

# Light

Light makes it right

# Lecture 8: Light

DHBW, Computer Graphics

Lovro Bosnar

22.2.2023.

# Syllabus

- 3D scene
    - Object
    - Camera
    - Light
  - Rendering
  - Image and display
- Light
    - Light and color
    - Real world light sources
    - Light model
      - Physical light
      - Non-physical light
    - Shadow





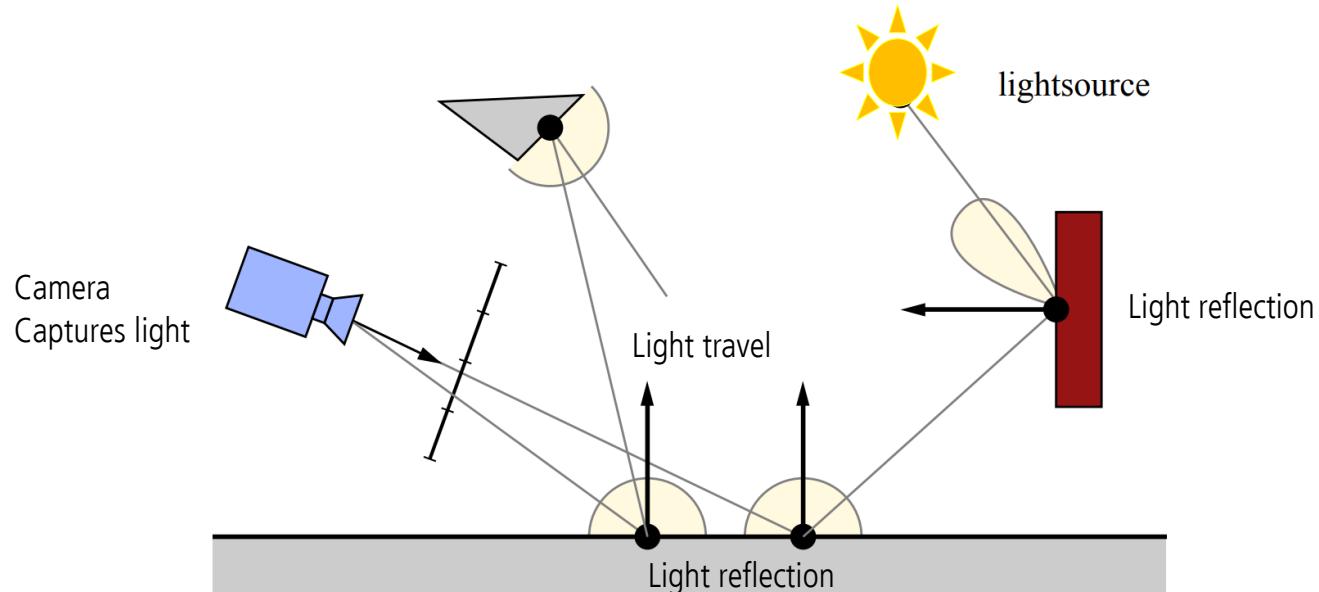
Light is important because without light...

Except of light sources<sup>3</sup>

# Light and color

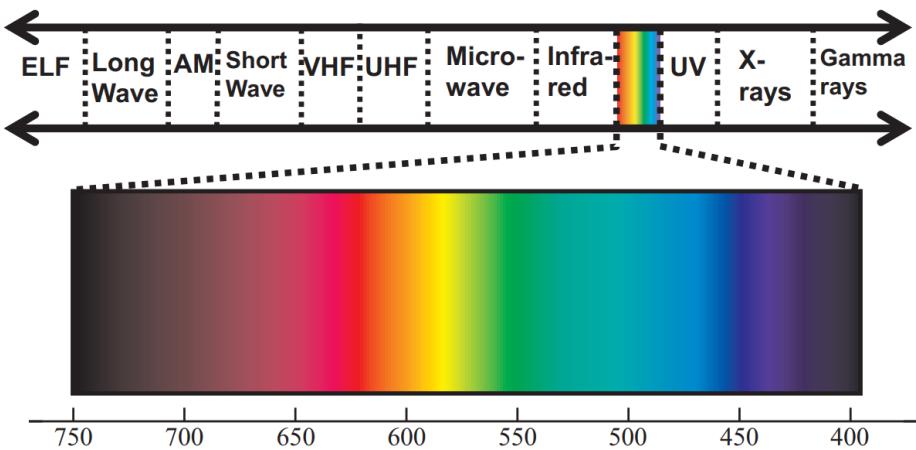
# Light and image formation

- Light is emitted from light sources, travels along rays and interacts with object in the scene (e.g., surface reflection)
- Image created by the sensor (e.g., camera film) depends on light falling on it
- This process is described with **rendering equation**.

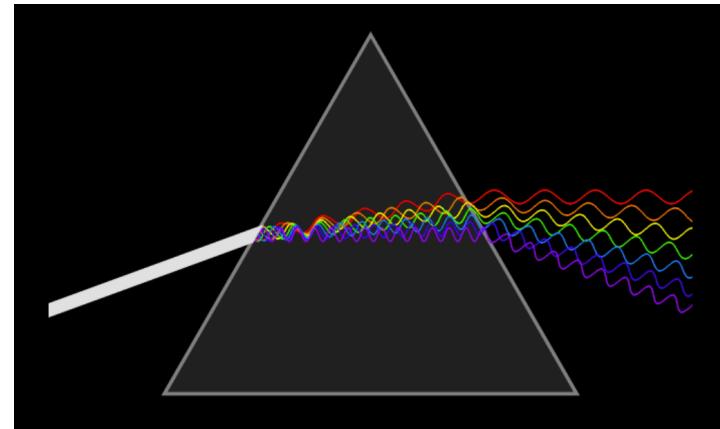


# Light frequency and color

- Light is electromagnetic wave (or particle) visible by eye
  - Visible spectrum: wavelength 400 – 700 nm
  - Different wavelengths (frequencies) are perceived as different **color**
    - color is **perceptual (psychophysical)** phenomena rather than **physical** one



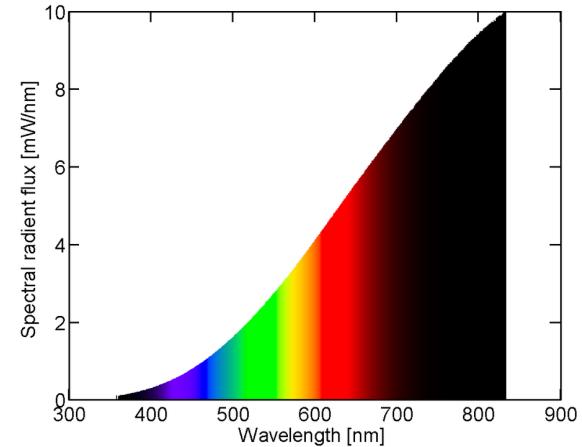
[https://blog.selfshadow.com/publications/s2013-shading-course/#course\\_content](https://blog.selfshadow.com/publications/s2013-shading-course/#course_content)



White light is made of all visible colors – adding all colors from light spectrum gives white light (color additivity, e.g. display)  
[https://en.wikipedia.org/wiki/Dispersive\\_prism](https://en.wikipedia.org/wiki/Dispersive_prism)

# Spectral power distribution

- Light is generally characterized by **spectral power distribution (SPD)**
  - range and intensity of colors it produces at each wavelength in the visible spectrum
- Sun is reference for natural light
  - Standard: **daylight illuminant - D65** by International Commission on Illumination (midday sun Western/Northern Europe)
- **Color rendering index**
  - Ability of light source to reproduce color of object seen under D65, standard by CIE (Commission Internationale de l'Eclairage)
- White color is composed of all colors in equal amount: constant SPD
  - Doesn't exist in real world: CIE; E illuminant

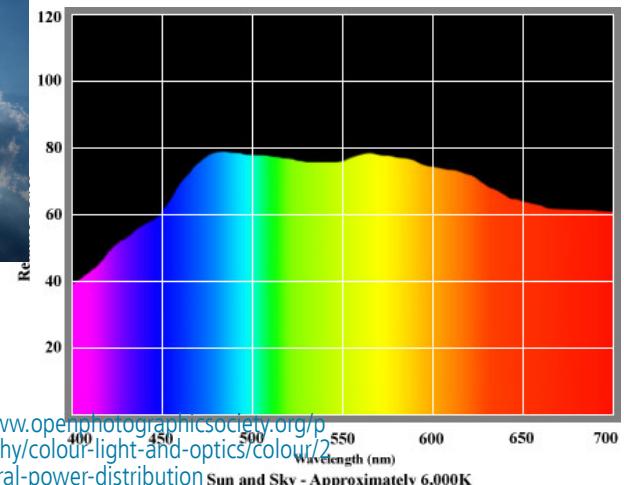


Perceive light as yellow since eye is less sensitive to red

[https://en.wikipedia.org/wiki/Incandescent\\_light\\_bulb](https://en.wikipedia.org/wiki/Incandescent_light_bulb)



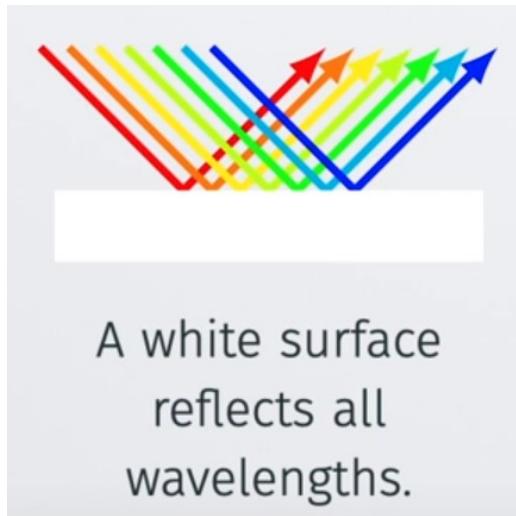
<https://commons.wikimedia.org/wiki/File:Sun-in-the-sky.jpg>



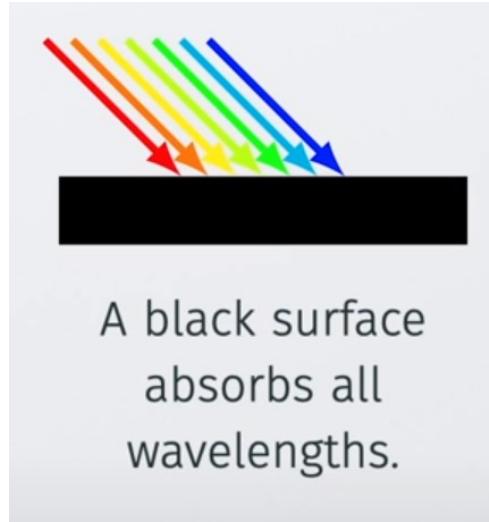
[http://www.openphotographicsociety.org/b\\_photography/colour-light-and-optics/colour/2\\_03-spectral-power-distribution](http://www.openphotographicsociety.org/b_photography/colour-light-and-optics/colour/2_03-spectral-power-distribution) Sun and Sky - Approximately 6,000K

# Material reflectance

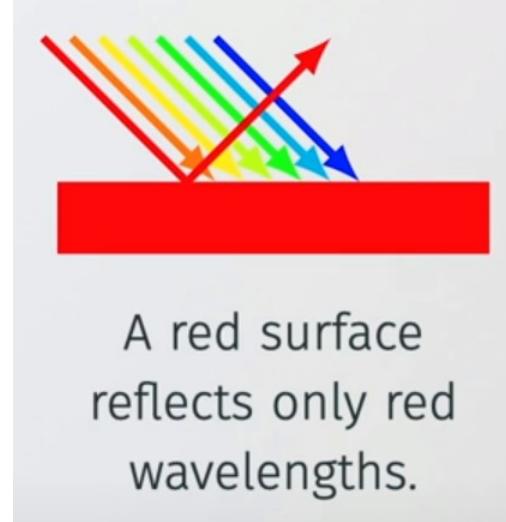
- Light falling on opaque surface will reflect and absorb
- Different materials are characterized with different **spectral reflectance curves**
  - Reflected light is perceived as color of object surface.



A white surface  
reflects all  
wavelengths.



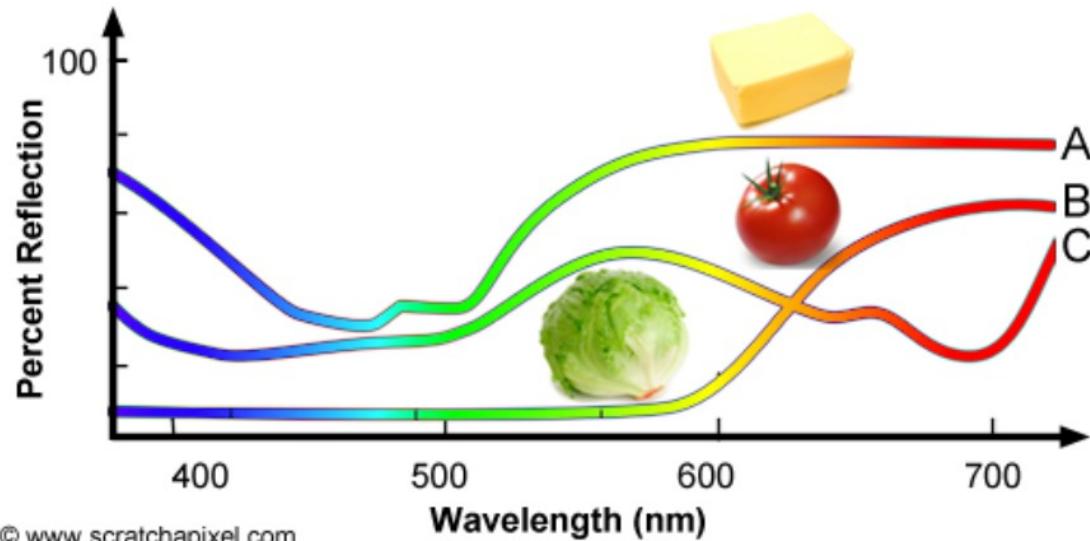
A black surface  
absorbs all  
wavelengths.



A red surface  
reflects only red  
wavelengths.

# Spectral reflectance curve

- SPD of light source defines its color and intensity
- Spectral reflectance curve: SPD describing light reflected from objects
  - Fraction of the light reflected by the surface of the object over the amount of white light illuminating the object.

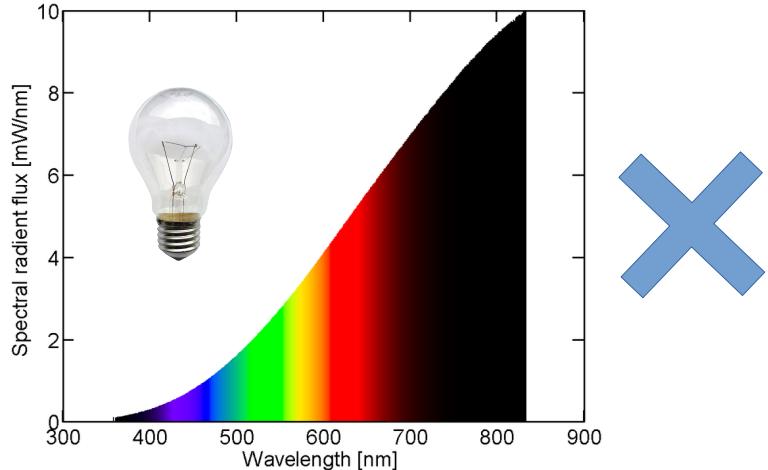


© www.scratchapixel.com

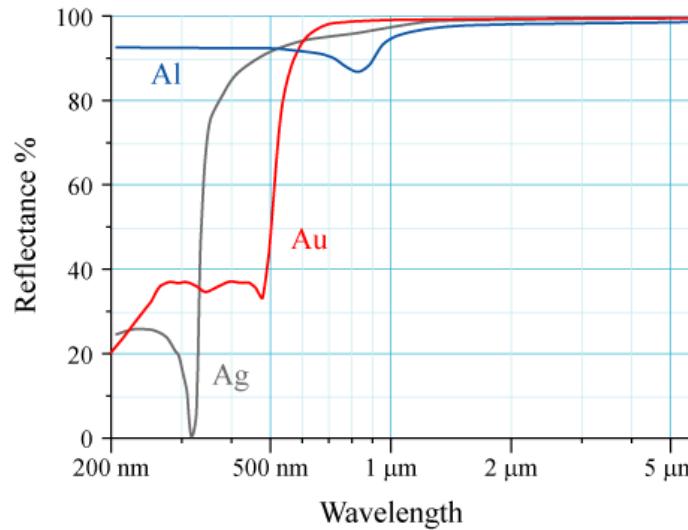
<https://www.scratchapixel.com/lessons/digital-imaging/colors>

# Material reflectance

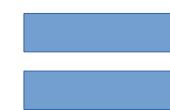
- Reflected energy is calculated using reflectance  $R$  or absorbtion  $A = 1 - R$ ,  $A + R = 1$ 
  - $R$  and  $A$  are spectral values, but in graphics we use RGB values



Power spectral distribution of light → incoming light energy ( $E$ )



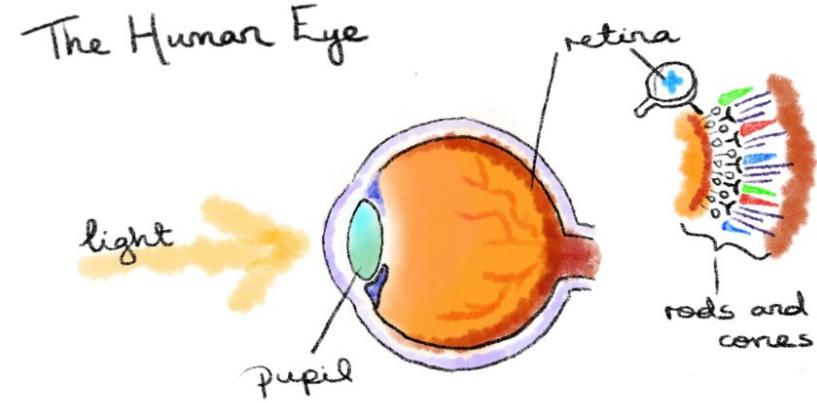
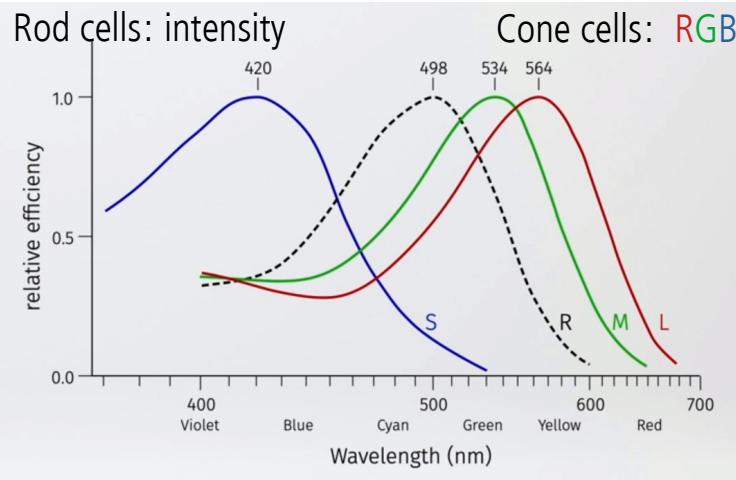
Spectral reflectance curve → Material reflectance ( $R$ )  
<https://en.wikipedia.org/wiki/Reflectance>



$$\text{Reflected} = E * R$$

# Light and human visual system

- Human visual system; back of eye (the retina) is covered with light sensitive receptors – photo-receptive cells:
  - Cone cells – responsible for **trichromatic** color vision
    - “Active” in well lit environments (e.g., daylight)
    - Three types sensitive to: **red, green and blue** light
  - Rod cells – sensitive to smaller amount of light intensity but not reproducing colors
    - “Active” in low lit environments (e.g., night)

