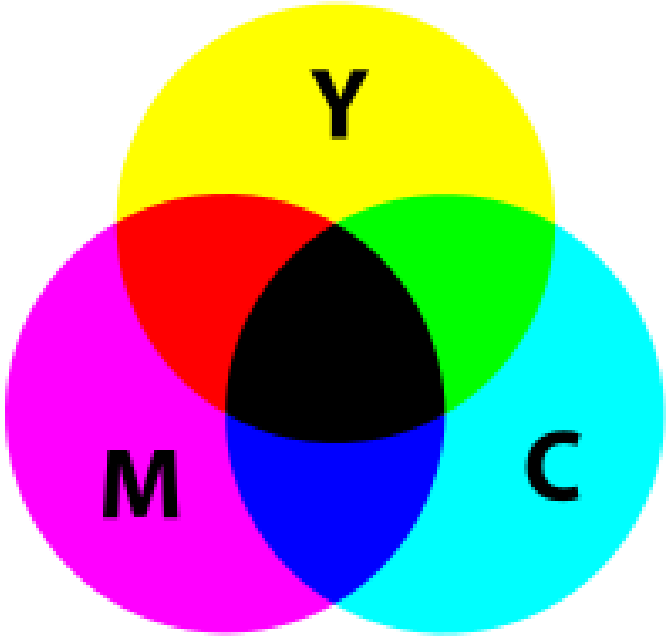


Image: Horst Frank

Additive Color Space



Subtractive Color Space



Color Spaces

Different color spaces are suited for **different applications**.

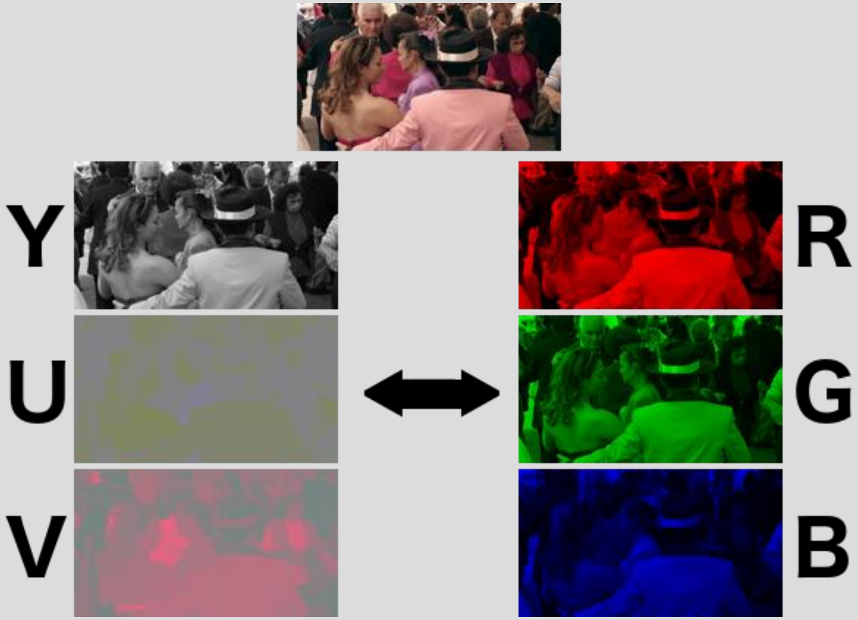


Image: Ronald S. Bultje

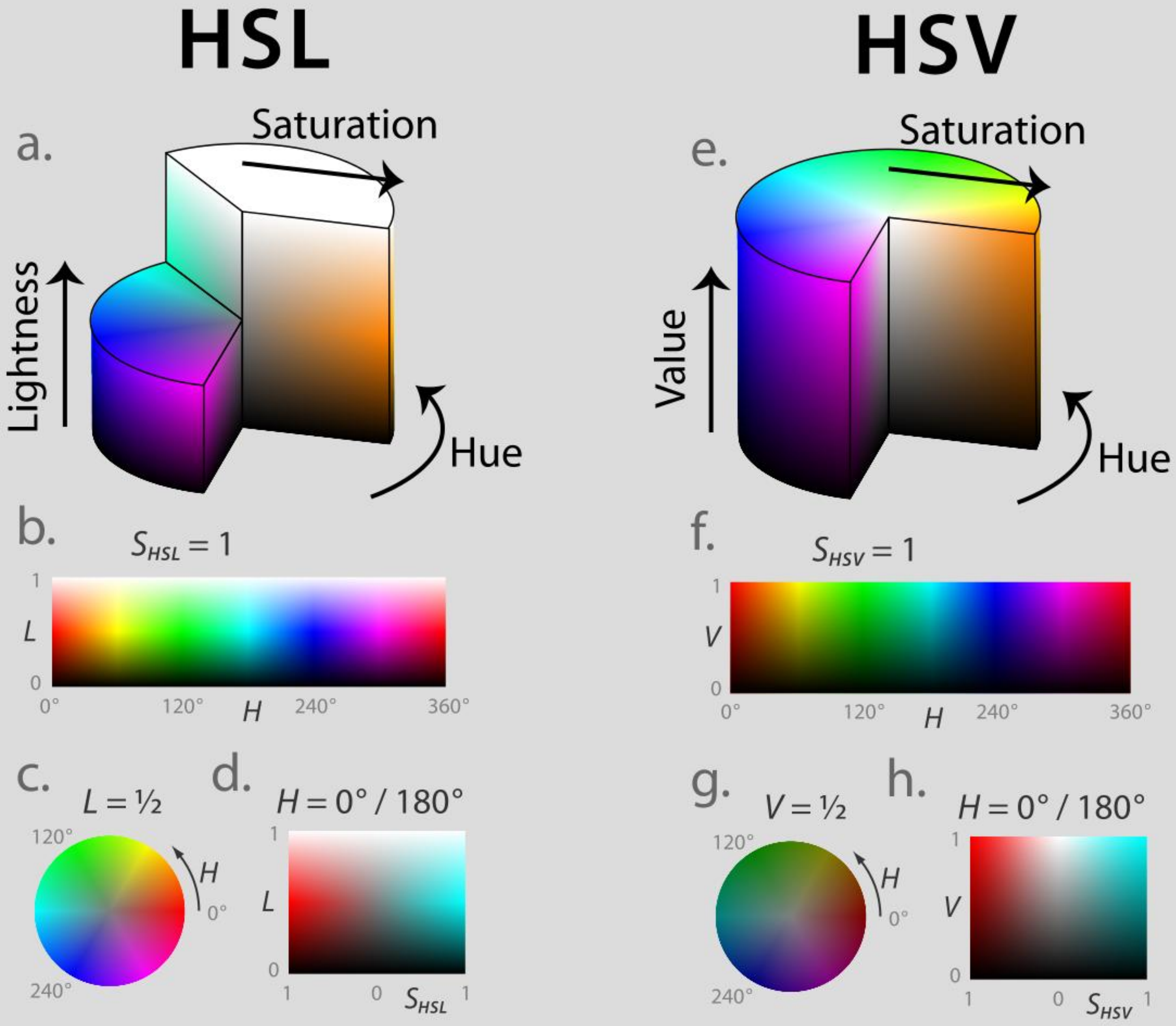


Image: Jacob Rus

$$Rose \coloneqq \begin{bmatrix} 0.867 \\ 0.356 \\ 0.047 \end{bmatrix} = \begin{bmatrix} 86.7\% \\ 35.6\% \\ 4.7\% \end{bmatrix}$$

Perform **computations** in **Linear RGB**.