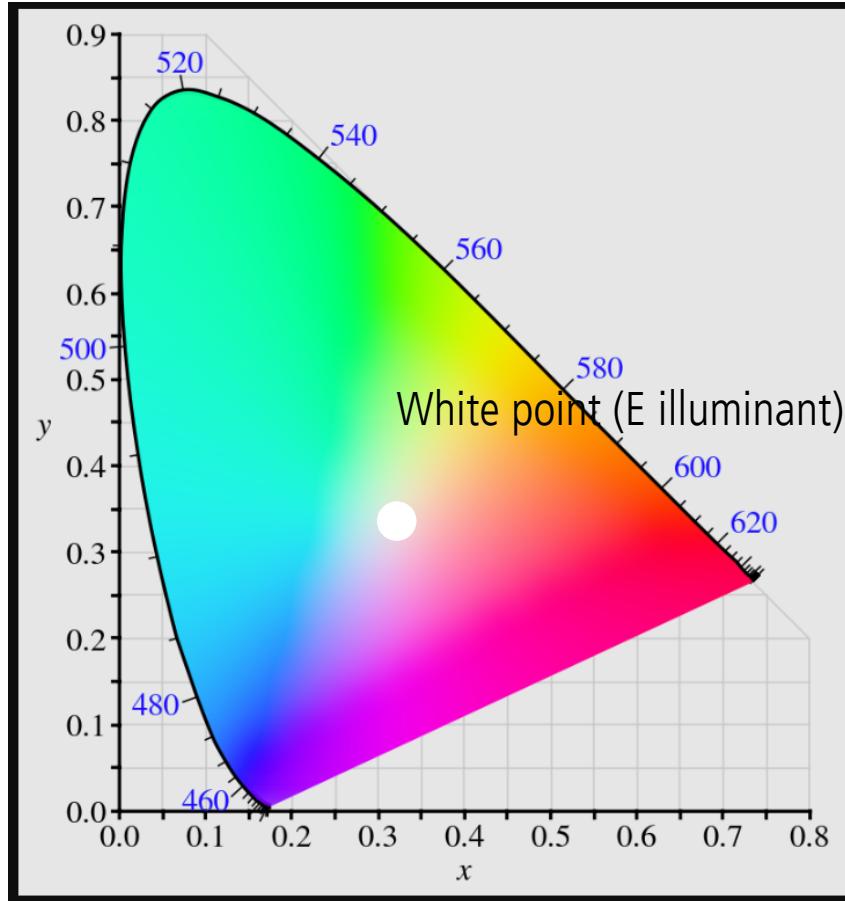


# From SPD to color spaces

- Light from **light source** is characterized with spectral power distribution (SPD)
- Light reflected from **object** is defined with PSD; spectral reflectance curve
- Human visual system can not directly perceive spectral values, that is, PSDs
  - Specifying colors as wavelengths is impractical and in most cases not needed due to trichromatic human vision
  - Wave effects such as diffraction, interference and polarization will be ignored
- PSDs must be transformed in color models or color spaces suited for trichromatic vision
  - PSD is transformed in color hue and color brightness values which we can perceive
  - Geometric optics assumption: light is described with rays with intensity can color

# Colorspace and gamut

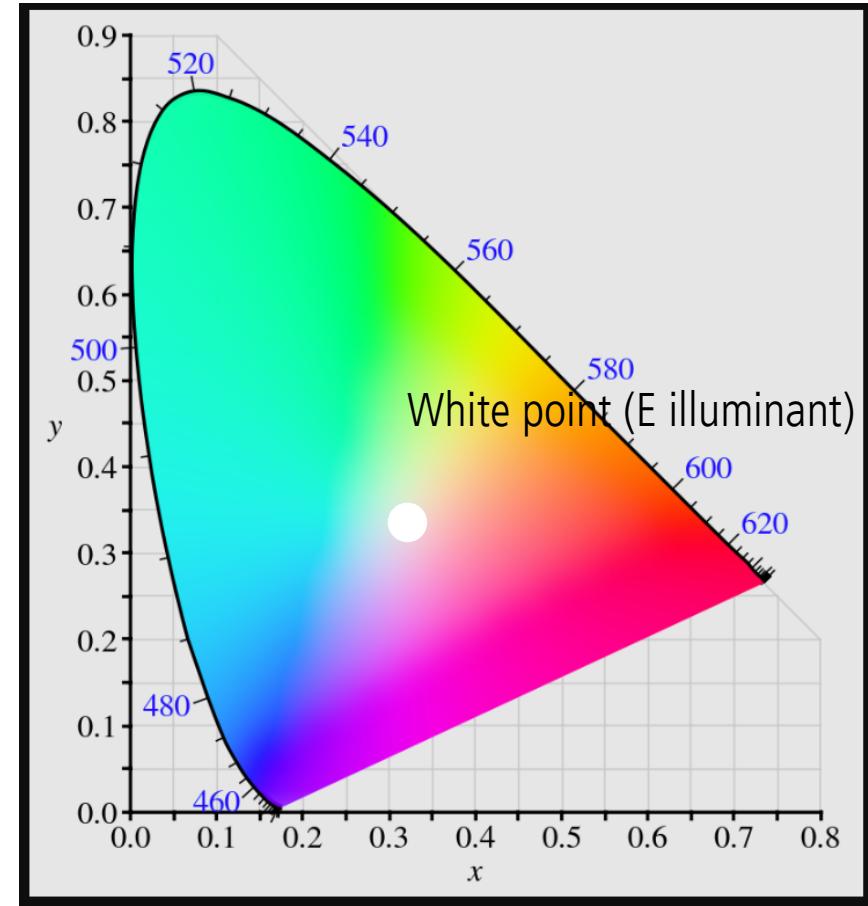
- **Colorspace**
  - Model to represent colors perceived by human visual system → reference for interpretation
  - Way of representing colors through numerical values
  - Specify color of pixels in an image
- **CIE XYZ** is foundation for all color models
  - RGB model is most important for computer graphics
- **Gamut**
  - Range of possible colors that can be represented by particular colorspace



Gamut of XYZ color space:  
[https://en.wikipedia.org/wiki/CIE\\_1931\\_color\\_space](https://en.wikipedia.org/wiki/CIE_1931_color_space)

# XYZ colorspace

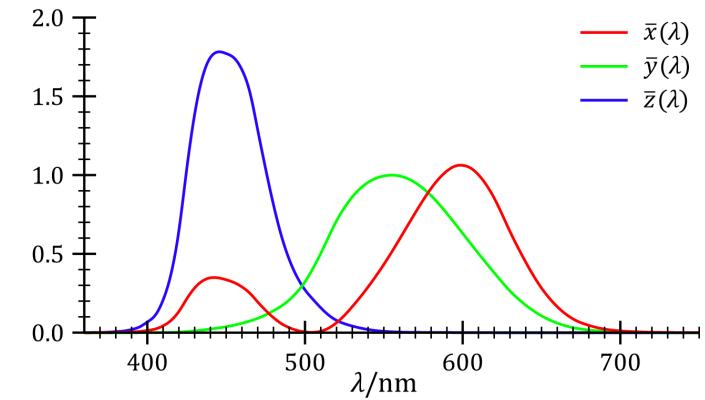
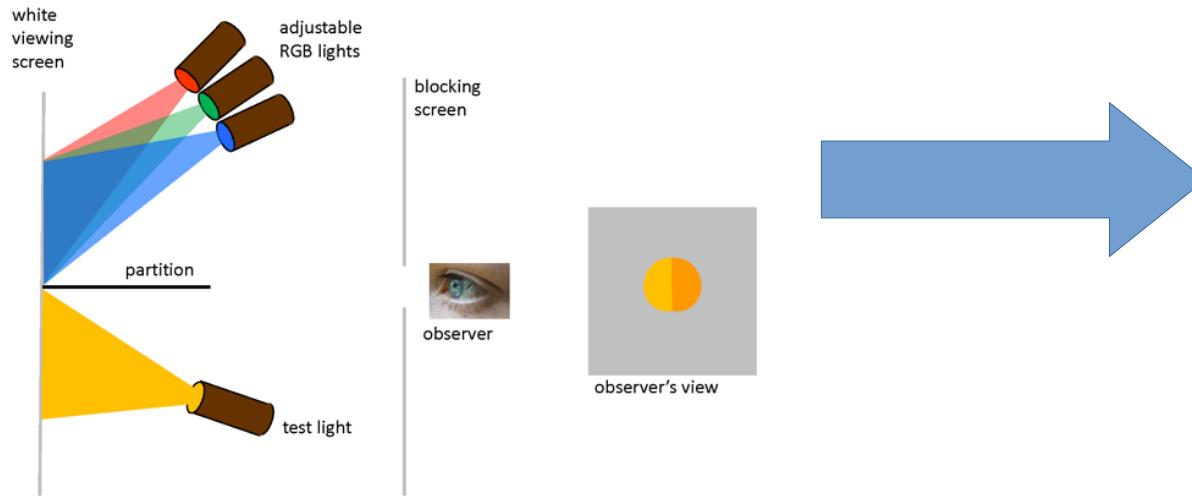
- Assumption: any color (in an additive color model) can be represented by a combination of red, green and blue light
  - Based on cone cells – trichromatic system
- Three curves: **CIE standard observer color matching functions**
  - Y curve is close to luminosity function → expressing the brightness
- Using color matching curve, PSD can be converted to X, Y and Z values
  - Artificial additive primary colors (X, Y, Z)



Gamut of XYZ color space:  
[https://en.wikipedia.org/wiki/CIE\\_1931\\_color\\_space](https://en.wikipedia.org/wiki/CIE_1931_color_space)

# Color matching experiment

- Human visual system, cone cells can distinguish 3 signals: RGB → **trichromatic theory** of color vision
- Therefore, any color perceived by human visual system can be achieved by combining R, G and B signals.
- Colormatching experiment used that idea to form **color matching curves**.



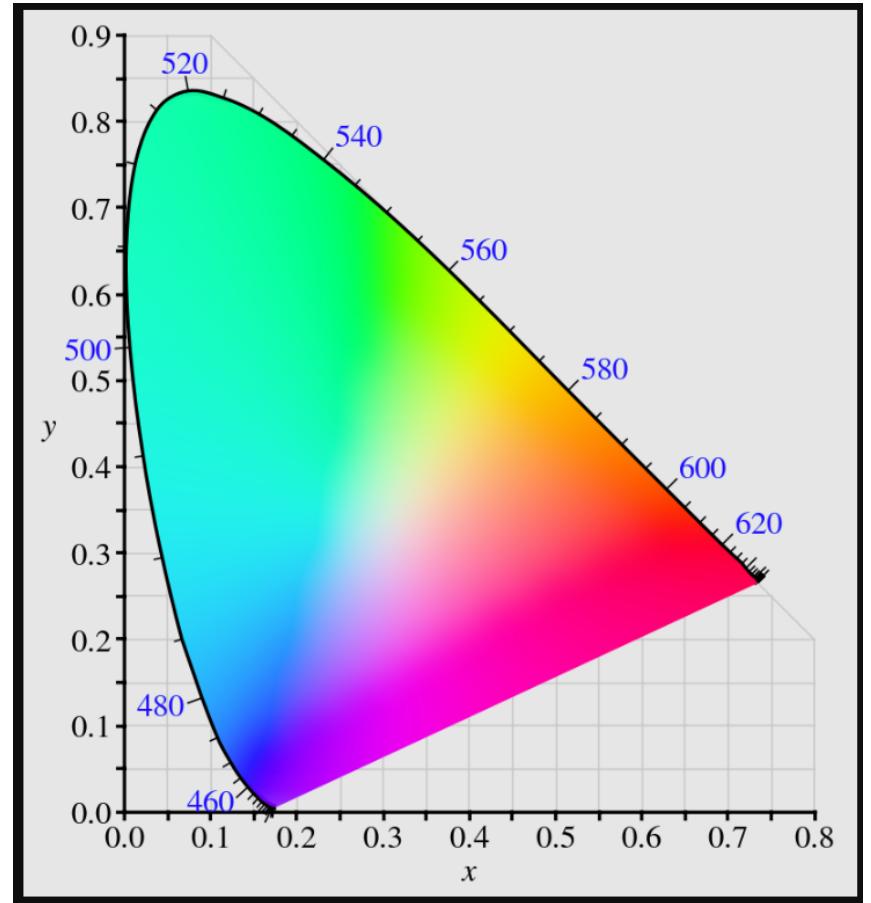
Color matching curves

# XYZ colorspace

- $X + Y + Z = 1$ , gamut can be visualized using:

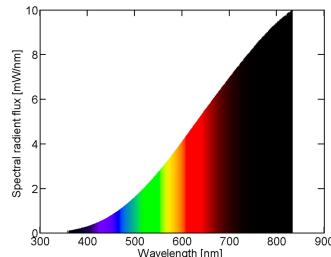
$$x := \frac{X}{X+Y+Z}, \quad y := \frac{Y}{X+Y+Z}$$

- Gamut of **XYZ color space** - gamut of human vision
  - Full range of possible colors visible by human visual system
- XYZ colorspace:
  - International, device-independent standard for color specification
  - System capable representing all colors human visual system can perceive
  - Gold standard for color storage
  - XYZ have to be transformed to RGB space to be visualized

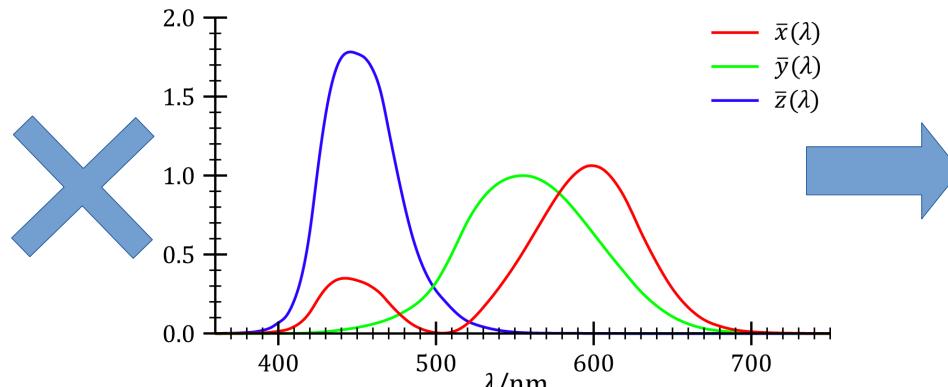


# From PSD of light source to XYZ

- XYZ colorspace is obtained from PSD of light source using color matching curves
  - spectrum and the color-matching functions are discrete in a computer

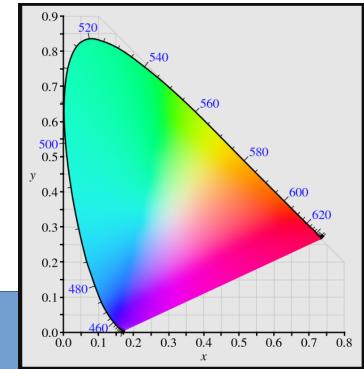


PSD of light source



[https://en.wikipedia.org/wiki/CIE\\_1931\\_color\\_space](https://en.wikipedia.org/wiki/CIE_1931_color_space)

Color matching curves



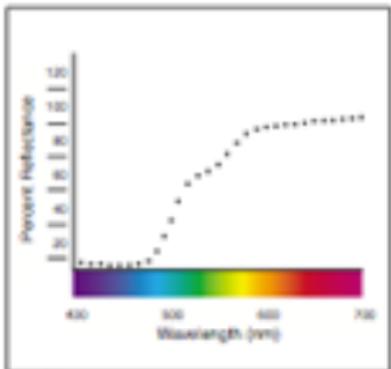
$$X = v_1$$

$$Y = v_2$$

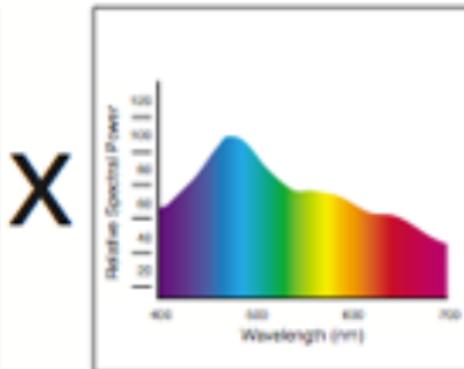
$$Z = v_3$$

# From PSD of reflected light to XYZ

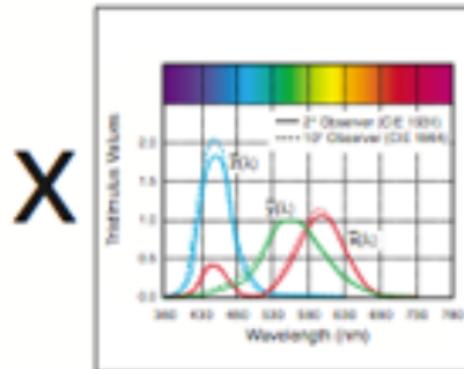
- Light reflected from object is represented as spectral reflectance curve
  - Therefore it must be multiplied by white light which caused the reflection and the standard is D65.



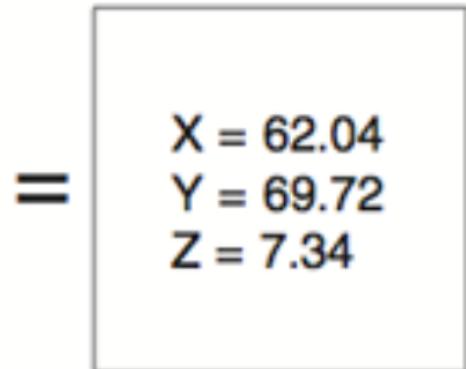
**Spectral  
Reflectance  
Curves**



**D65 Illuminant**



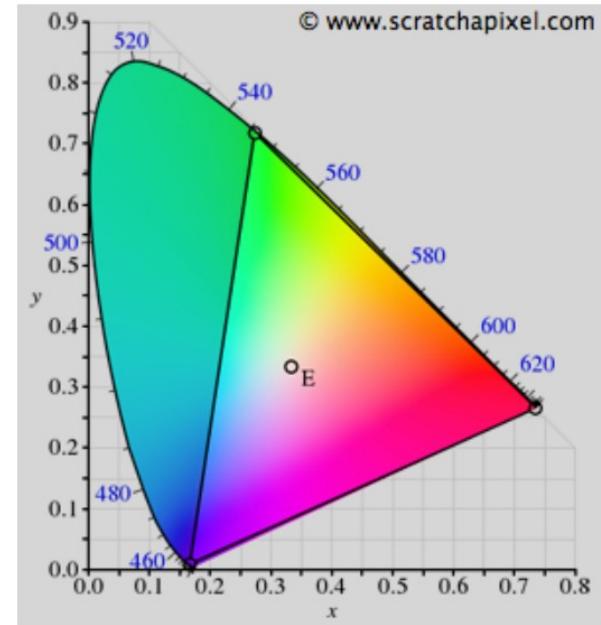
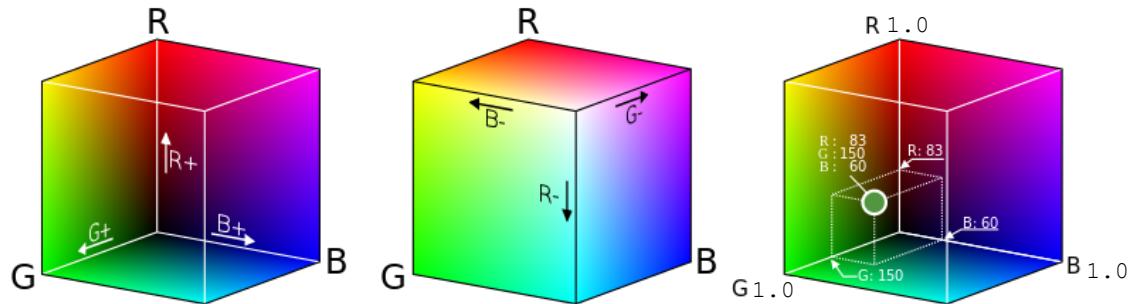
**CIE Color  
Matching Functions**



**Tristimulus  
Values**

# RGB colorspace

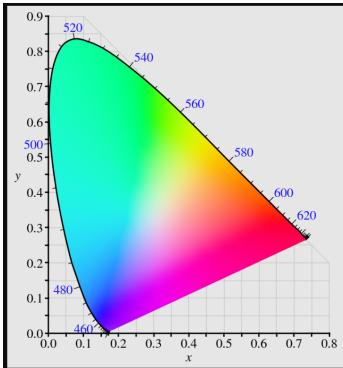
- Similar as human visual system: represent colors as simple sum of primary colors (R, G, B).
- Visualized as cube where R, G and B are placed on axis
- R,G,B intensities [0,1]



XYZ vs RGB gamut

# XYZ to RGB

- XYZ and RGB colorspaces are defined by three coordinates
  - They define a 3D space, which can be plotted (cube for RGB)
- Conversion of color from XYZ to RGB can be seen as moving a point from one 3D space to another.
  - Linear transformation performed using 3x3 matrix
  - Color is not changed, only expressed in different colorspace



XYZ

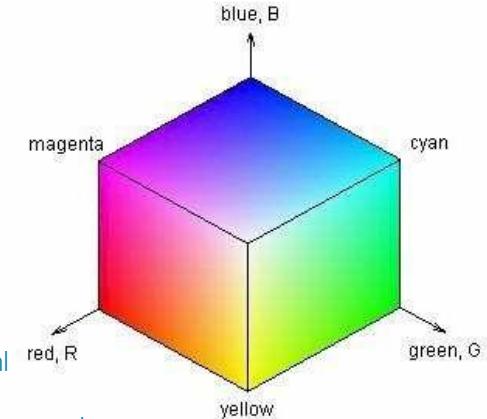
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124564 & 0.3575761 & 0.1804375 \\ 0.2126729 & 0.7151522 & 0.0721750 \\ 0.0193339 & 0.1191920 & 0.9503041 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

[http://www.brucelindbloom.com/index.html?Eqn\\_RGB\\_XYZ\\_Matrix.html](http://www.brucelindbloom.com/index.html?Eqn_RGB_XYZ_Matrix.html)

<http://www.russellcottrell.com/photo/matrixCalculator.htm>

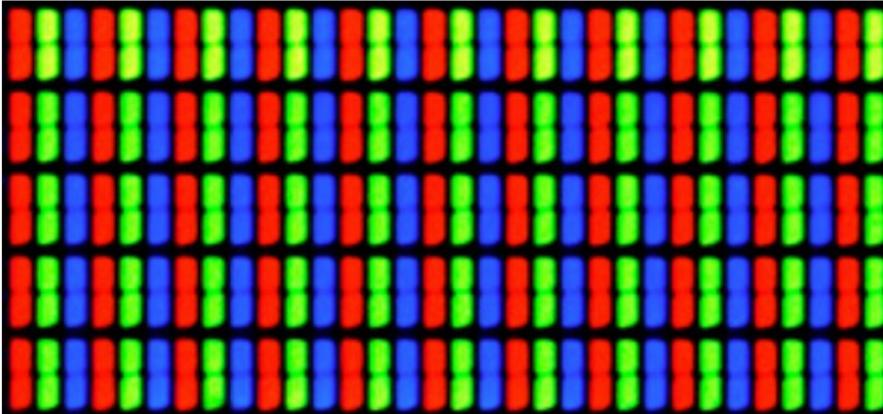
<https://www.oceanopticsbook.info/view/photometry-and-visibility/from-xyz-to-rgb>

<https://terathon.com/blog/rgb-xyz-conversion-matrix-accuracy/>



# RGB colorspace

- Most **computer screens technology** is based on system similar to RGB color model.
  - Summing all RGB values gives white color
- RGB and XYZ are **linear** colorspace
- For rendering, RGB color space is often used
  - PSD is used in **spectral rendering** - more correct rendering, but expensive: computing XYZ from PSD for calculation takes time

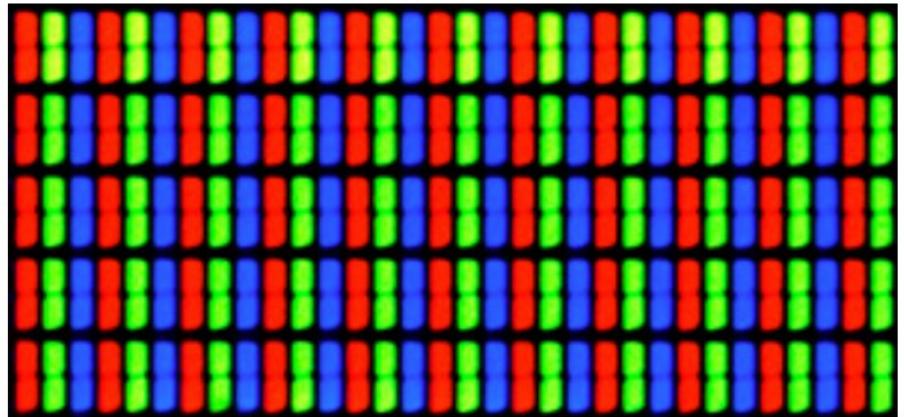
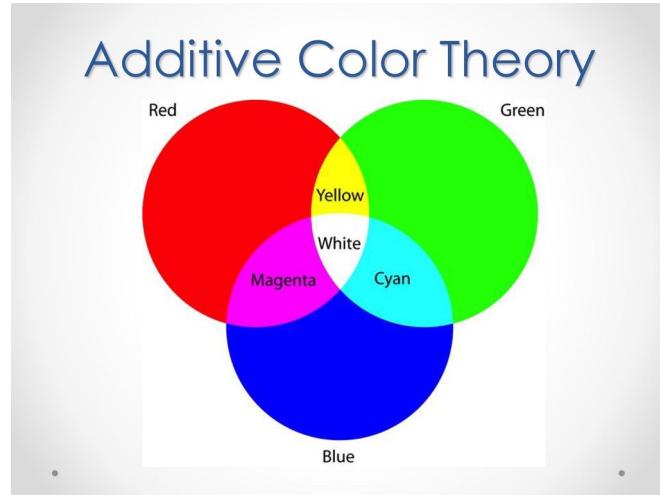


Close-up of LCD screen:

<https://jakubmarian.com/the-illusion-of-rgb-screens/>

# Additive colorspaces

- Main colors: red, green, blue
- Any color is created by adding red, green or blue
- White color is made of red green and blue
- Usage: display devices, computer science, lighting technology → colors are created with light emission
- Display device (e.g., monitor) can further increase or decrease the overall global image intensity (brightness)



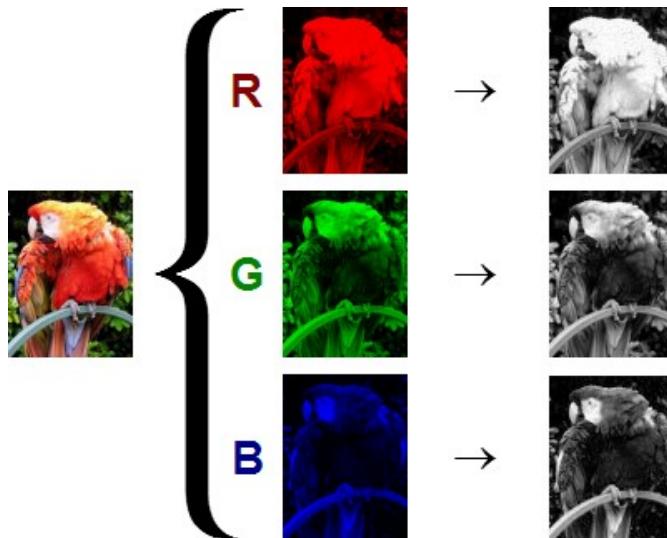
Close-up of LCD screen:

<https://jakubmarian.com/the-illusion-of-rgb-screens/>

<https://carsonparkdesign.com/what-makes-white-on-the-screen/>

# Additive colorspaces

- Typical image formats store three intensities in three channels: R, G, B.
  - e.g., standard images: 8 bit per channel [0,255] range
    - $2^{(8 \times 3)} = 16.7$  million colors (term: true color)
  - e.g., HDR images: 10 or 12 bit per channel



[https://www.mathworks.com/help/matlab/creating\\_plots/image-types.html](https://www.mathworks.com/help/matlab/creating_plots/image-types.html)

# Subtractive colorspace

- Example: CMY colorspace
- Main colors: **cyan, magenta, yellow** - white light is made by subtracting
- Usage: art, painting, printing
- Color from canvas is not a result of light emitted from canvas, but light reflected from canvas
  - When white light hits canvas, what is not absorbed is reflected

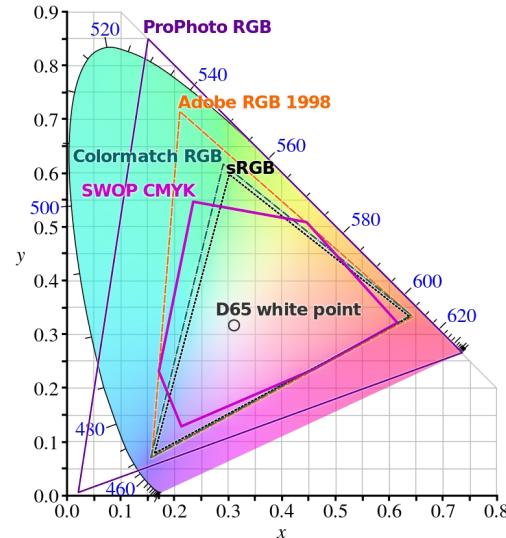


<https://www.greenleafblueberry.com/blogs/news/modern-primary-colors>

# Colorspaces and transformations

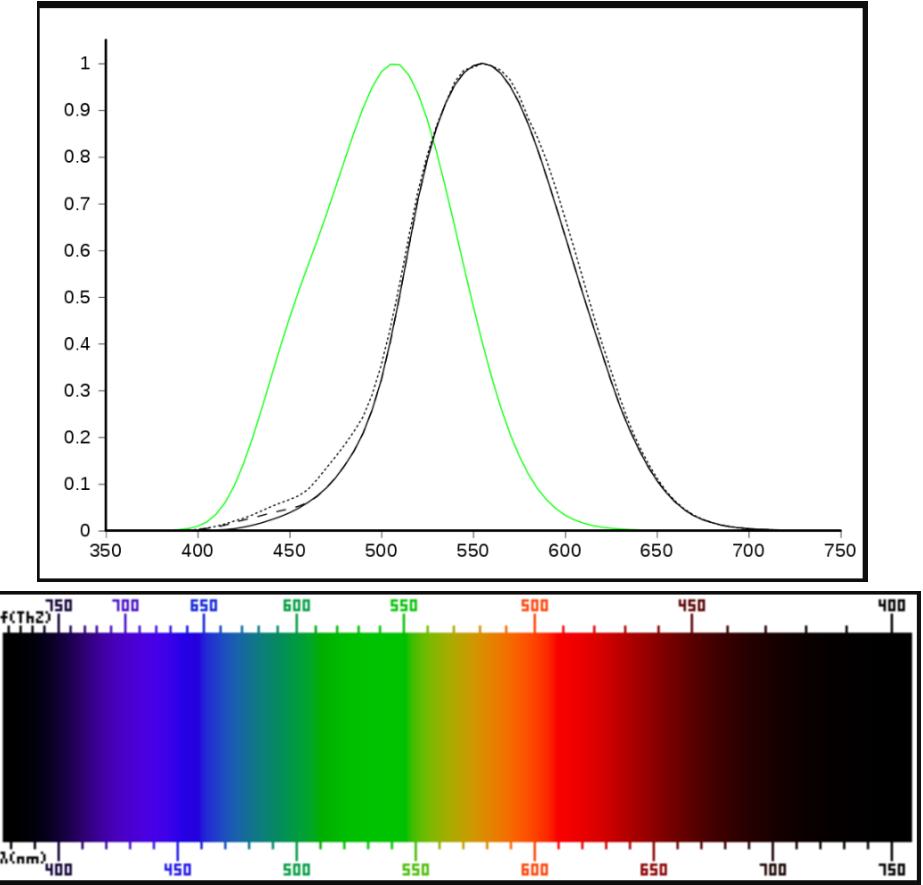
- RGB and XYZ are linear colorspaces and can be transformed using 3x3 matrix
- There are other colorspaces such as sRGB, HSV, HSL are non-linear
- 3x3 matrices can not be used to convert between non-linear and linear colorspace (and vice versa)
  - Any non-linear colorspace must be converted to linear colorspace before 3x3 matrix transformation.
- Different colorspace represent different gamut of colors
- RGB and CMY have small gamut! → device limitations

[https://en.wikipedia.org/wiki/Color\\_space](https://en.wikipedia.org/wiki/Color_space)



# Color brightness

- 555 to 560 nm (green towards yellow) are perceived by the human eye as being the brightest
  - Blue color: the dimmest
  - Green color: the brightest
- How bright color appear to human eye: **luminosity function**
  - Average sensitivity of human eye to different wavelengths for well lit conditions - photopic (cone rods).
- Brightness or lightness is present in **HSV** color model
- **Photometry**: science of measuring light taking in account human visual system
  - CIE photopic spectral luminous curve

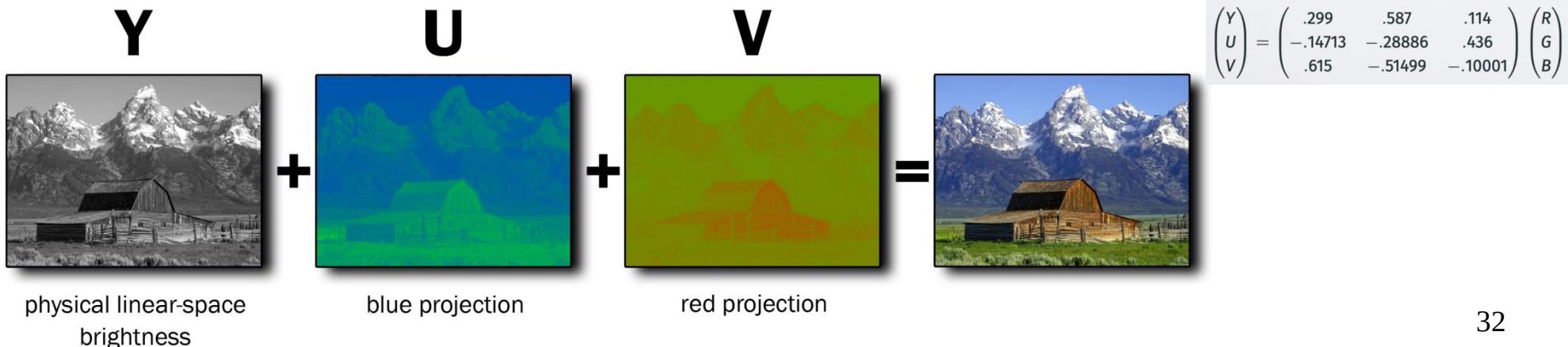


Luminosity function for photopic – green - (well lit) and scotopic (low lit) – black.