$0 \mathrel{\widehat{=}} No\ red$	$0 \mathrel{\widehat{=}} No\ green$	$0 \mathrel{\widehat{=}} No\ blue$
$\vdots$	$\vdots$	$\vdots$
$1 \mathrel{\widehat{=}} Pure\ red$	$1 \mathrel{\widehat{=}} Full\ green$	$1 \mathrel{\widehat{=}} Pure\ blue$
$\vdots$	$\vdots$	$\vdots$
$\infty \mathrel{\widehat{=}} Infinitely\ red$	$\infty \mathrel{\widehat{=}} Infinitely\ green$	$\infty \mathrel{\widehat{=}} Infinitely\ blue$

Display colors in **sRGB** with 8-bits per component.

$$0 \triangleq No\ red \quad \vdots \quad 255 \triangleq Full\ red$$

$$0 \triangleq No\ green \quad \vdots \quad 255 \triangleq Full\ green$$

$$0 \triangleq No\ blue \quad \vdots \quad 255 \triangleq Full\ blue$$

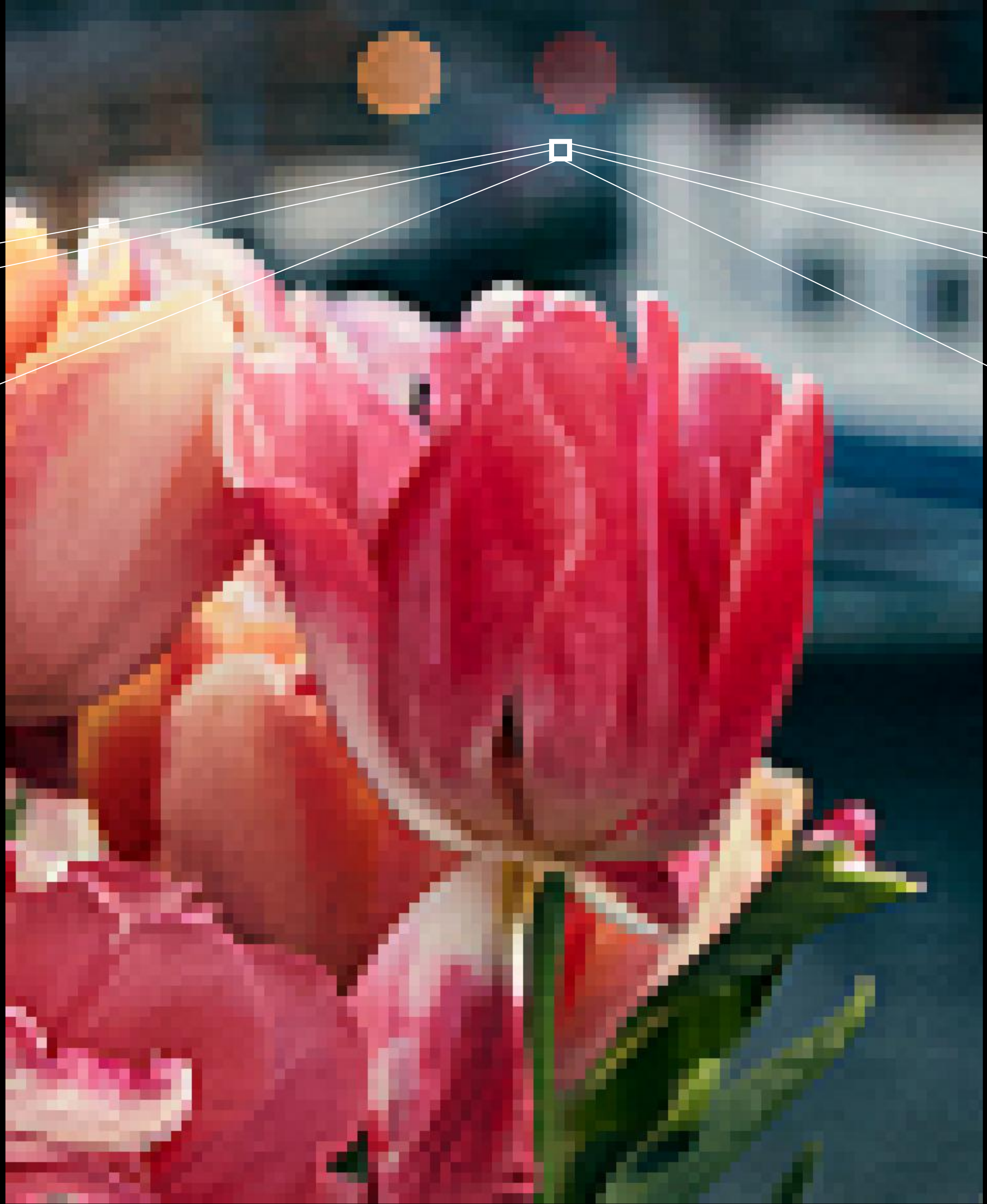
Linear RGB

0.072 Red
0.165 Green
0.238 Blue

Used for **computations,**
linear operations work.

High **accuracy.**

Unlimited dynamic range.



sRGB

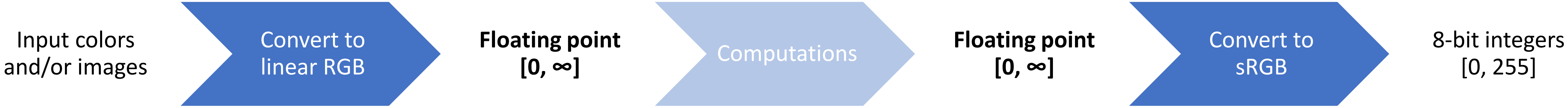
76 Red
113 Green
134 Blue

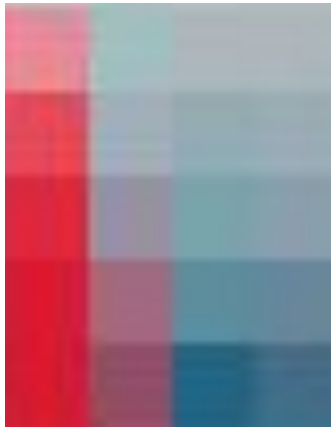
Used for **output,**
not useful for computations.

Compact representation.

Limited dynamic range.

Color Spaces in Image Processing Pipelines





RGBA32 is a common format to represent each pixel with four bytes.

BGRA32 is an alternative, with R and B swapped.

0x00958BE8	0x00C1BEA3	0x00BCB6AD	0x00BAB8AB	(padding)
0x005B49E7	0x00B8AFA0	0x00B3AC93	0x00B5AC9A	(padding)
0x003E28DD	0x00A39297	0x00ACA37B	0x00AB9F89	(padding)
0x003219DA	0x007E6AA2	0x009F8D5E	0x009B8769	(padding)
0x00331AD5	0x00644E92	0x00836621	0x00866D45	(padding)

Stride

```
int[] memory;

 0x00000000 =       (red)
            + (green << 8)
            + (blue << 16)
```

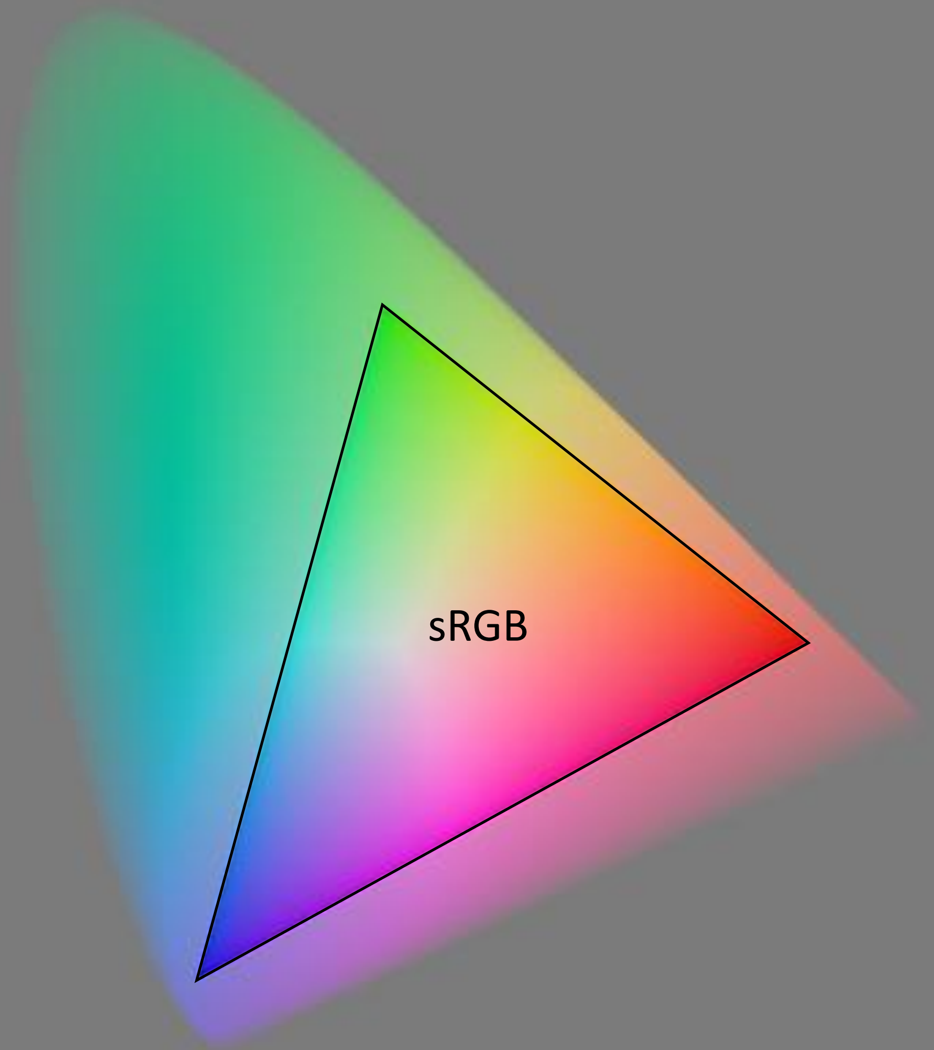
Color Gamut

Which green is “100% green”?

$$\mathit{Green} := \begin{bmatrix} 0.0 \\ 1.0 \\ 0.0 \end{bmatrix} = \begin{bmatrix} 0\% \\ 100\% \\ 0\% \end{bmatrix}$$

Wide color gamut

Narrow color gamut



Color Gamut

Wide Gamut

Narrow Gamut

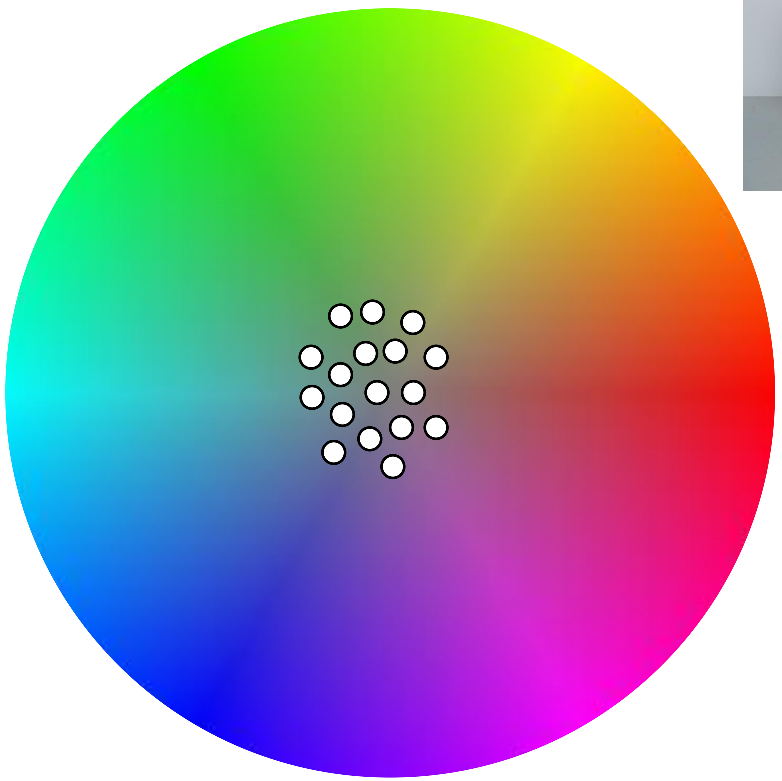
Image: Imaging Resource

Banding

Only a **limited number** of different colors can be represented with 8 bits.