[https://en.wikipedia.org/wiki/Luminous\\_efficiency\\_function](https://en.wikipedia.org/wiki/Luminous_efficiency_function)

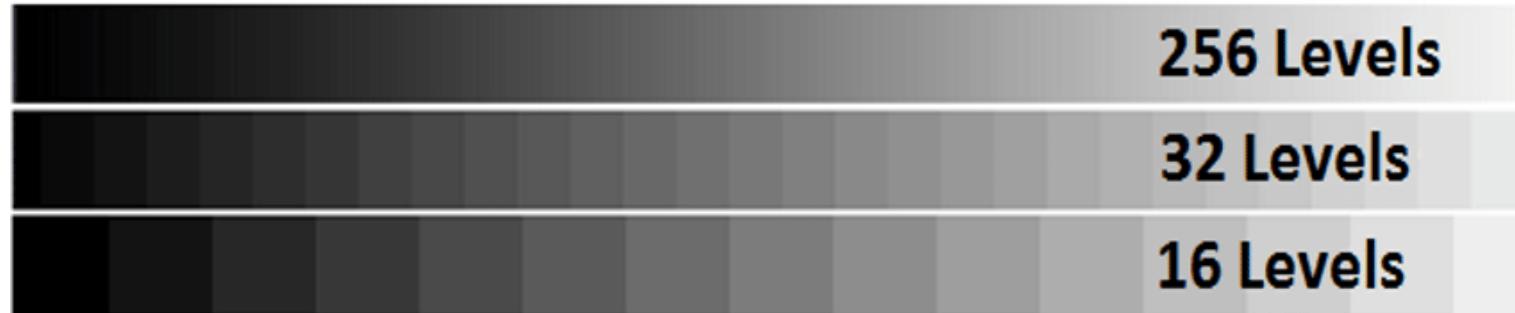
# Brightness (luminance)

- Human eye is much more sensitive to spatial variations in brightness (gray scale) than to spatial variations in color
- Therefore, YUV color space can be particularly useful to highlight the importance of gray scale brightness
  - Y – brightness (luminance)
  - U and V – color (chroma)
- YUV can be obtained from RGB using linear transformations
- YUV is useful for compression: Y channel should get the most resolution (e.g., JPEG and H.264/H.265 formats)



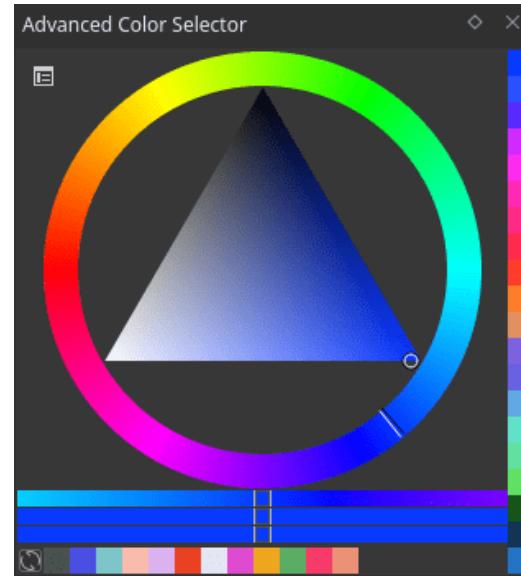
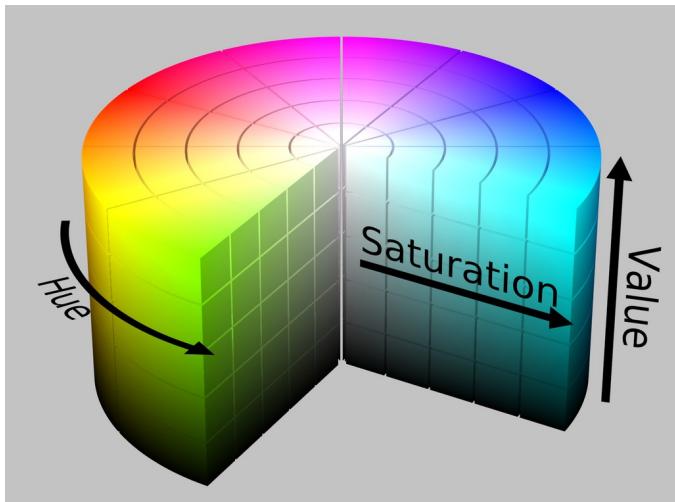
# Brightness (luminance)

- Human eye can very well see small changes in brightness
- Many levels for brightness is needed to describe it without visual discontinuities
- **Dynamic range:** ratio between image with brightest white level and image with darkest black level
  - Very high ratios are possible in real world → image and display limitations!



# HSV colorspace

- RGB colorspace is not always intuitive for “color picking”
- HSV is transformed from RGB:
  - H – Hue - color (rainbow of colors), wheel, given as angle
  - S – Saturation - how white the color is, how much color is mixed with white
  - V – Value – brightness of color
- Non-additive colorspace

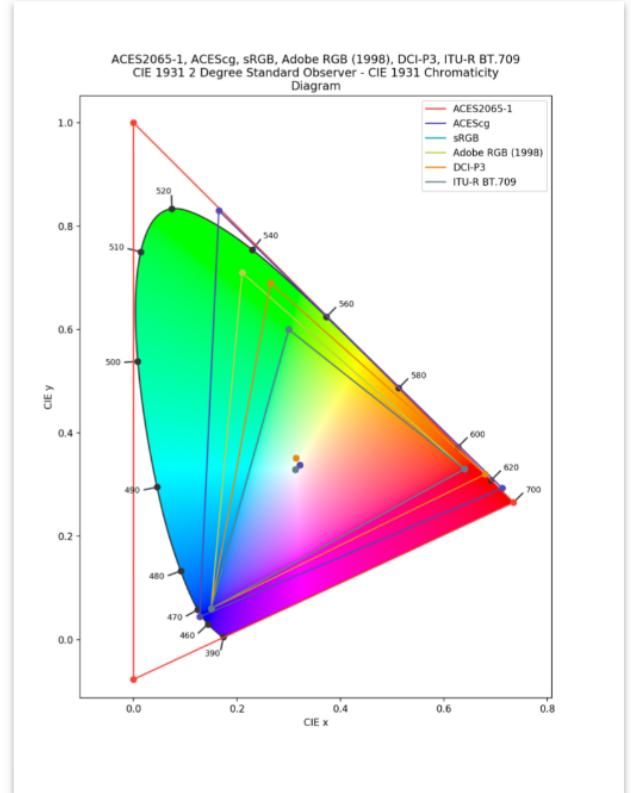


HSV is often used for colorpicking in modeling environments or painting programs:

[https://docs.krita.org/en/reference\\_manual/dockers/advanced\\_color\\_selector.html](https://docs.krita.org/en/reference_manual/dockers/advanced_color_selector.html)

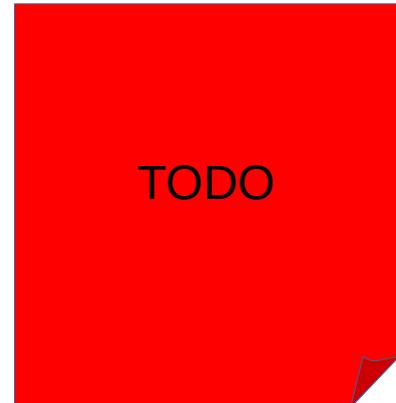
# Note: ACES colorspace

- Academy Color Encoding Specification (ACES) is developed by Academy of Motion Picture Arts & Science technology committee
- Designed to cover the full gamut
- Goal: images should always look the same regardless of used camera and display device.
- Images in ACES color space can't be directly displayed to the screen. Conversion steps:
  - Reference rendering transform
  - Output device transform: depends on display device (computer monitor, digital projector, etc.)
- Becoming defacto standard for professional production



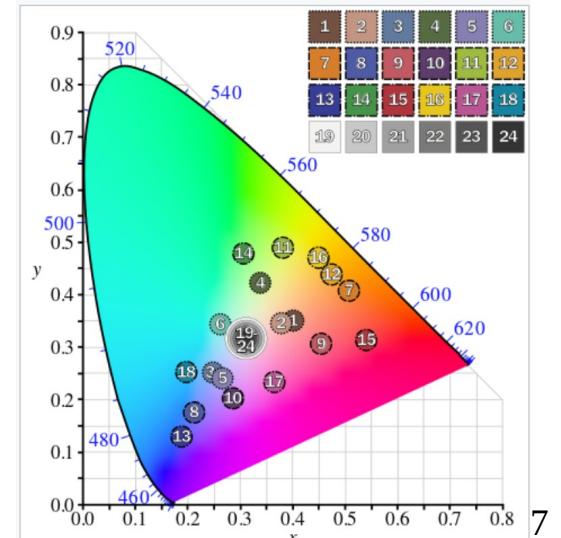
# Color in production

- Digital drawing and image manipulation:
  - Krita: <https://krita.org/en/item/krita-features/>
  - Gimp
- 3D modeling:
  - Blender
- Game engines:
  - Godot



# Note: Macbeth ColorChecker

- Colors have spectral reflectances mimicking materials which are often photographed.
  - Human skin, sky, flowers, etc.
- Goal: consistent color appearance under varying lighting conditions
  - Reference for tweaking exposition and color settings of camera



# Light characterization

- Quantifying light:
  - **Radiometry** – measurement of radiation, physical transmission of light
  - **Photometry** – measurement related to human visual system
    - Summary measures: computed from radiometric quantities.
  - **Colorimetry** – Color perception

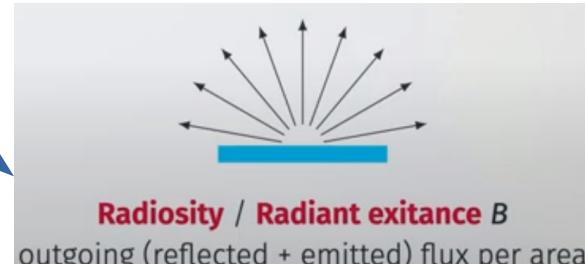
# Radiometric quantities

- Radiant energy ( $Q$ ) – **energy** carried by light (photons) [J]
- Radiant flux ( $\Phi$ ) – power carried by light - **energy over time** [ $W=Js^{-1}$ ]

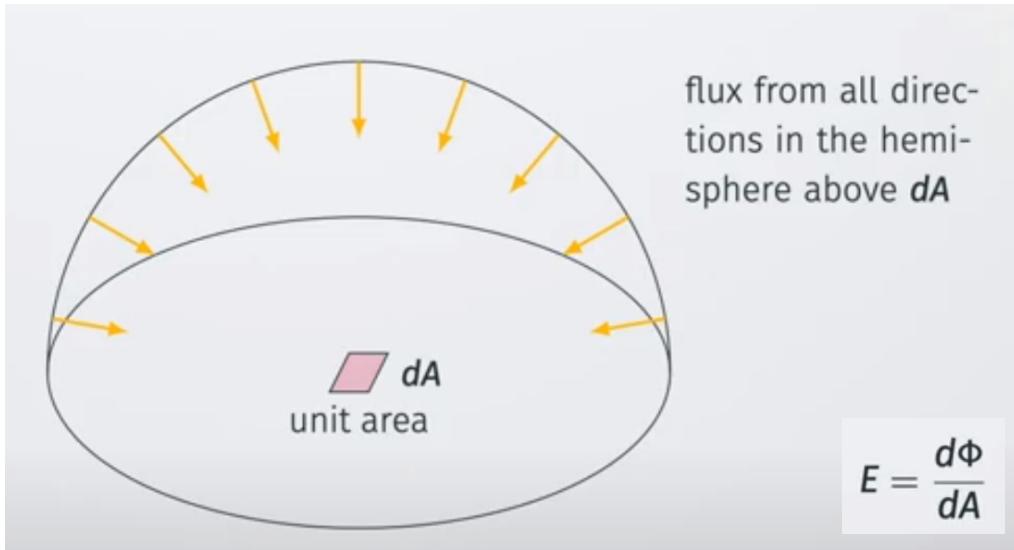
$$\Phi = \frac{dQ}{dt}$$

- Flux density ( $E$ ) – **radiant flux over area** [ $Wm^{-2}$ ]

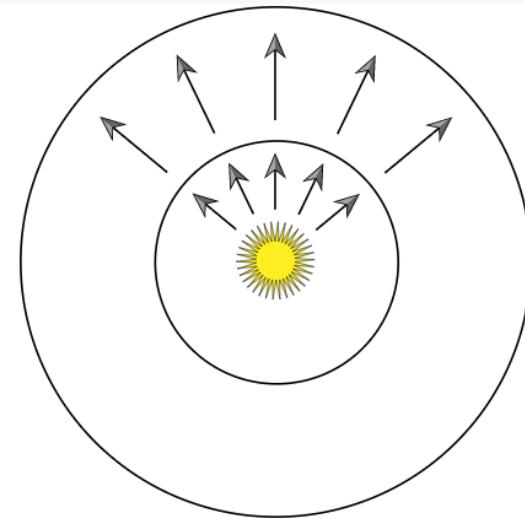
$$E = \frac{d\Phi}{dA}$$



# Radiometric quantities



Irradiance – overall radiant flux (light flow) incoming onto surface.

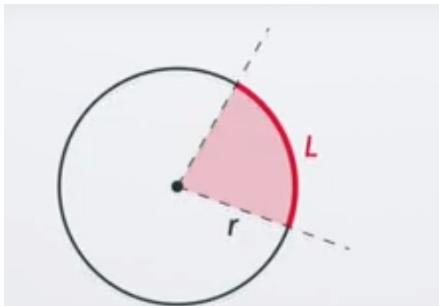


Radiant exitance – radiant flux (light flow) leaving a light source.

Amount of energy received from a light falls off with the squared distance from the light.

[https://www.pbr-book.org/3ed-2018/Color\\_and\\_Radiometry/Radiometry](https://www.pbr-book.org/3ed-2018/Color_and_Radiometry/Radiometry)

# Radiometric quantities

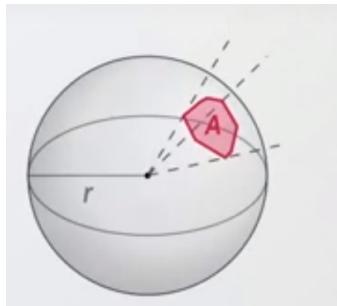


**Planar angle:** angle of segment of length  $L$  on a circle of radius  $r$

$$\alpha = \frac{L}{r}.$$

Angle of full circle:

$$\alpha = \frac{2\pi r}{r} = 2\pi \text{ [rad = radians].}$$

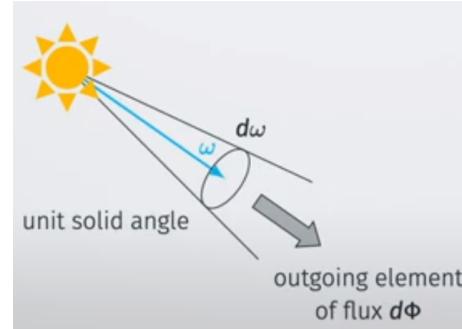


**Solid angle:** area of size  $A$  on a sphere of radius  $r$  (3D extension of the concept of plane angle)

$$\Omega = \frac{A}{r^2}.$$

Angle of full sphere:

$$\Omega = \frac{4\pi r^2}{r^2} = 4\pi \text{ [sr = steradians].}$$

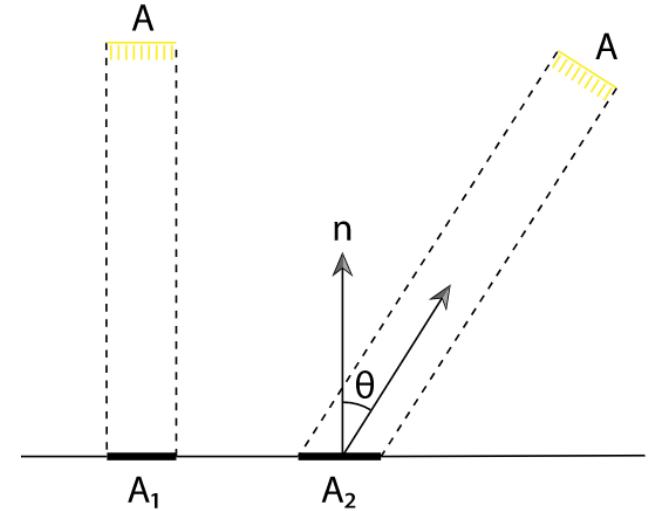


**Radiant intensity ( $I$ )** – radiant flux with respect to **solid angle** [W / sr]

- Radiant flux of light source in specific direction.
- Radiant flux reaching certain point from some point.

# Radiometric quantities

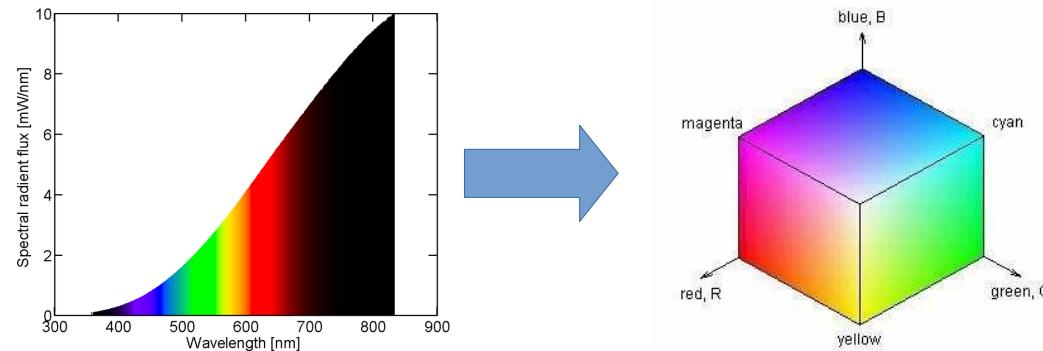
- **Radiance (L)** – radiant flux per unit solid angle, per unit projected area
  - electromagnetic radiation in single ray ( $\text{W} / (\text{m}^2 \text{ sr})$ )
  - This area is measured in a plane perpendicular to ray, if surface has orientation then cosine correction factor must be used
- **Radiance is what cameras measure** – prime importance for rendering
- **The purpose of shading is to calculate radiance along a given ray from a shaded point on object to camera**
  - Radiance is physically based value which is simplified with **color and intensity**
- **Radiance distribution** describes light traveling anywhere in space
  - Rendering can be seen as evaluating radiance distribution for each eye-camera direction\*
- Radiance is not affected by distance
  - Surface is more distant and covers less area.



\* Image-based rendering uses this concept, namely light field technique.