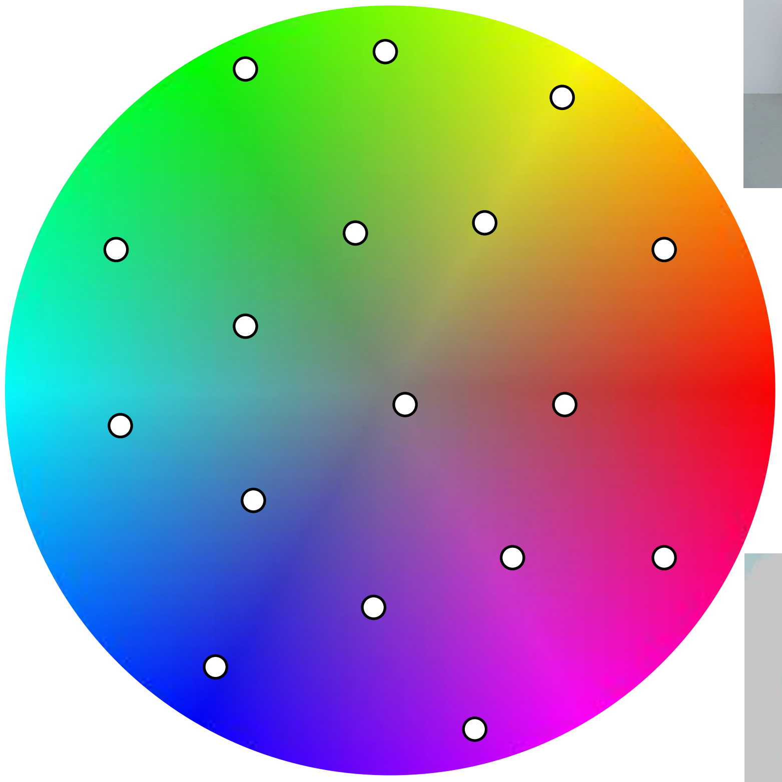
To cover a wide color gamut, representable colors are spread out further.

Thus, wide gamut color spaces have bigger gaps between colors (a.k.a. **banding**)



Banding

Only a **limited number** of different colors can be represented with 8 bits.



To cover a wide color gamut, representable colors are spread out further.

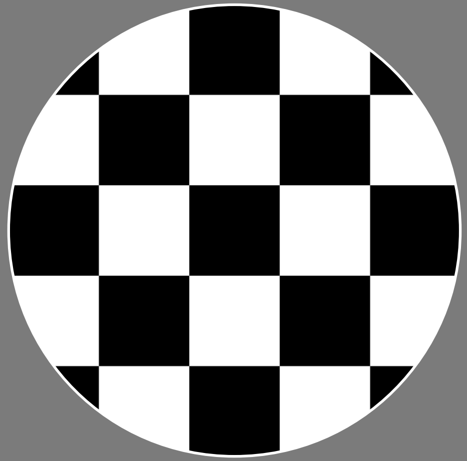
Thus, wide gamut color spaces have bigger gaps between colors (a.k.a. **banding**)


Pixelmator Pro
File
Edit
Insert
Image
Tools
Format
Arrange
View
Window
Help
Wed Dec 21 9:41 AM
Liam
Edited
Deband Split Comparison
Done
DEC
21

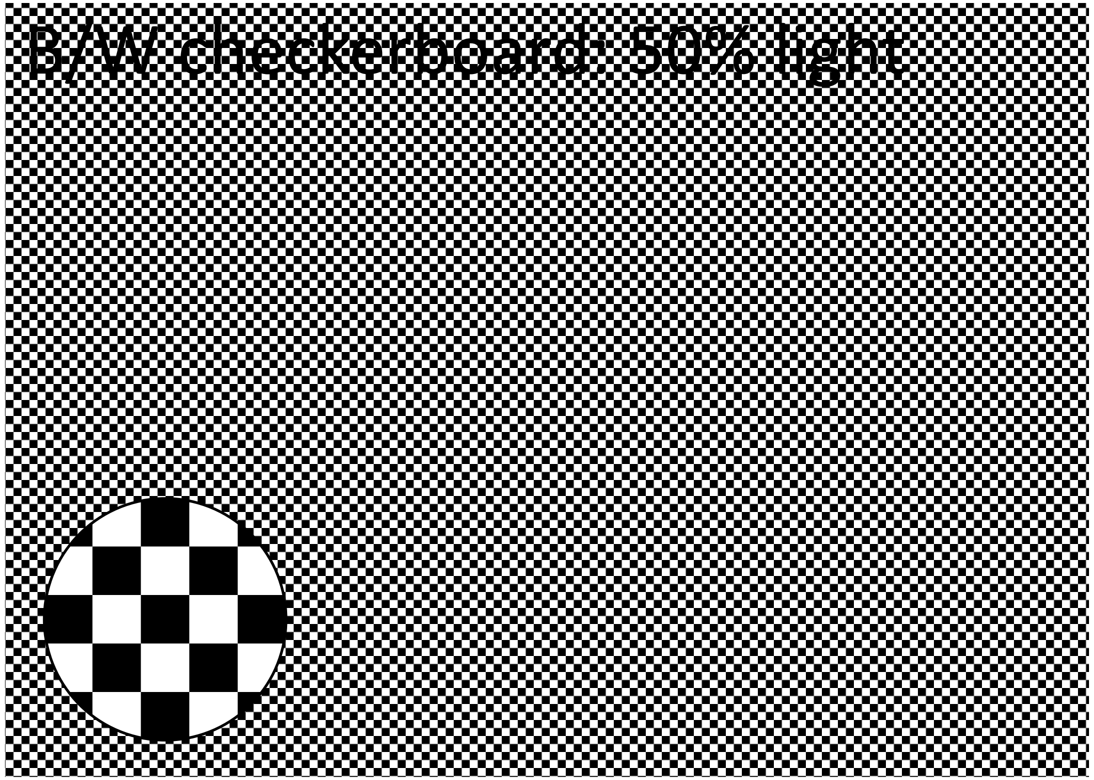
Brightness

How bright is “50% gray”?

$$Gray := \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 50\% \\ 50\% \\ 50\% \end{bmatrix}$$



Brightness Perception



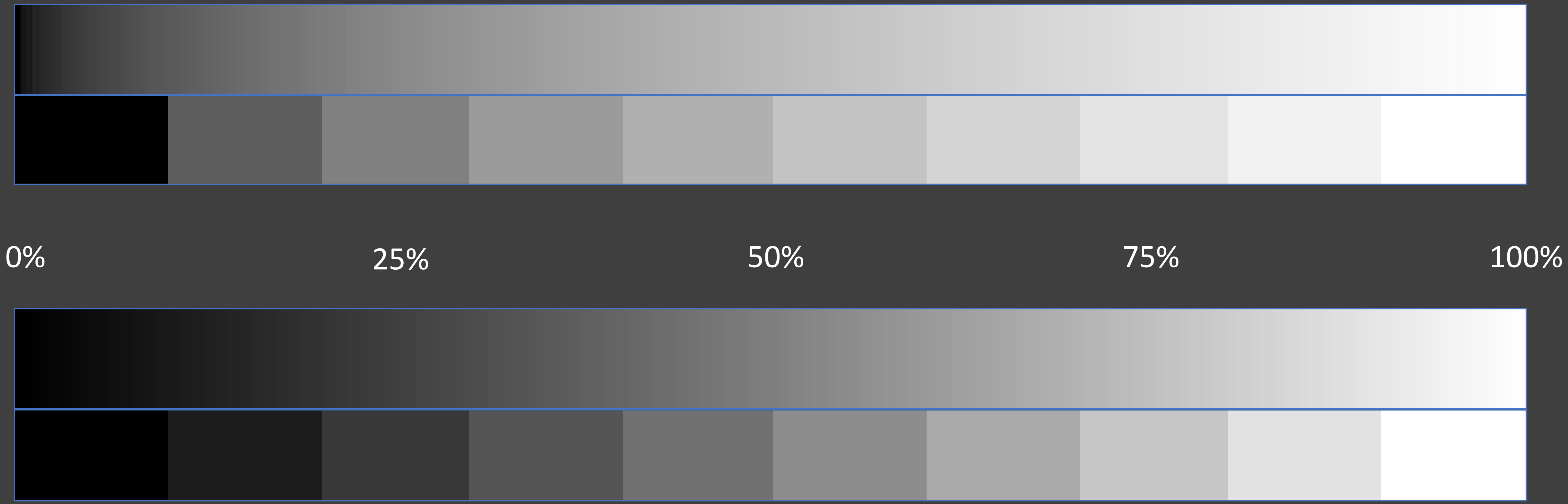
sRGB: 187/187/187

sRGB: 128/128/128

sRGB 128 is not half bright!

Brightness Perception

Linear RGB (physical, computations)



sRGB (perceptually linear, input and output)

Brightness Perception – Gamma Correction

Linear RGB (physical, computations)

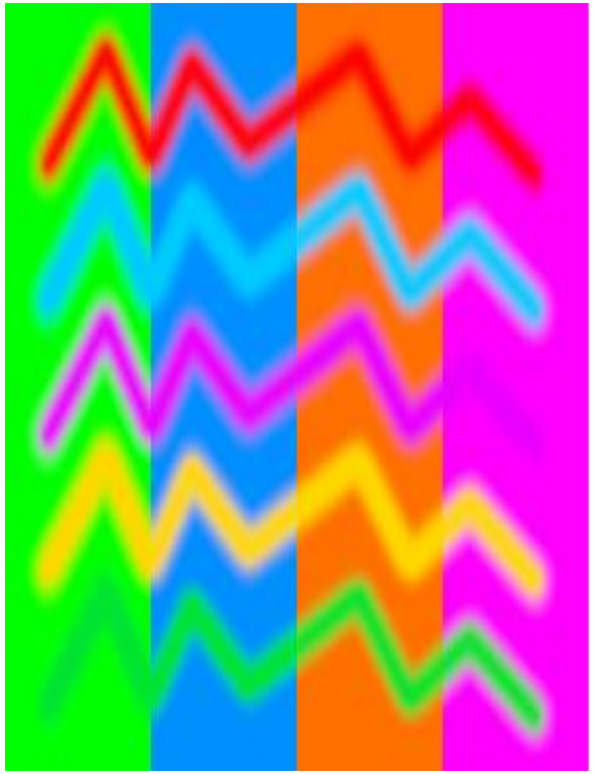
$$Linear = sRGB^{\gamma}$$

$$sRGB = Linear^{\frac{1}{\gamma}}$$

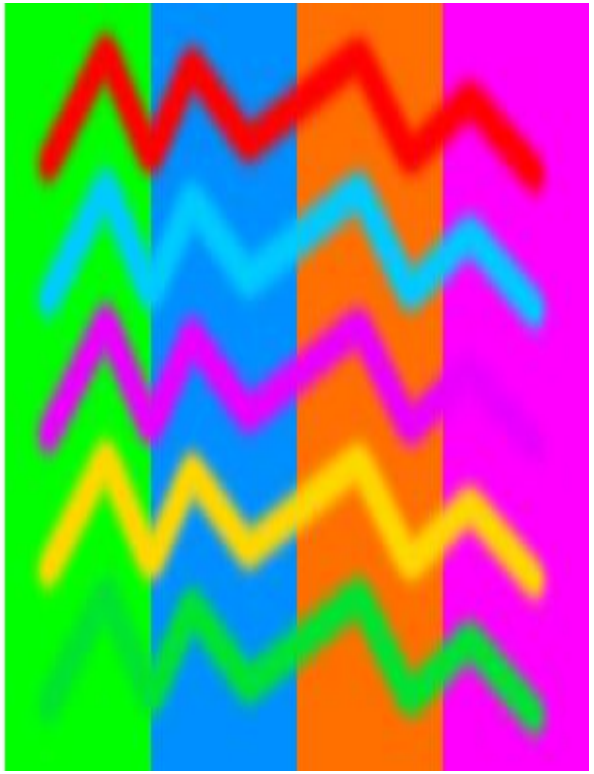
sRGB (perceptually linear, input and output)

$\gamma = 2.2$ (typ.)