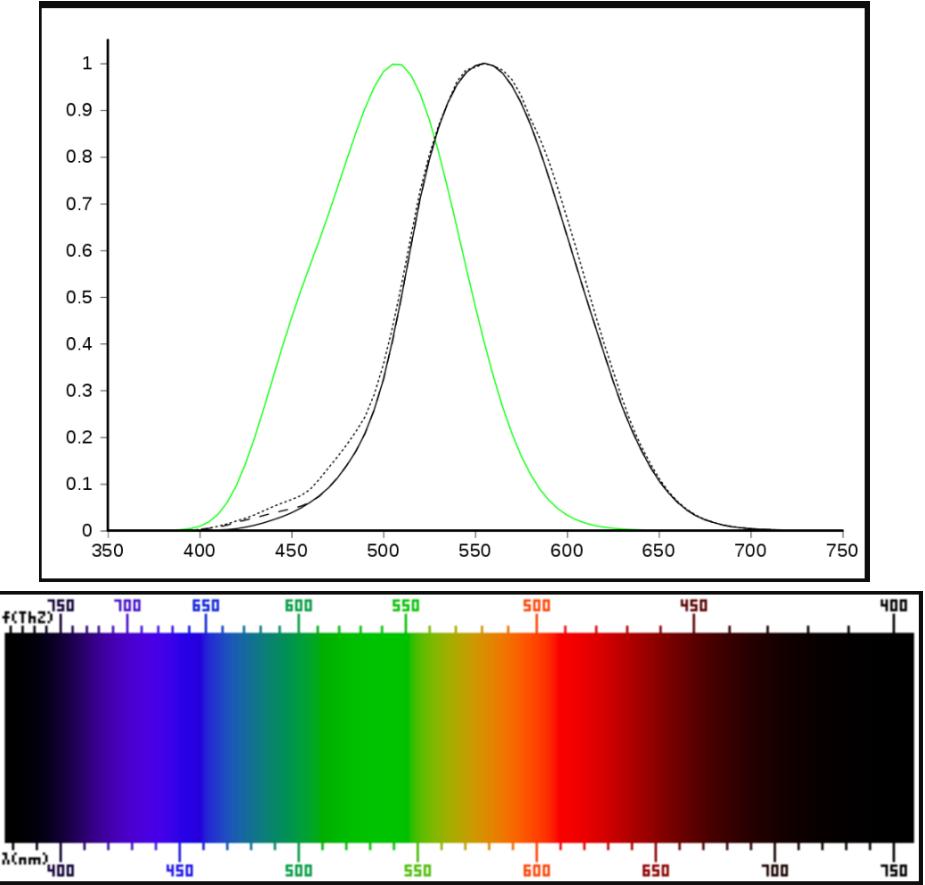
# Radiometric quantities and PSD

- Most light waves contain mixture of many different wavelengths – **polychromatic** (light with one wavelength is called **monochromatic**).
  - Visualized using spectral power distribution (SPD) – plot showing how light's energy is distributed across wavelengths.
- All radiometric quantities have spectral distributions
  - Using full SPDs for rendering is complex\*, often radiometric quantities are represented as RGB triplets.



# Photometry

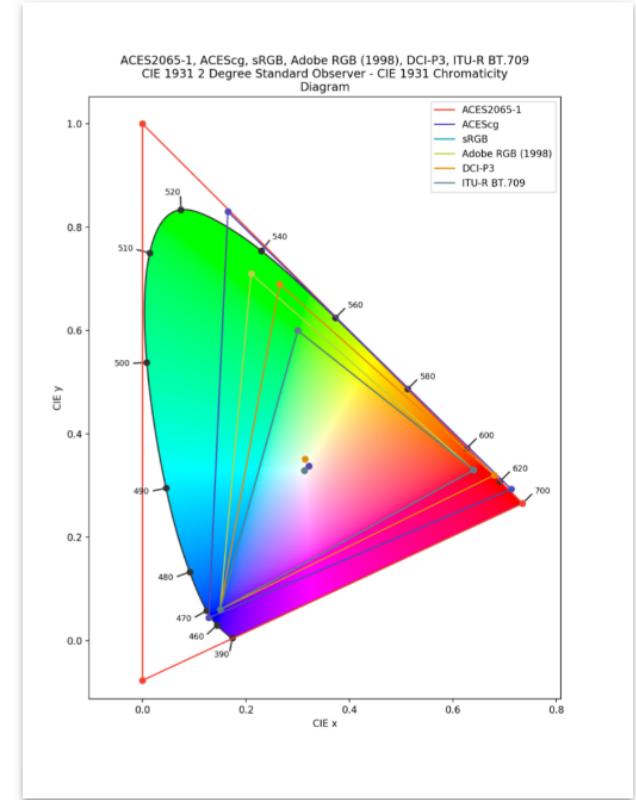
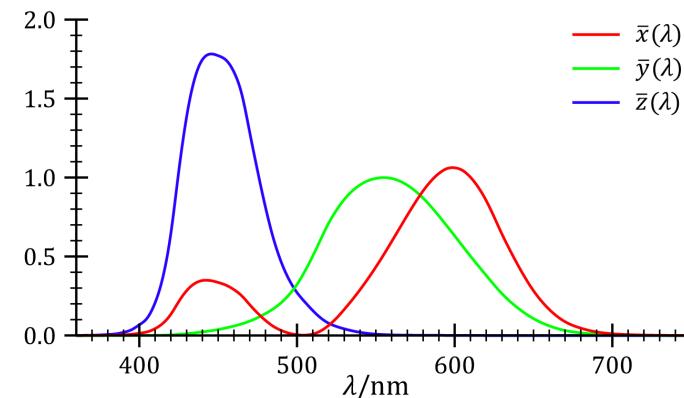
- Photometry is field like radiometry but it takes visual system and perception in account.
- Each radiometric quantity has equivalent photometric quantity
  - Radiant flux → Luminous flux (lumen - lm)
  - Irradiance → Illuminance (lux - lx)
  - Radiant intensity → Luminous intensity (candela - cd)
  - Radiance → Luminance ( $\text{cd} / \text{m}^2$ )
- Radiometric quantities are converted to photometric by using **CIE photopic spectral luminous curve**.



Luminosity function for photopic – green - (well lit) and scotopic (low lit) – black.  
[https://en.wikipedia.org/wiki/Luminous\\_efficiency\\_function](https://en.wikipedia.org/wiki/Luminous_efficiency_function)

# Colorimetry

- Colorimetry: relationship between PSD and perception of color
- CIE standard observer color matching functions
  - PSD → XYZ colorspace
- Define colorspaces and their gamut for visualization
- RGB color spaces are most interesting for rendering:
  - SRGB (linear colorspace with sRGB primaries and white point)



# Rendering with RGB

- RGB is perceptual rather than physical quantity
- RGB in Physically based rendering is a category error
  - Correct would be to use spectral quantities for rendering and RGB conversion in the end for visualization
- For interactive and appearance-based applications (e.g., film) RGB rendering is acceptable.
  - Object surface reflection is defined using RGB rather than spectral reflectance curve
  - Light sources are described with RGB rather than PSD
- For predictive simulations rendering spectral quantities must be used.



Tristimulus vs spectral rendering:  
<https://hal.inria.fr/hal-03331619/document>

# Recap

- Light has physical but also perceptual properties.
- Discussion on radiometry, photometry and colorimetry was needed to understand light and its perception:
  - Color (hue)
  - Intensity
- Light in computer graphics is represented using:
  - RGB values in  $[0, 1]$
  - Intensity, multiplier of RGB values

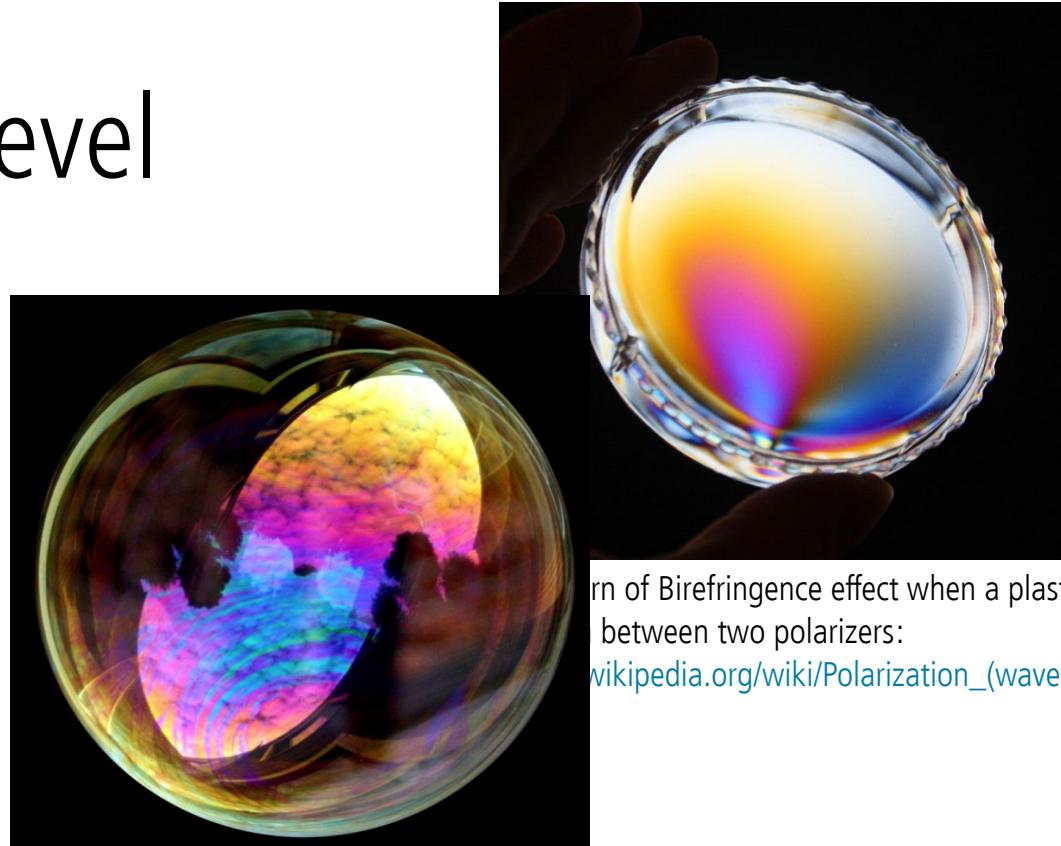
# Real world light and sources of light

# Light description

- Light can be described both on:
  - Microscopic scale: wave optics
  - Macroscopic scale: **geometric optics**

# Light: microscopic level

- Light is quantized – it comes in individual and indivisible packets called **photons**
- At same time light is **wave-like**
- Some effects of light-matter interaction can be only explained (and thus modeled\*) using **wave optics**:
  - Interference
  - polarization
  - Diffraction
  - Iridescence
  - Pleochroism and birefringent
  - Etc.
- Also, knowing microscopic background often helps in getting intuition for some shading and light transport methods\*\*



Learn of Birefringence effect when a plastic box between two polarizers:  
[wikipedia.org/wiki/Polarization\\_\(waves\)](https://en.wikipedia.org/wiki/Polarization_(waves))

White light interference in a soap bubble – iridescence phenomena (thin-film interference) -  
[https://en.wikipedia.org/wiki/Wave\\_interference](https://en.wikipedia.org/wiki/Wave_interference)

\* It is important to note that complexity of model level depends on application. Often, in computer graphics wave effects can be “faked” to achieve desired appearance:

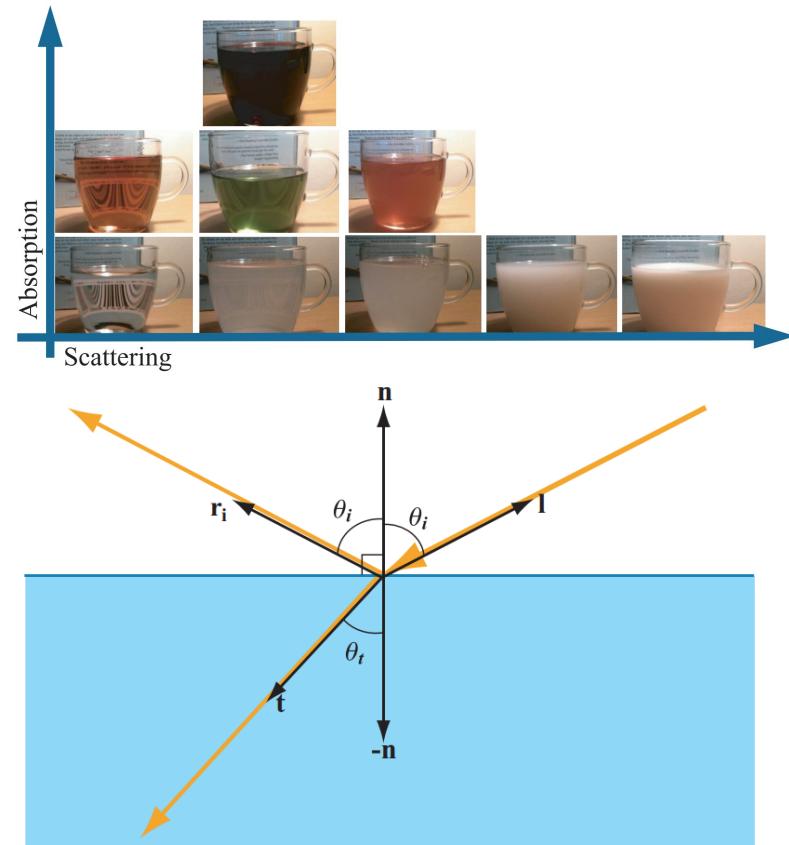
<https://developer.nvidia.com/gpugems/gpugems/part-i-natural-effects/chapter-8-simulating-diffraction>. On the other hand, physically based methods for achieving wave-effects are actively researched area

[https://sites.cs.ucsb.edu/~lingqi/publications/202203\\_practical\\_plt\\_paper\\_lowres.pdf](https://sites.cs.ucsb.edu/~lingqi/publications/202203_practical_plt_paper_lowres.pdf)

\*\* For example photon mapping: [http://web.cs.wpi.edu/~emmanuel/courses/cs563/write\\_ups/zackw/photon\\_mapping/PhotonMapping.html](http://web.cs.wpi.edu/~emmanuel/courses/cs563/write_ups/zackw/photon_mapping/PhotonMapping.html)

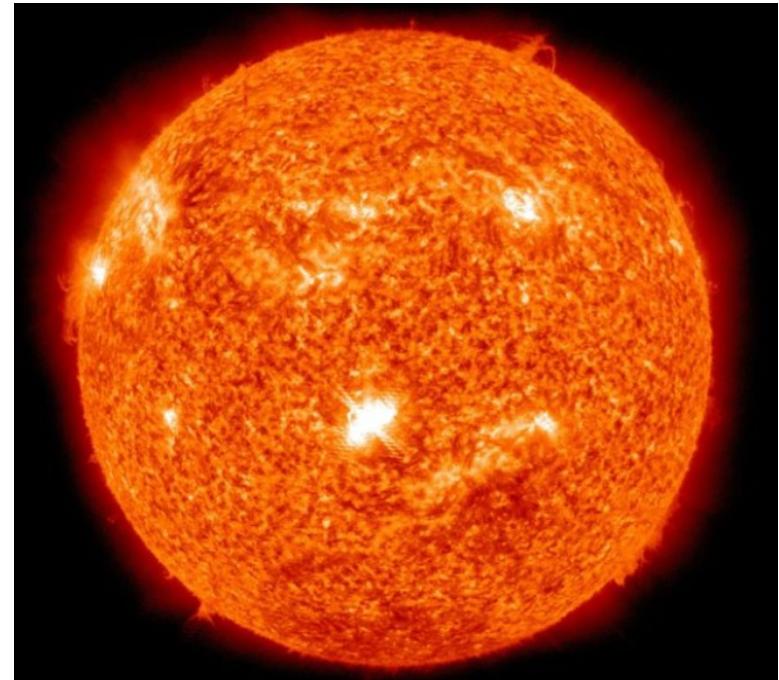
# Light: macroscopic level

- Light in computer graphics is often modeled at macroscopic level – **geometric optics**.
  - Assumption: features with which light interacts are relative to light wavelength (or larger)
  - Light is described with rays.
  - Microscopic light behavior is described with **index of refraction (IOR)**
- In medium of constant IOR, light energy travels in a straight line
  - It only can get **absorbed**
- Light passing from one to another medium with different IOR changes speed
  - This is described with light **scattering**
  - Interface between media is represented by **surface**
- Scattering of light on surface is described with reflection and refraction described with Snell law (light direction) and Fresnel equations (light energy).



# Sources of light

- In a real world, **every** light source has a physical shape and size, e.g., Sun or very hot objects.
- Motion of atomic particles that hold electrical charge causes objects to emit electromagnetic radiation over a range of wavelengths (Maxwell's equations)
- Different way how energy is converted into electromagnetic radiation:
  - Incandescent (tungsten) lamps
  - Halogen lamps
  - Gas-discharge lamps
  - LED lights



<https://education.nationalgeographic.org/resource/sun>

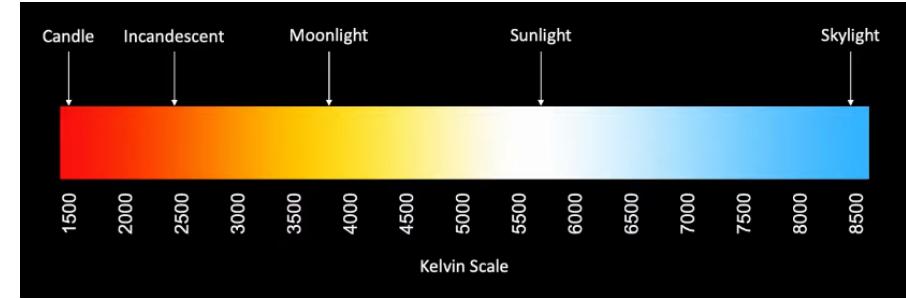
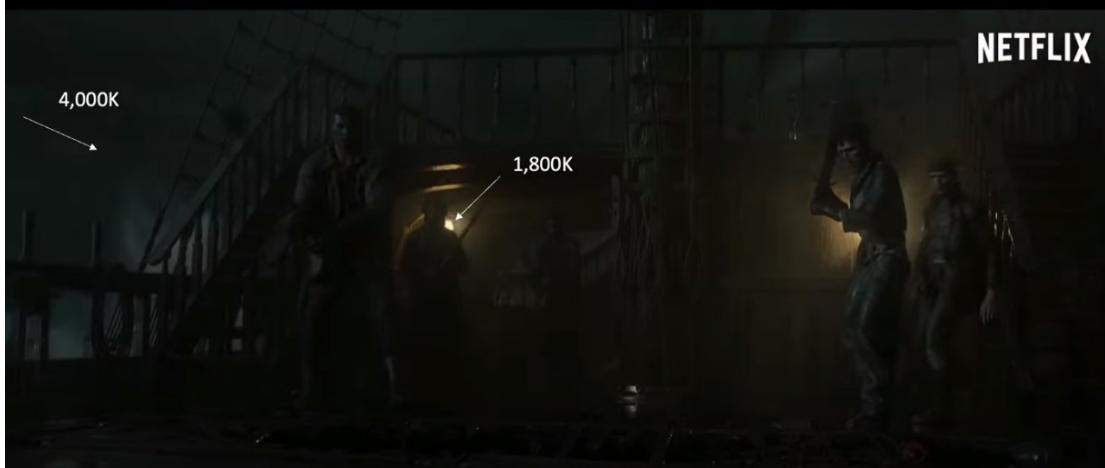
# Color temperature

- Object which is heated to certain temperature emits light
- Color of emitted light depends on temperature
- Black body radiation



[https://www.inlineelectric.com/color\\_temperature](https://www.inlineelectric.com/color_temperature)

# Color temperature



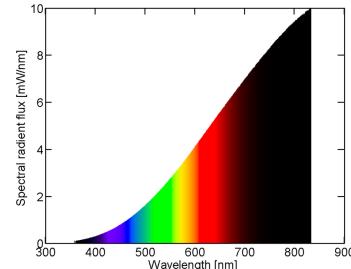
Secrets of photorealism:

[https://www.youtube.com/watch?v=Z8AAX-ENWvQ&t=2s&ab\\_channel=Blender](https://www.youtube.com/watch?v=Z8AAX-ENWvQ&t=2s&ab_channel=Blender)

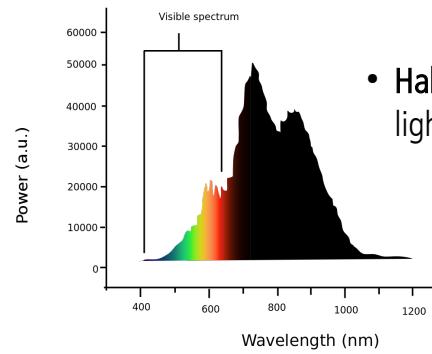


Toy Story 4 (2021)

# Sources of light

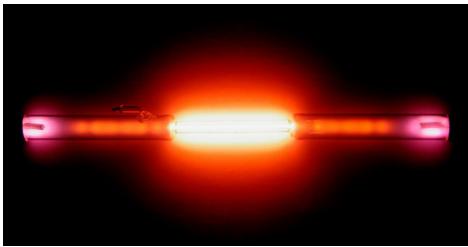


- **Incandescent (tungsten) lamps:**
  - flow of electricity through tungsten filament heats it up and causes it to emit electromagnetic radiation with distribution of wavelengths depending on filament temperature.
  - A frosted glass enclosure is often present to absorb some of the wavelengths.