Linear vs. Non-Linear Color Spaces



Computation in linear RGB

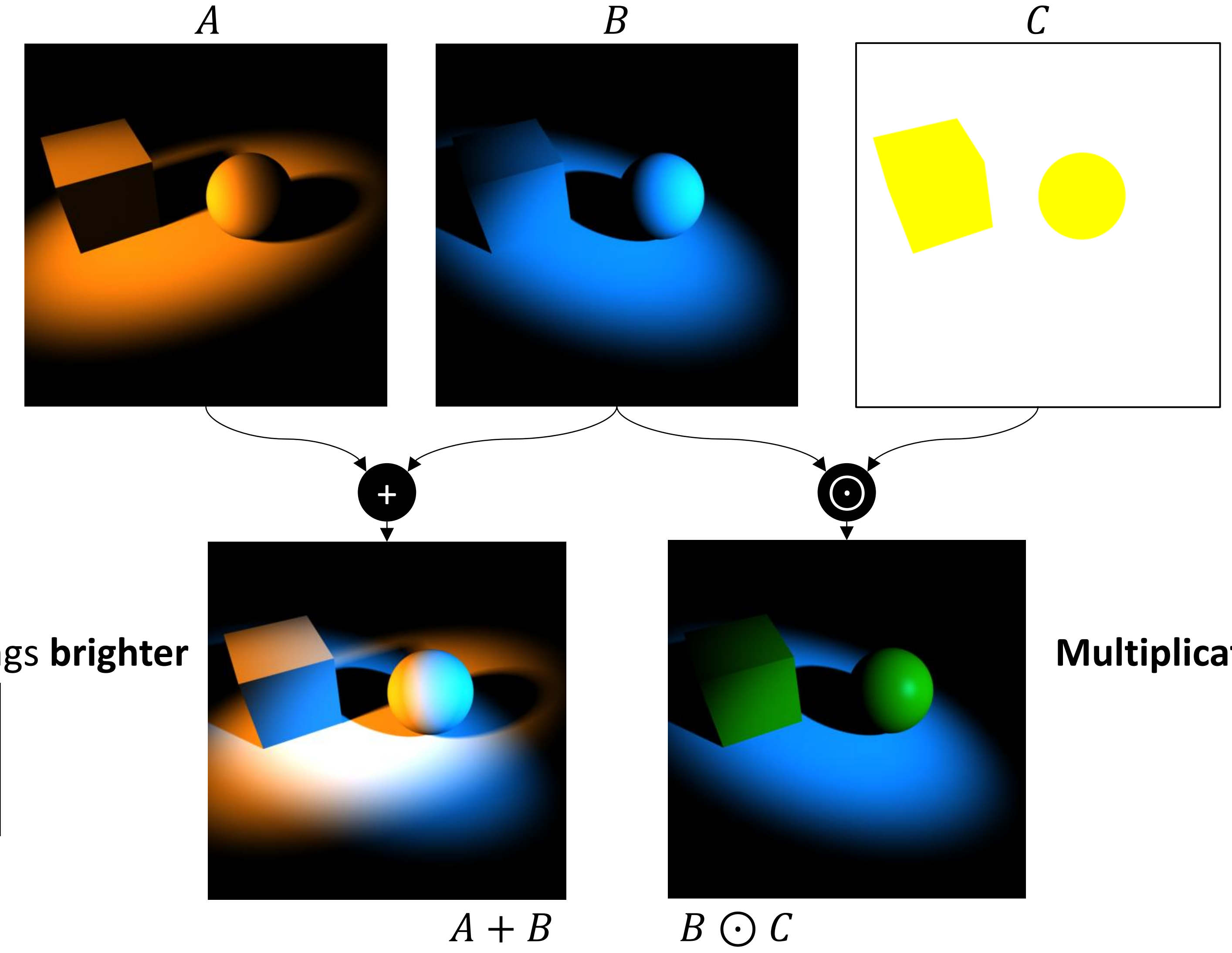


Computation in sRGB

Computations in non-linear color spaces (e.g. sRGB) produces **incorrect results**.

Image: John Novak

The Two Operations with Light



Addition makes things **brighter**

$$\begin{bmatrix} A_r + B_r \\ A_g + B_g \\ A_b + B_b \end{bmatrix}$$

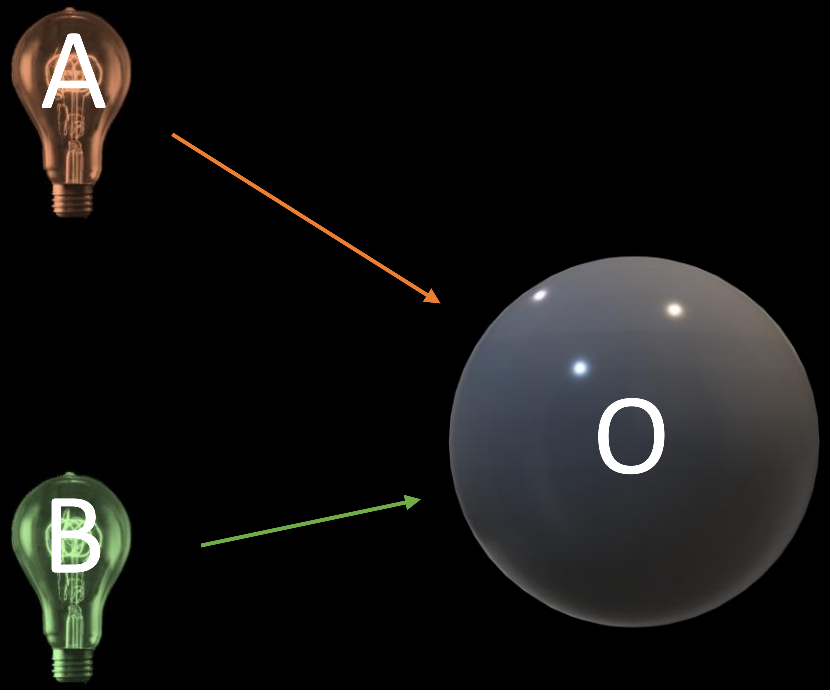
Multiplication makes things **darker**

$$\begin{bmatrix} B_r \cdot C_r \\ B_g \cdot C_g \\ B_b \cdot C_b \end{bmatrix}$$

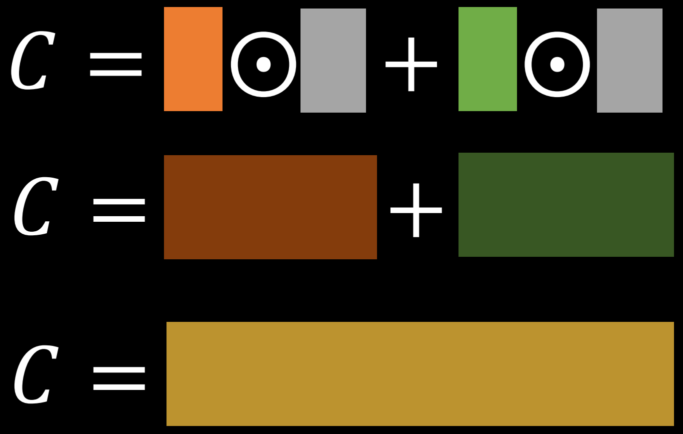
The effects of two light sources add up

Yellow surfaces (C) illuminated by a blue light (B)