- **Halogen lamps:** tungsten filament is enclosed in halogen gas. Part of the filament in an incandescent light evaporates when it's heated. Halogen causes evaporated tungsten to return to filament

# Sources of light



Helium: [https://en.wikipedia.org/wiki/Gas-discharge\\_lamp](https://en.wikipedia.org/wiki/Gas-discharge_lamp)



LED lights:  
[https://en.wikipedia.org/wiki/Light-emitting\\_diode](https://en.wikipedia.org/wiki/Light-emitting_diode)

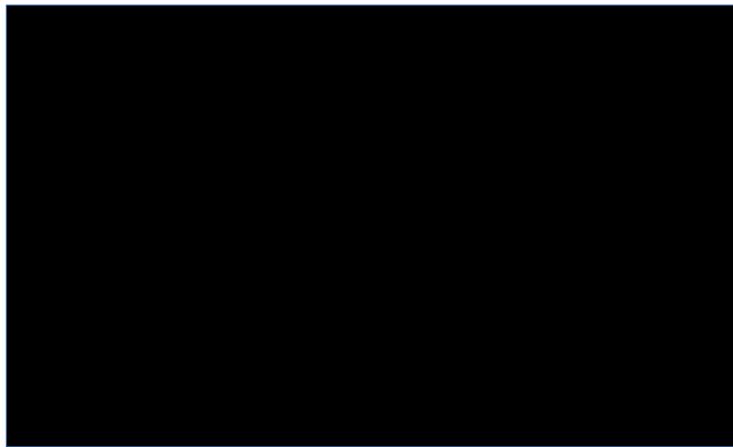
- **Gas-discharge lamps:** passing electrical current through hydrogen, neon, argon, or vaporized metal gas, causes light to be emitted at specific wavelengths that depend on the particular atom in the gas.
- Fluorescent coating on the bulb's interior is often used to transform the emitted frequencies to a wider range.

- **LED lights** are based on electroluminescence: they use materials that emit photons due to electrical current passing through them.

# Light model

# Rendering and light

- Light is emitted from a light source, bounces around the scene, interacting with object and some (very small!) portion enters camera enabling us to see objects in the scene
- Rendering:
  - Generating image from 3D scene
  - Each pixel of rendered image is associated with 3D surface point visible to camera.
- Color of the object surface is calculated in shading step.
  - Shading takes in account 3D object shape (i.e., normal), object material and incoming light
  - Without light resulting image would be completely black.



# Light and shading

- **Shading:** calculating color of the objects in 3D scene
- Simple shading: light color and scattering model  $f(l, n, v)$

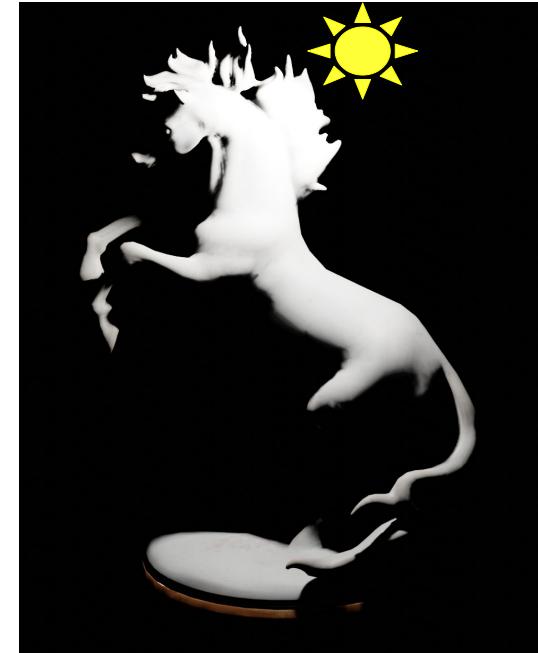
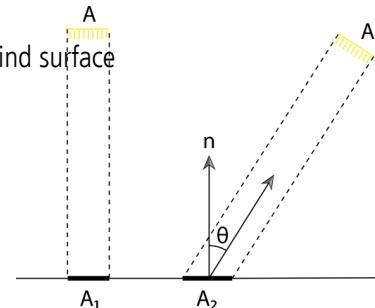
$$c_{shaded} = f(l, n, v)c_{light}$$

- If multiple light sources are present, they add up linearly

$$c_{shaded} = \sum_{i=1}^n f(l_i, n, v)c_{light_i}$$

- Effect of light on surface can be visualized as rays
  - Density of rays hitting the surface correspond to the light intensity.
  - Density of light rays (their spacing) is proportional to cosine of angle between  $l$  and  $n$ .
  - Ray density is proportional to dot product when positive, otherwise light comes from behind surface

$$c_{shaded} = \sum_{i=1}^n (l_i \cdot n)^+ f(l_i, n, v)c_{light_i}$$



# Light and shading

- Final, simplified shading:

$$c_{shaded} = \sum_{i=1}^n (l_i \cdot n)^+ f(l_i, n, v) c_{light_i}$$

- If  $f(l, n, v)$  is constant color,  $\rightarrow$  Lamberitan shading
- Conclusion, light source interacts with shading model with:
  - Vector  $l$  pointing toward light
  - Light color
- Different types of light sources differ in how these two parameters are varied over the scene



# Local and global illumination

- More generally, light reflected from surface is represented with **reflectance equation**:

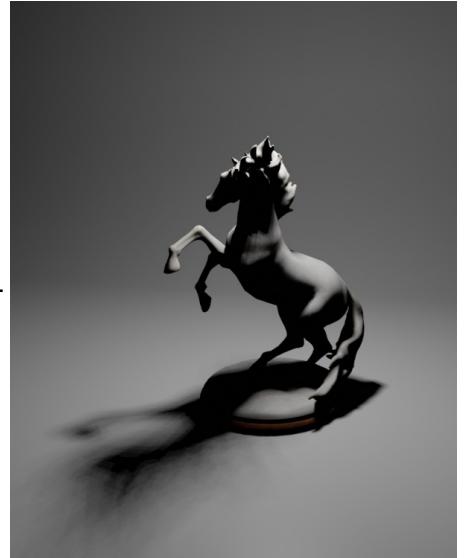
$$L_o(p, v) = \int_{l \in \omega} f(l, n, v) L_i(p, l)(n \cdot l) dl$$

- If shading is considering only light coming from light sources: **local illumination**
  - $L_i$  is given as color and intensity. Position of light source is calculated from its position
  - Therefore, introduced simplified version can be used

$$L_o(p, v) = \sum_{i=1}^n f(l_i, n, v) c_{light_i}(n \cdot l_i)$$

- Any surface can reflect light on another surface: indirect illumination  $\rightarrow$  **global illumination**
  - Integrating over hemisphere to gather all incoming light  $L_i$  is needed
  - Global illumination algorithms simulate light transport through the scene.
  - They solve **rendering equation** where reflectance equation is special case.

Local illumination is used in rasterization-based rendering (Blender, EEVEE)



Global illumination is used in ray-tracing-based rendering (Blender, Cycles)



# Light and shading

- **Light sources** must define
  - Position and thus **direction** of light
  - Color and **intensity** of light
- Additionally, light sources can have:
  - Size
  - Shape



# Light models

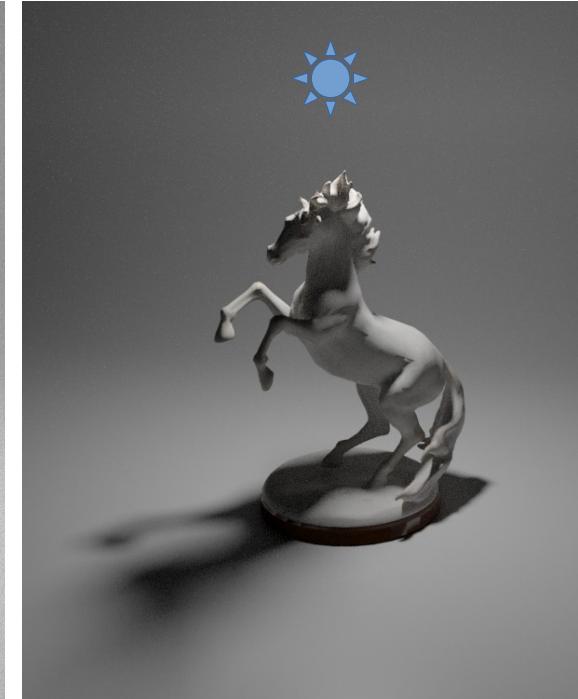
- Representation and modeling of light **physically**, with shape and size is very computationally expensive. Such lights are called **geometric area light or physical lights\***
- Simplification are lights with no physical size, thus called **delta or non-physical lights**



\* When we discussed surface material we mentioned scattering and absorption. Also, materials are able of emission. Therefore, physical lights have 3D shape and material which is emissive. Emissive material is modeled as black body meaning that it absorbs all light falling on it.

# Light models parameters

- Non-physical:
  - Color: RGB triplet in [0,1]
  - Intensity: floating point [0,1] as multiplier to color
  - Position → Direction
- Physical
  - Color
  - Intensity
  - Position
  - Shape
  - Size



# Physical lights

# Physical lights

- Geometry with emissive material
  - Color and intensity
  - Position
  - shape and size
- Physical lights have area which has to be sampled
  - Interaction of physical lights with the scene can not be computed in closed form. Advanced, approximation-based methods such as Monte Carlo integration must be used
- Often used in offline rendering, ray-tracing based rendering

TODO

# Physical lights: offline rendering

TODO

# Physical lights: real-time rendering

- For real-time applications non-physical lights are often used by artists to simulate effects of physical light sources.
- Another alternative is to precompute light and store it in a texture which is then used during shading in render-time
  - General note: this is another example showing that it is always important to assess the application of Computer graphics and see what is really important to model.

TODO

# Non-physical lights

TODO

# Non-Physical lights

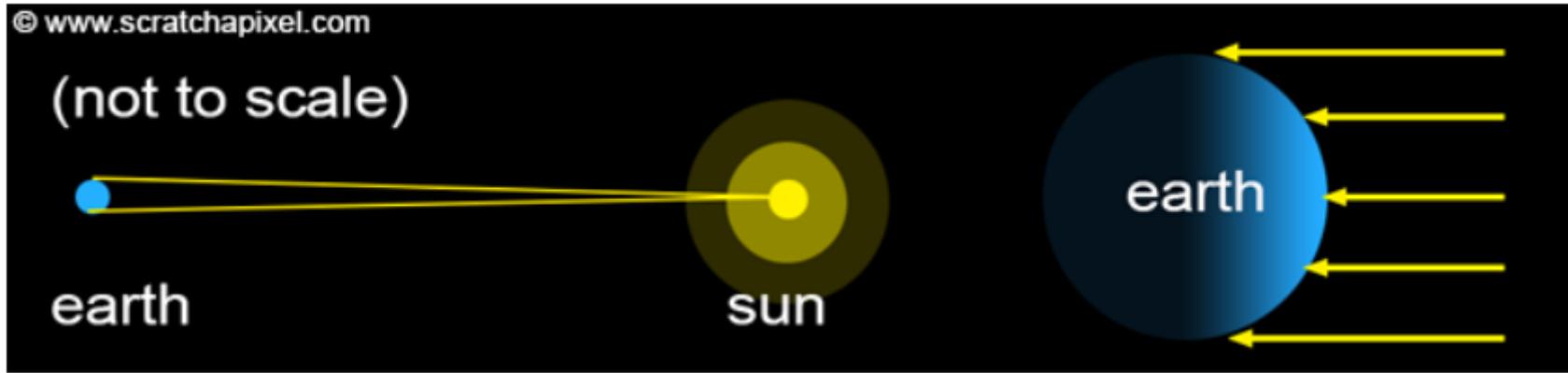
- Types:
  - Directional (distant) lights
  - Point (spherical, omnidirectional) lights

- Non-physical lights enable us to control light fall-off via which objects are illuminated, which objects cast shadows.
- For physically-based rendering, non-physical light should be avoided
  - Example: size of reflection of object by glossy or mirror-like surface depends on its size and distance to reflective surface. If light source has no shape nor size how reflection should look like? Hack: size parameter which is only used during shading.

TODO

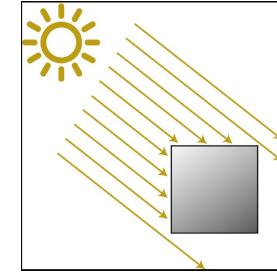
# Directional lights

- Directional - **distant** lights - considered so far from the objects in a 3D scene that they can be represented by parallel rays.
  - We only care for the **direction** of those parallel rays
  - Example of directional light: is Sun light on Earth.
    - Sun has spherical shape, but it is so far and Earth is so small compared to it that light rays reaching Earth can be considered parallel (small cone of directions – solid angle).
  - For scenes that cover small area of Earth's surface, Sun light can be assumed parallel.



# Directional lights

- Directional light is simplest model of light source.  
Parameters:
  - **Direction** – unit vector (not affected by position!)
  - **Color** – RGB vector in  $[0, 1]$
  - **Intensity** – float value in  $[0, \infty]$
  - Direction ( $\mathbf{l}$ ), color ( $\mathbf{c}$ ) and intensity ( $I$ ) are constant over 3D scene
- Intensity may be attenuated by shadowing and surface orientation
- Works well when light source distance is large compared to size of the 3D scene



[https://link.springer.com/chapter/10.1007/978-1-4842-4457-9\\_12](https://link.springer.com/chapter/10.1007/978-1-4842-4457-9_12)

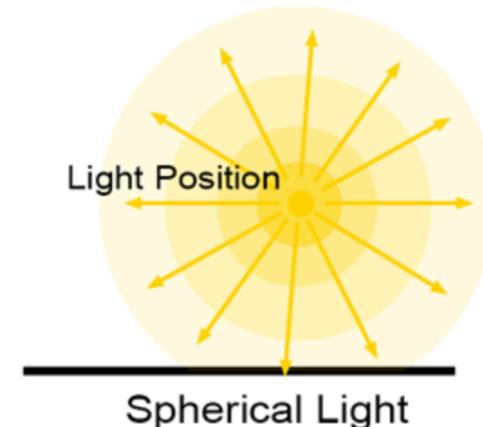


# Point lights

- Most common light sources in nature and around us are spherical
  - e.g., light bulbs or candle flame
  - Can be approximated as a collection of spherical light sources.
- Point light
  - Infinitesimally small in size → no shape
  - Light is emitted in all directions (omnidirectional, isotropic) therefore, no “direction” parameter
- Parameters
  - **Position** – vector determining position in world space (lights can be also transformed using 4x4 matrices)
  - **Color** – RGB vector in [0,1]
  - **Intensity** – float value in [0,inf]
  - Not affected by scale or rotation.



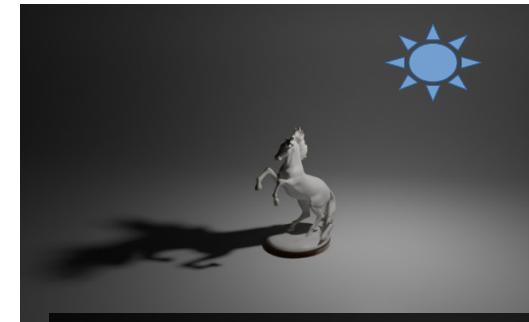
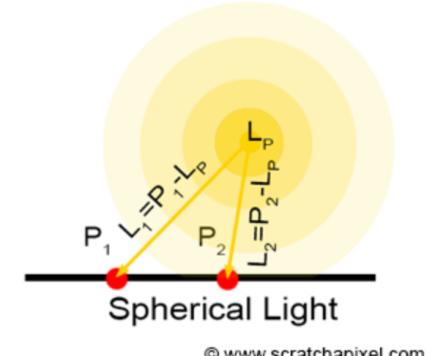
<https://polyhaven.com/models>



<https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-shading/shading-lights>

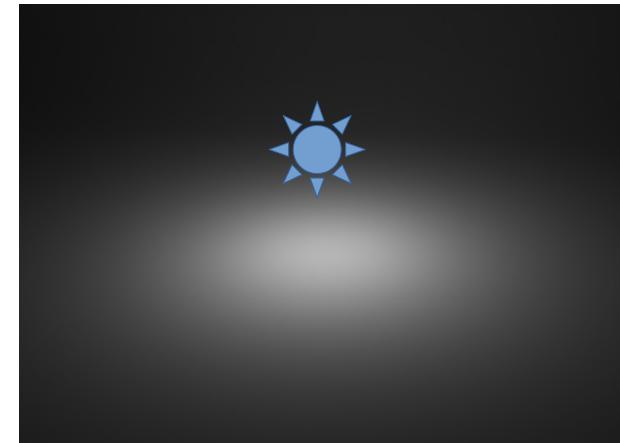
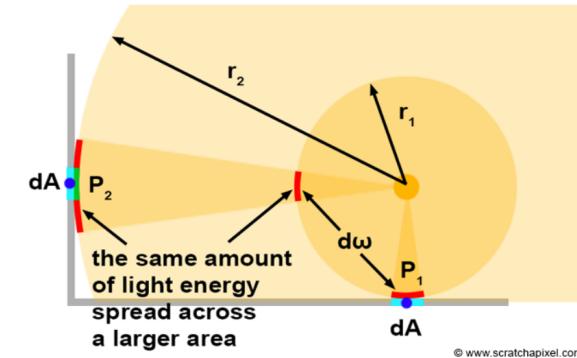
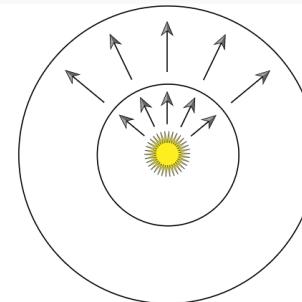
# Point lights

- Point light position determines:
  - **Direction of incoming light** ray for each surface position for each object in the scene
    - $\text{LightDirection} = \text{normalize}(P - L_p)$
  - **Distance** to the each surface position for each object in the scene – **light falloff**



# Point lights: direction and light falloff

- Line between point light and surface point (shading point):
  - Direction of light
  - Distance of point light
- Point light sources emit light radially
  - Energy of light source is redistributed over sphere
  - As the sphere keeps expanding in the space, energy becomes spread across much larger area → more distant objects in the scene receive less light – **light falloff**.
  - Distance of point light to surface point determines how much is that point illuminated → **intensity of point light varies as a function of distance**.