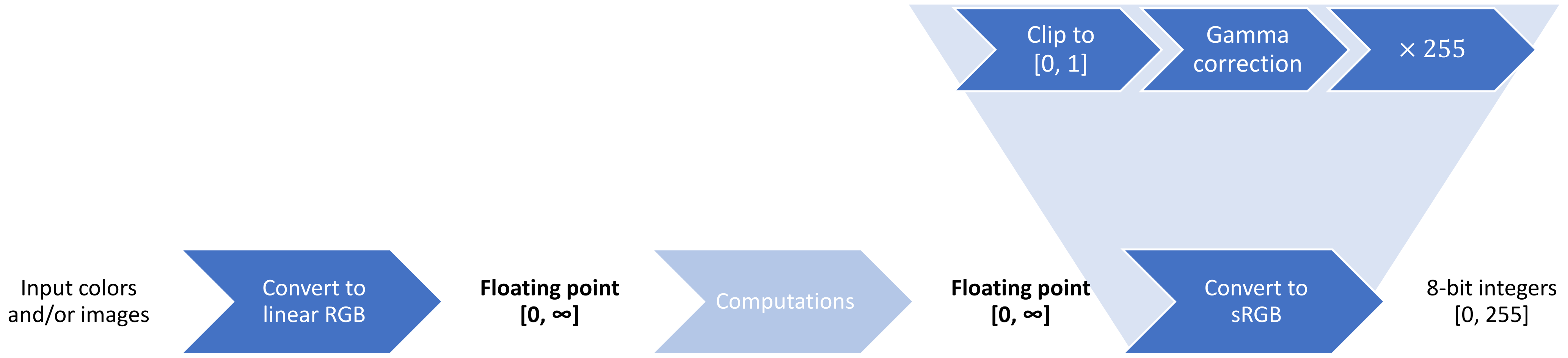
The Two Operations with Light



$$C = A \odot O + B \odot O$$



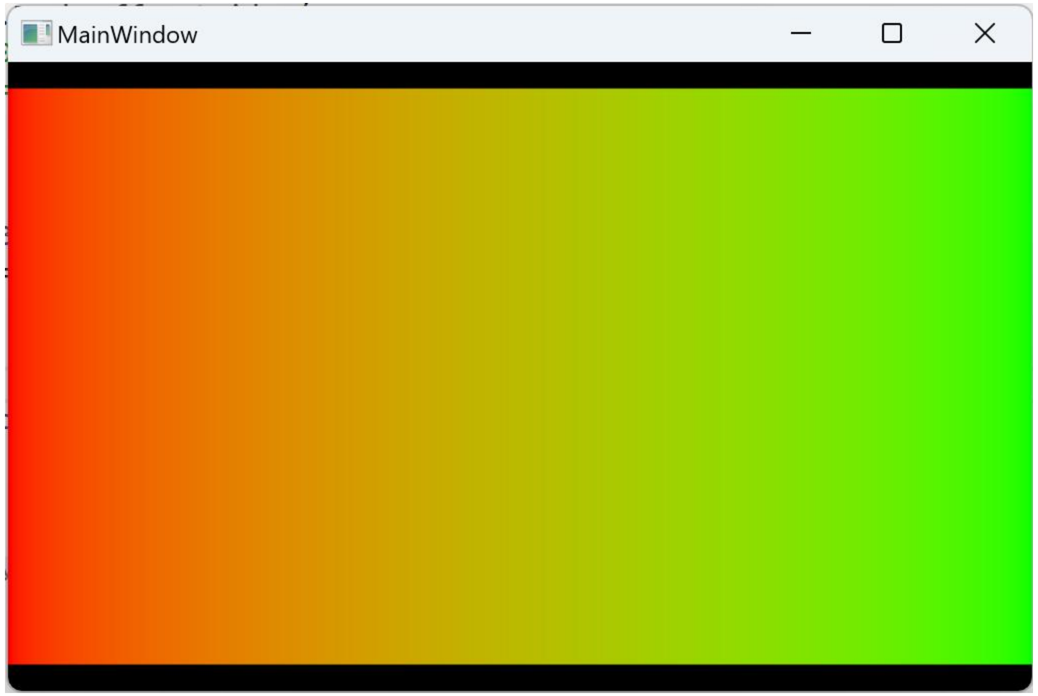
Color Spaces in an Image Processing Pipeline



This Week's Lab

Create a skeleton in C#, Java or C/C++. It should

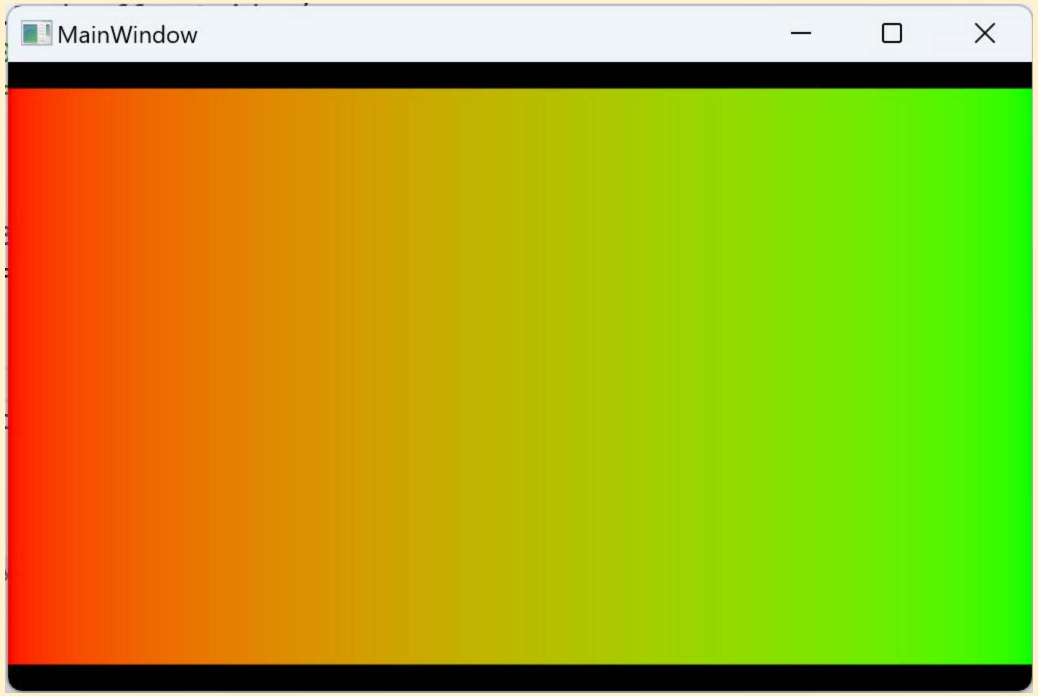
- Perform color interpolation in **Linear RGB**
- Convert the result to sRGB
  1. Clip to $[0 \ldots 1]$
  2. Gamma correction
  3. $\times\, 255$
- Draw pixels in sRGB