# Point light - light falloff

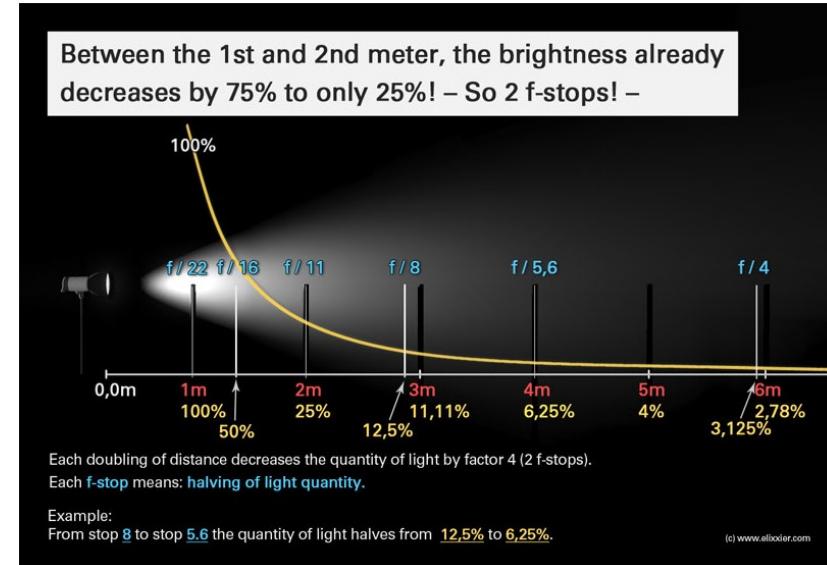
- Point light distributes energy spherically. Larger distance → larger sphere → smaller energy → attenuated light - **inverse square law falloff**:

$$L_i = \frac{\text{light intensity} * \text{light color}}{4\pi r^2}$$

- Problems:
  - **Small distances**: as  $r \rightarrow 0$ , light intensity  $\rightarrow$  infinity. Fix:

$$L_i = \frac{\text{light intensity} * \text{light color}}{4\pi(r + \text{epsilon})^2}$$

- **Large distance**: for efficient rendering, it is desired to light reach 0 at some point. Otherwise, light calculation should be done no matter how far the camera is from the light
- Light falloff function can be any other function of distance
  - For stylized appearance different functions may be more interesting



<https://petapixel.com/inverse-square-law-light/>

[https://www.youtube.com/watch?v=Z8AAX-ENWvQ&t=2s&ab\\_channel=Blender](https://www.youtube.com/watch?v=Z8AAX-ENWvQ&t=2s&ab_channel=Blender)



# Point light - light falloff



[https://www.youtube.com/watch?v=Z8AAX-ENWvQ&t=2s&ab\\_channel=Blender](https://www.youtube.com/watch?v=Z8AAX-ENWvQ&t=2s&ab_channel=Blender)



Toy Story 4 (2021)

# Point lights: directional light falloff

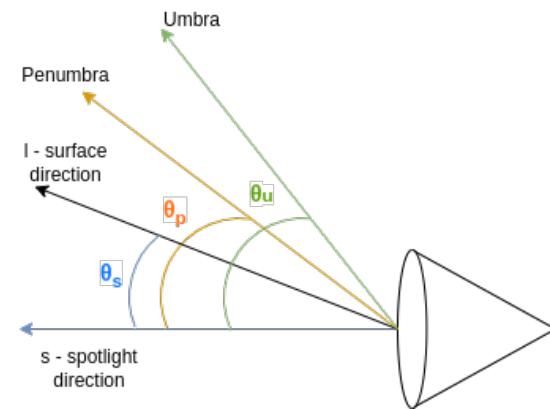
- Next to distance light falloff, real-world illumination intensity also **varies by direction**, generally:

$$L_i = (\text{light intensity} * \text{light color}) f_{distance}(r) f_{direction}(l)$$

- Different choices of distance and directional light falloff functions produce different light sources:
  - **Spotlights** - emit light in a cone of directions from their position

# Spotlights

- Projects light in circular cone.
- Directional falloff function is rotational symmetric around a spotlight direction  $s$
- Parameters:
  - Angle of umbra  $\theta_u$ 
    - Outer cone shell
  - Angle of penumbra  $\theta_p$ 
    - Inner cone where the light is at full intensity
  - Color: RGB vector in  $[0,1]$
  - Intensity: float in  $[0, \infty]$

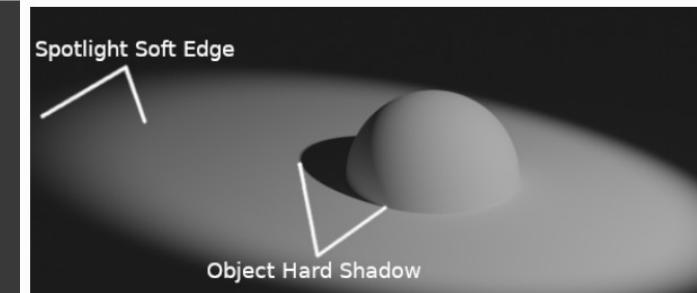
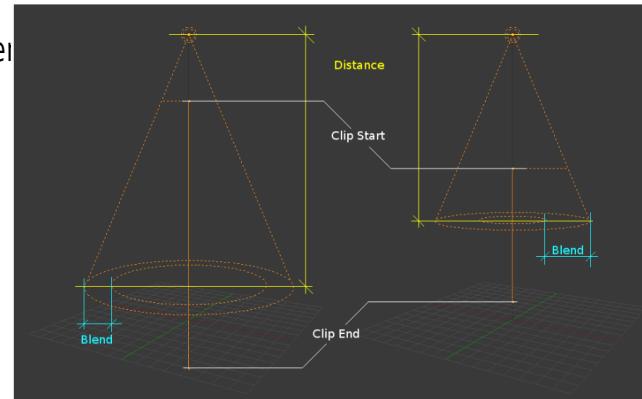


$$L_i = (\text{light intensity} * \text{light color}) f_{distance}(r) f_{direction}(l)$$

$$t = \text{clamp}(0, 1, \frac{\cos(\theta_s) - \cos(\theta_u)}{\cos(\theta_p) - \cos(\theta_u)})$$

[https://seblagarde.files.wordpress.com/2015/07/course\\_notes\\_moving\\_frostbite\\_to\\_pbr\\_v32.pdf](https://seblagarde.files.wordpress.com/2015/07/course_notes_moving_frostbite_to_pbr_v32.pdf)

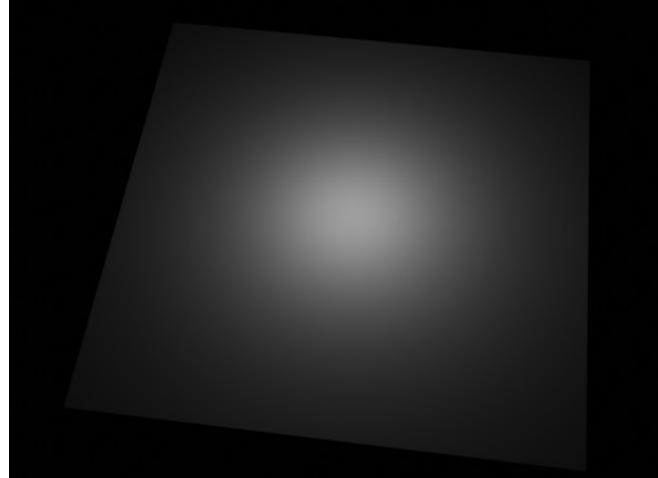
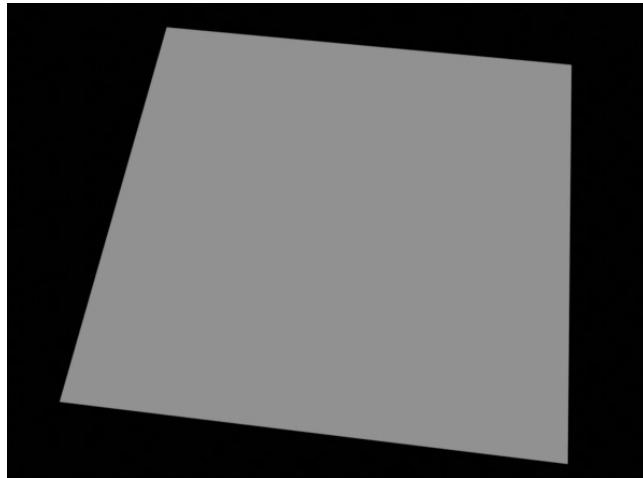
$$f_{directional}(l) = t^2$$



[https://docs.blender.org/manual/en/latest/render/lights/light\\_objects.html](https://docs.blender.org/manual/en/latest/render/lights/light_objects.html)

# Non-physical lights recap

- Directional, point, spot light



# Multiple lights

- Contribution of light adds up linearly
  - Required for photorealistic rendering where scene contains multiple lights
  - Required for Compositing rendered images: adding multiple rendered images where each has contribution from one light

$$L_o(p, v) = \int_{l \in \omega} f(l, n, v) L_i(p, l)(n \cdot l) dl$$

$$L_o(p, v) = \sum_{i=1}^n f(l_i, n, v) c_{light_i}(n \cdot l_i)$$



# Other distance and directional light falloff functions

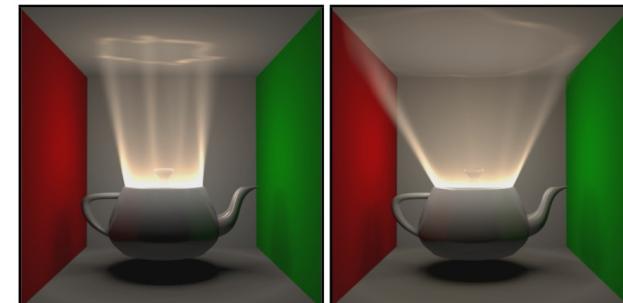
- **Textured lights**

- Use of **texture** for visual richness of light sources: intensity, color and directional/distance variations
- For **cone-type lights**: projective textures (slide projector effect, gobo/cookie lights)
- For **omni-directional-type lights**: texture for distance falloff – attenuation maps (e.g., light beams)

$$L_i = (\text{light intensity} * \text{light color}) f_{distance}(r) f_{direction}(l)$$



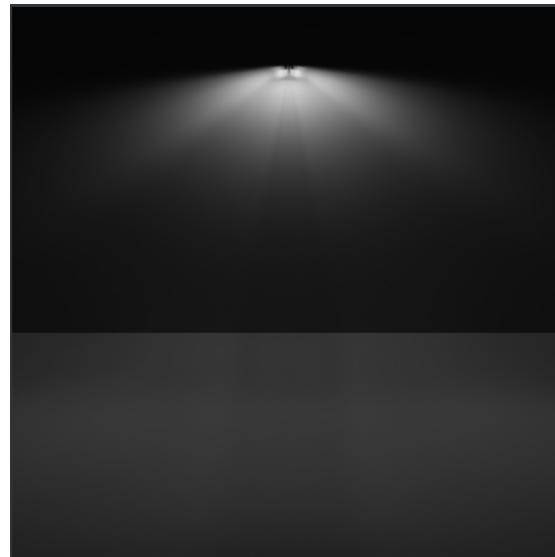
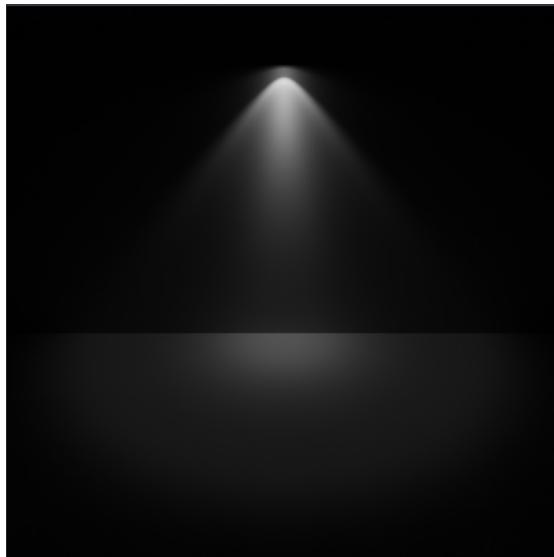
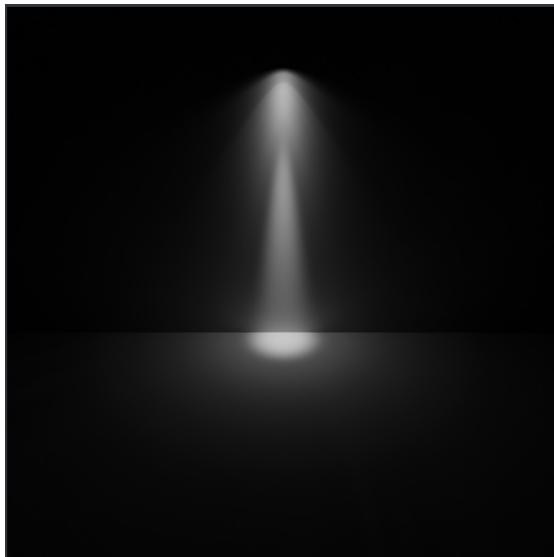
[https://developer.valvesoftware.com/wiki/Env\\_projectedtexture](https://developer.valvesoftware.com/wiki/Env_projectedtexture)



[https://www.researchgate.net/publication/220183756\\_A\\_Programmable\\_System\\_for\\_Artistic\\_Volumetric\\_Lighting](https://www.researchgate.net/publication/220183756_A_Programmable_System_for_Artistic_Volumetric_Lighting)

# Other distance and directional light falloff functions

- Tabulated patterns
  - Standard: (illuminating engineering society) **IES profiles**
  - Measured from the real world using: Goniophotometer



IES profiles: <https://ieslibrary.com/en/home>

# Environment illumination

- TODO
  - Cube maps

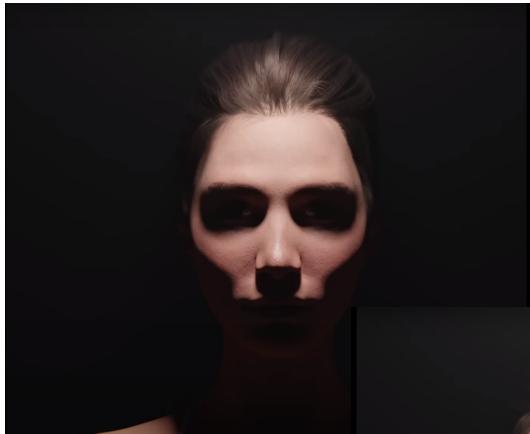
# Practical tip: modeling with lights

- Non-physical lights can be placed to simulate illumination
  - TODO

TODO

# Practical tip: modeling with lights

- Placement and orientation of light greatly determines object appearance

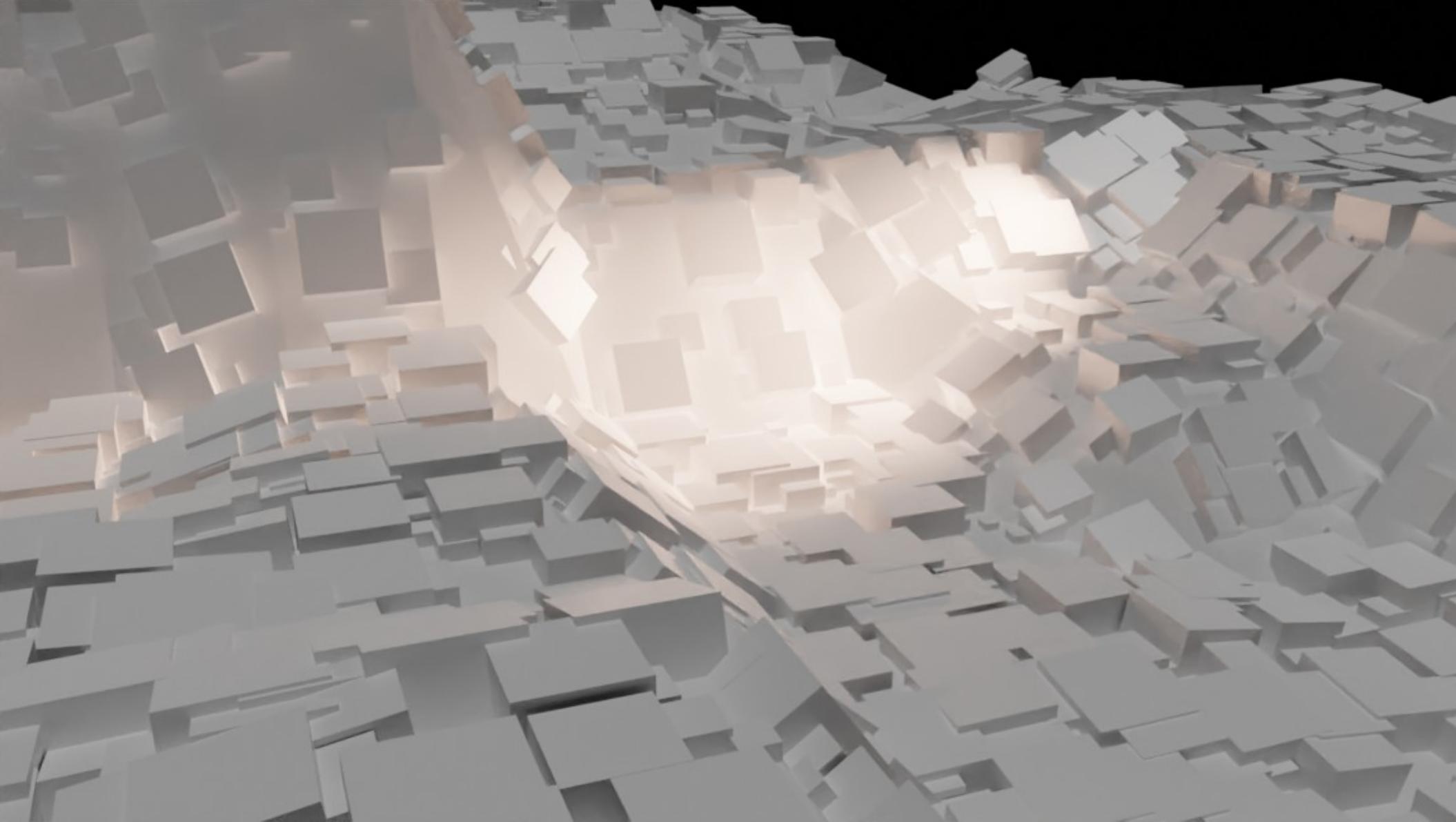


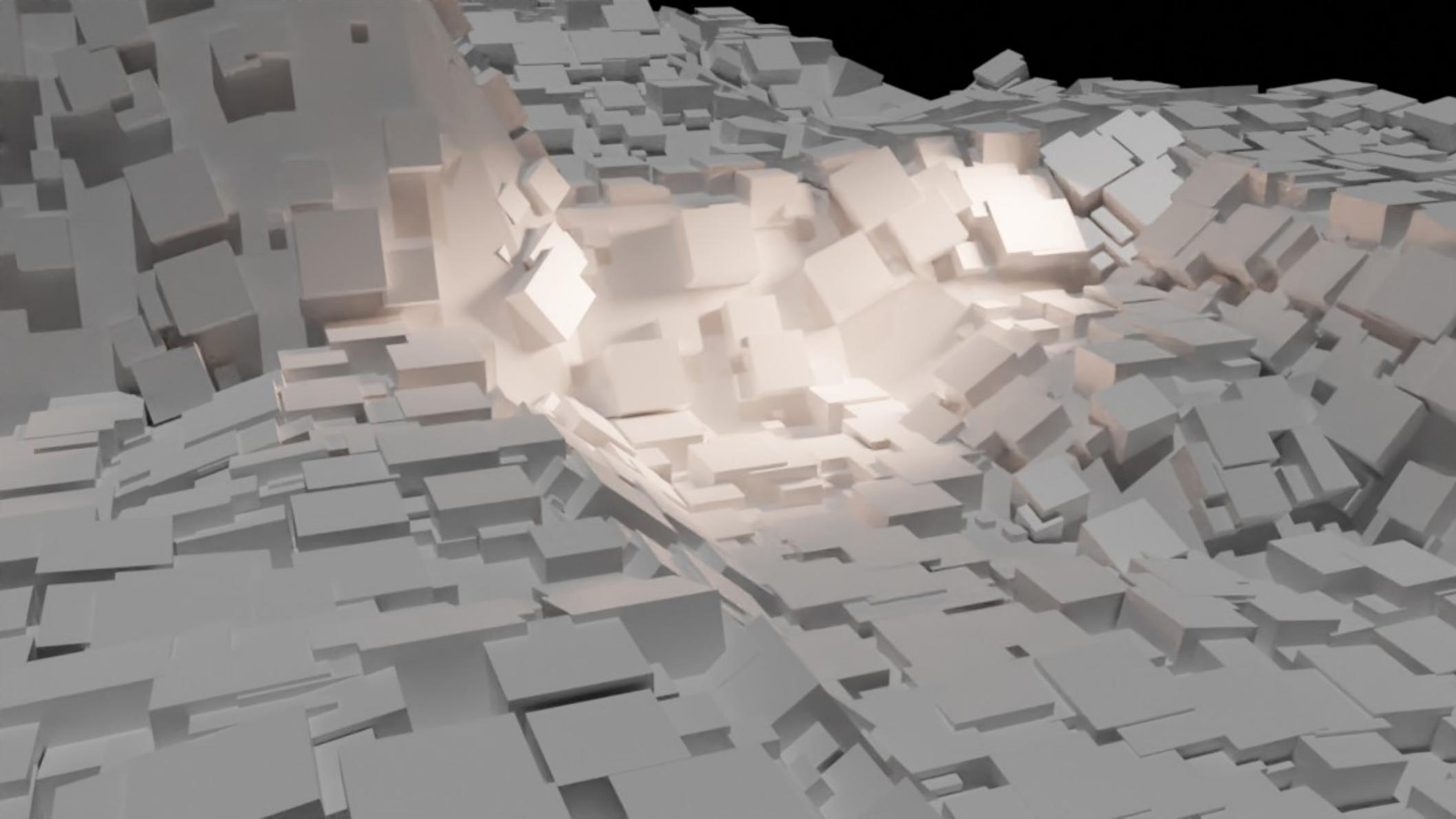
[https://www.youtube.com/watch?v=Ys4793edotw&t=1s&ab\\_channel=BlenderGuru](https://www.youtube.com/watch?v=Ys4793edotw&t=1s&ab_channel=BlenderGuru)

# More into light

- Lighting Artstation: <https://www.artstation.com/channels/lighting>
- Lighting artist: <https://80.lv/articles/amazing-lighting-for-simple-scenes/>
- Lighting for artists: [https://www.youtube.com/watch?v=Ys4793edotw&ab\\_channel=BlenderGuru](https://www.youtube.com/watch?v=Ys4793edotw&ab_channel=BlenderGuru)

# Shadows







Scene from:

<https://casual-effects.com/data/index.html>



# Light and shadow

- Light falling on object causes object to cast shadow – blocking light to fall on another surface
  - Occluders: objects casting shadow
  - Receivers: objects on which the shadow is cast
- Shadows are important for:
  - Understanding relation of objects to one another
  - Realistic image synthesis: needed for understanding details and surface characteristics



<https://creativecoding.soe.ucsc.edu/courses/cs488/reportsA/shadows.pdf>



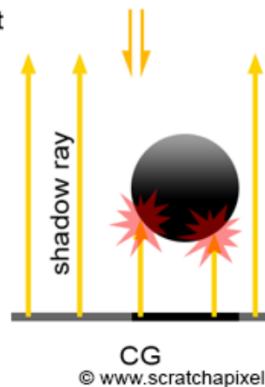
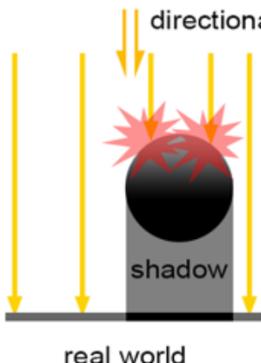
<https://lumion.com/blog/feature-spotlight-sky-light-soft-shadows-and.html>

# Rendering and shadow

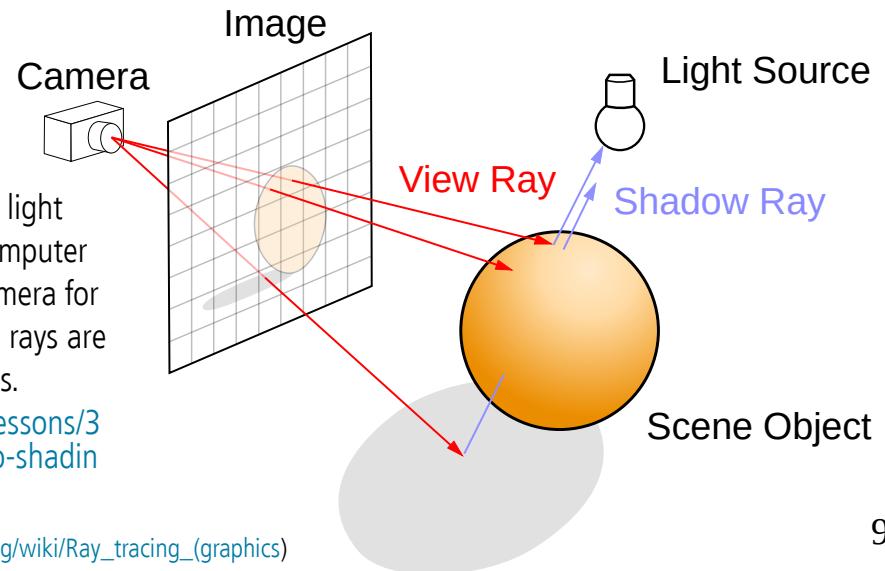
- Simulating shadow depends on rendering algorithm
  - Specifically: algorithm for solving the visibility problem
- In **ray-tracing based rendering**, shadows can be calculated during the main image is being rendered
- In **rasterization-based rendering** multi-pass rendering is required. Example:
  - One pass is needed to obtain visibility of objects looking from the position of light – **shadow map** – for each light in 3D scene
  - Shadow map is used in final pass which renders final image

# Shadow in ray-tracing

- For surface point for which color is being calculated (shading), ray is casted towards light source – **shadow ray**
  - If shadow ray intersects object on its way, then shaded surface point is in shadow
  - If object is opaque, shadow is dark
  - If object is transparent, shadow is lighter

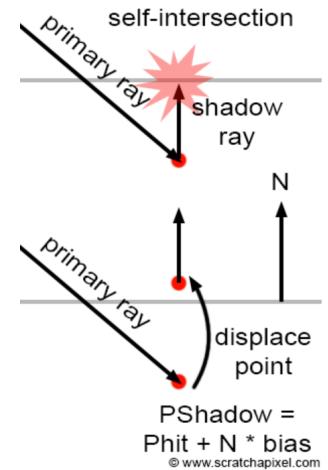
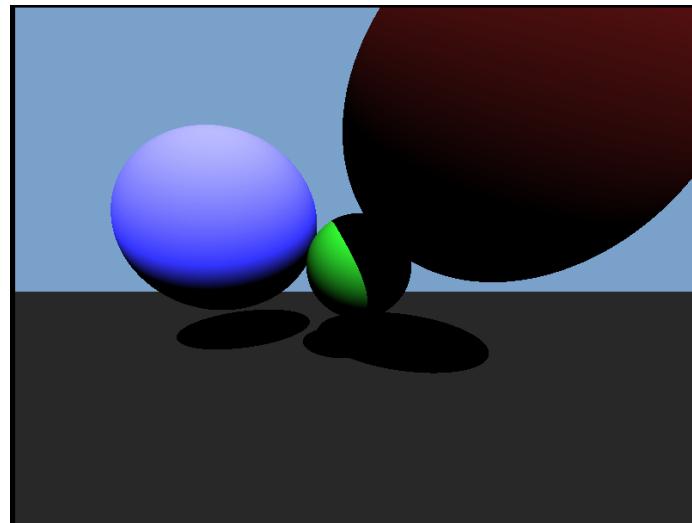
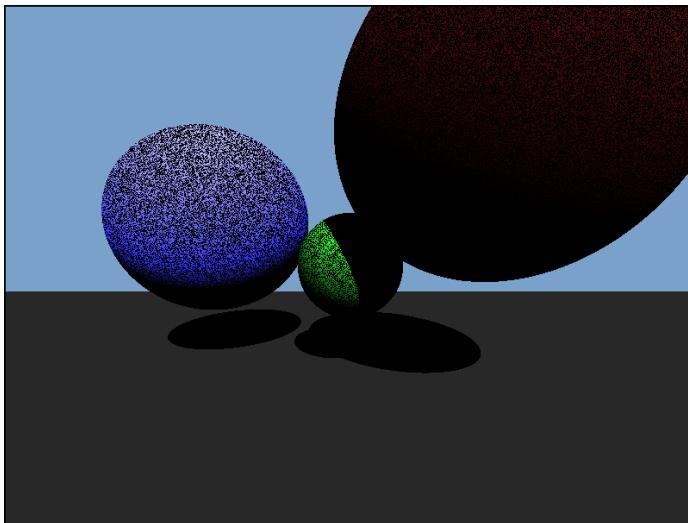


In real world, shadow is result of light being obstructed by object. In computer graphics, rays are traced from camera for intersections and then additional rays are sent to check if shadowing occurs.  
<https://www.scratchapixel.com/lessons/3-d-basic-rendering/introduction-to-shading/light-and-shadows>



# Shadow in ray-tracing: shadow acne

- Due to numerical limits in precision, intersection point might end up under the surface
  - Casting shadow ray from this point causes intersection and thus light obstruction
  - When surface is rendered, some points are light and some are dark → shadow acne
  - Solutions:
    - Use double float precision
    - Add small displacement to intersection point in direction of normal: **shadow bias** parameter

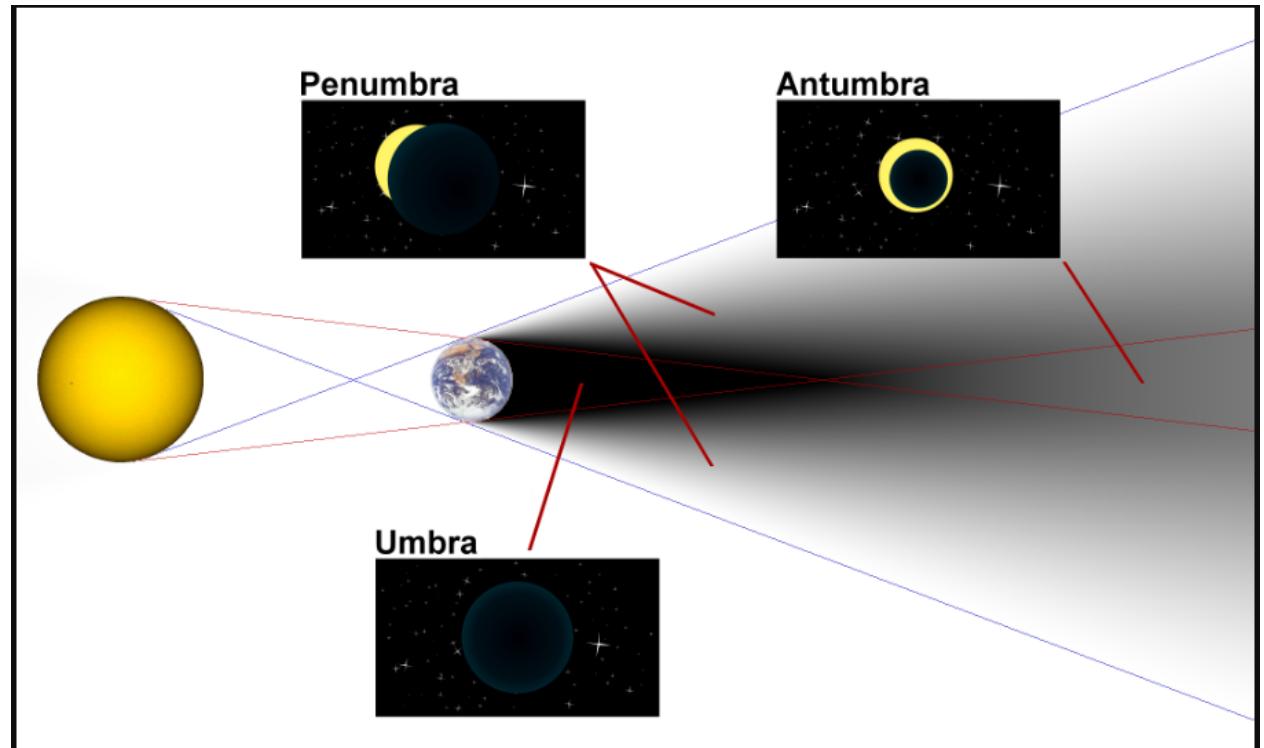


<https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-shading/light-and-shadows>

<https://bheisler.github.io/post/writing-raytracer-in-rust-part-2/>

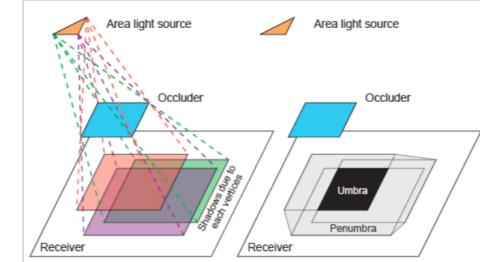
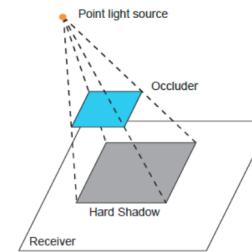
# Elements of a shadow

- Penumbra
  - Partially shadowed region
- Umbra
  - Fully shadowed region

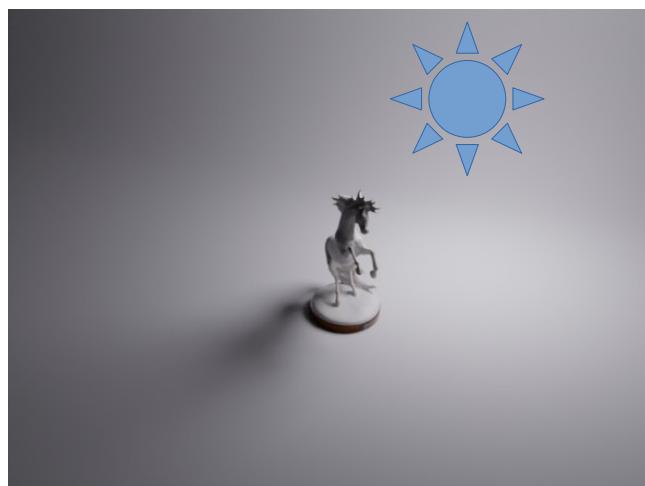


# Sharp vs smooth shadows

- Size of light determines shadow smoothness
  - Softer shadow edges are more realistic, but more expensive to render
  - Small light sources (e.g., point lights) → hard shadows; only umbra component
  - Large light sources (e.g, physical lights) → soft shadows; both umbra and penumbra component



<https://creativecoding.soe.ucsc.edu/courses/cs488/reportsA/shadows.pdf>



# Light and Shadow: Tips and Tricks

- Non-physical light sources as well as shadows they cause can be efficiently
- For real-time rendering, full physically based simulation of light is not feasible
- When creating 3D scene, (environment) artists understand the how the scene should be illuminated given some main sources of light. Using this knowledge they place non-physical light to fake actual light flow through the scene
- <https://80.lv/articles/dishonored-interiors-lighting-props/>

TODO

# To remember

- Light and color
  - PSD
  - Colorspaces (XYZ, RGB) and gamut
  - Rendering with RGB
- Light model
  - Geometric optics
  - Directional, points and spot lights
- Shadow
  - Rasterization vs raytracing
  - Shadow elements

# Literature

- <https://github.com/lorentzo/IntroductionToComputerGraphics/wiki/Foundations-of-3D-scene-modeling>