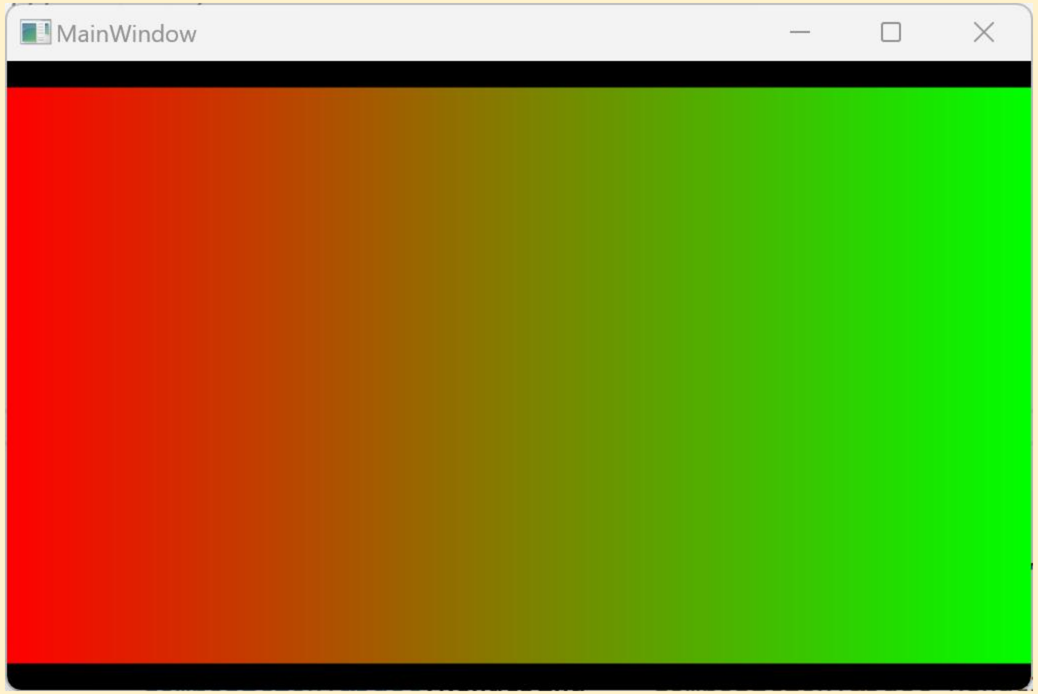
Take a screenshot of your results

Incorrect Results

Correct



Incorrect



Suggestions

C#

- Use `System.Numerics.Vector3` to represent colors
- Use WPF/`WriteableBitmap`

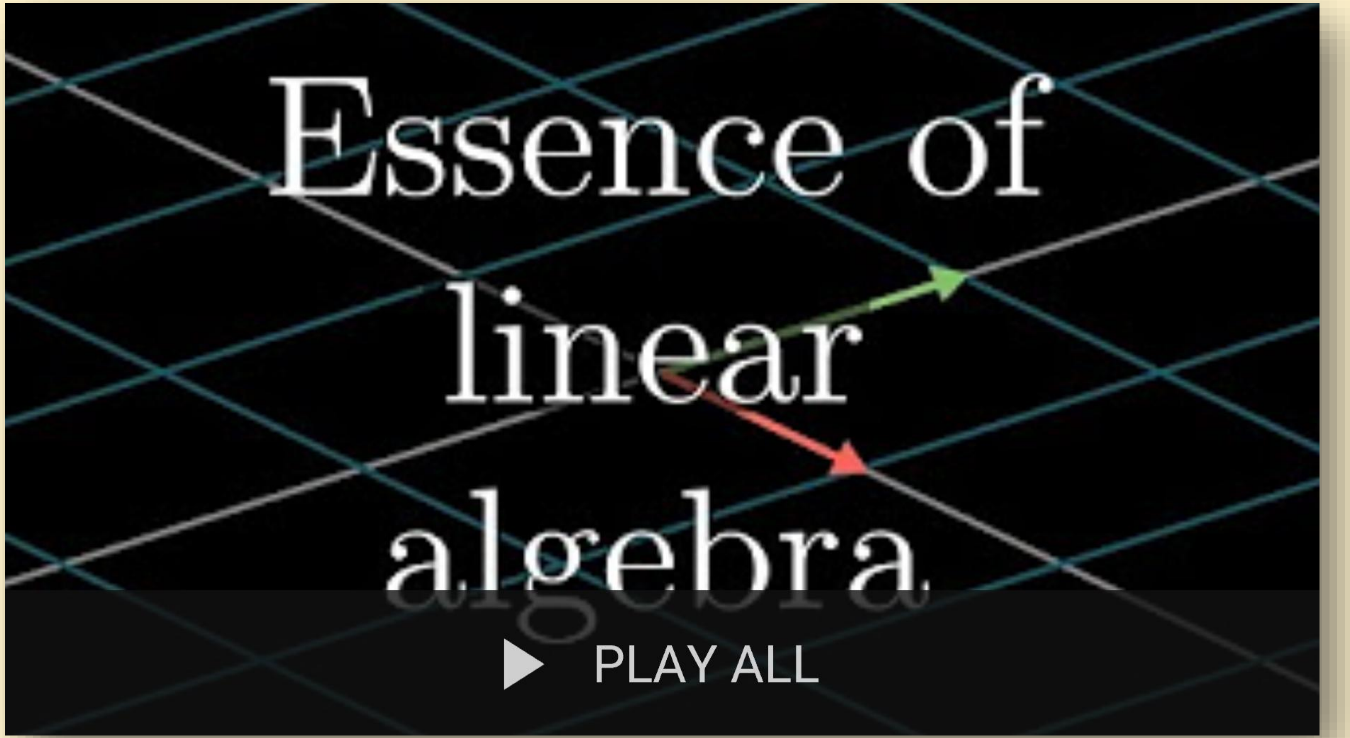
Java

- Use `Vector3` to represent colors (from `JavaVectors.zip` on Teams)
- Use Swing/`MemoryImageSource`

C/C++

- Use SDL

Linear Algebra Brush-Up



https://www.youtube.com/playlist?list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab