VSB TECHNICAL
UNIVERSITY
OF OSTRAVA

FACULTY OF ELECTRICAL
ENGINEERING AND COMPUTER
SCIENCE

DEPARTMENT
OF COMPUTER
SCIENCE

Computer Graphics I

460-4078

Fall 2024

Last update 3. 10. 2024

Radiometry

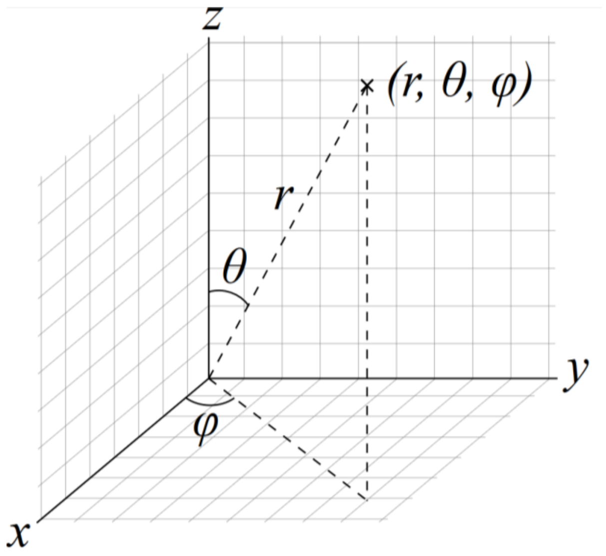
- Radiometry is a set of techniques for measuring electromagnetic radiation, including visible light.
- Photometry is the science of the measurement of light, in terms of its perceived brightness to the human eye.

Representation of Direction in 3D

- Cartesian coordinates

$$\hat{\boldsymbol{d}} = \left(d_x, d_y, d_z\right), \left\|\hat{\boldsymbol{d}}\right\| = 1$$

- Spherical coordinates (physics and ISO convention)
  - $\omega = (\theta, \varphi)$
  - Polar angle (theta) $\theta = \cos^{-1} d_z \in \langle 0, \pi \rangle$
  - Azimuthal angle (phi) $\varphi = \tan^{-1} \frac{d_y}{d_x} \in \langle 0, 2\pi)$

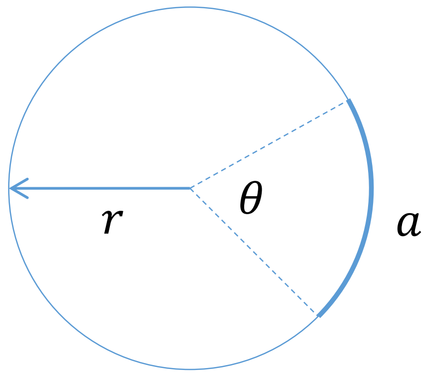


$$d_x = \sin\theta \cos\varphi$$
$$d_y = \sin\theta \sin\varphi$$
$$d_z = \cos\theta$$

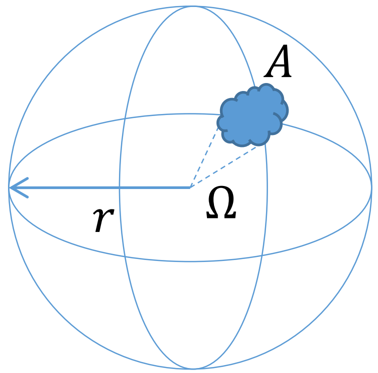
In C/C++, we can compute the azimuthal angle as $\varphi = \texttt{atan2f}\left(d_y, d_x\right) + \begin{cases} 2\pi & d_y < 0 \\ 0 & \text{otherwise} \end{cases}$

Solid Angle

- Planar angle (e.g. θ and φ)
  - Arc length on a unit sphere
  - A full circle has 2π radians (rad) (= length of unit circle)
  - $\theta = \frac{a}{r}$



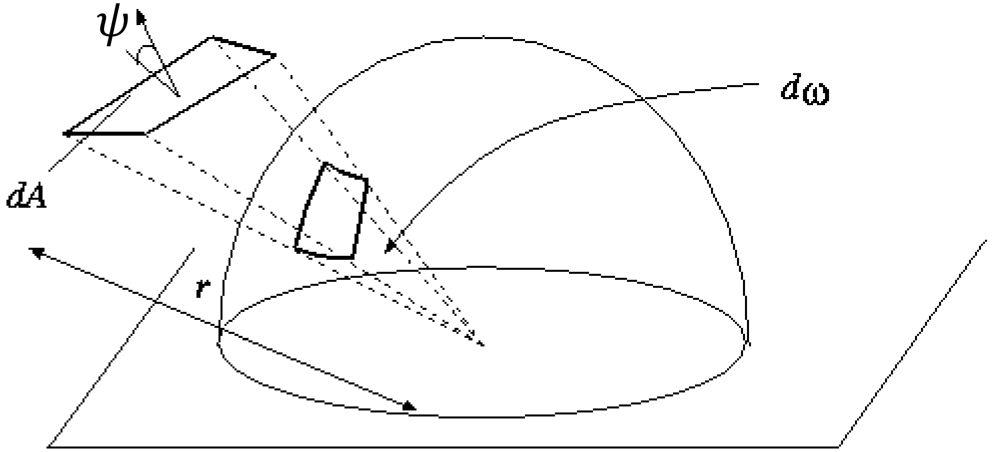
- Solid angle (Ω)
  - Surface area on an unit sphere
  - Full sphere has 4π steradians (sr) (= area of unit sphere)
  - $\Omega = \frac{A}{r^2}$



Differential Solid Angle

- Inifinitesimaly small solid angle around a given direction
- Magnitude $\mathrm{d}\omega$ represents size of a differential area on the unit sphere
- Direction $\boldsymbol{\omega}$ is the center of the projection of the differential area $\mathrm{d}A$ on the unit sphere

  - $\mathrm{d}\omega = \frac{\cos\psi\,\mathrm{d}A}{r^2}$



Differential Solid Angle

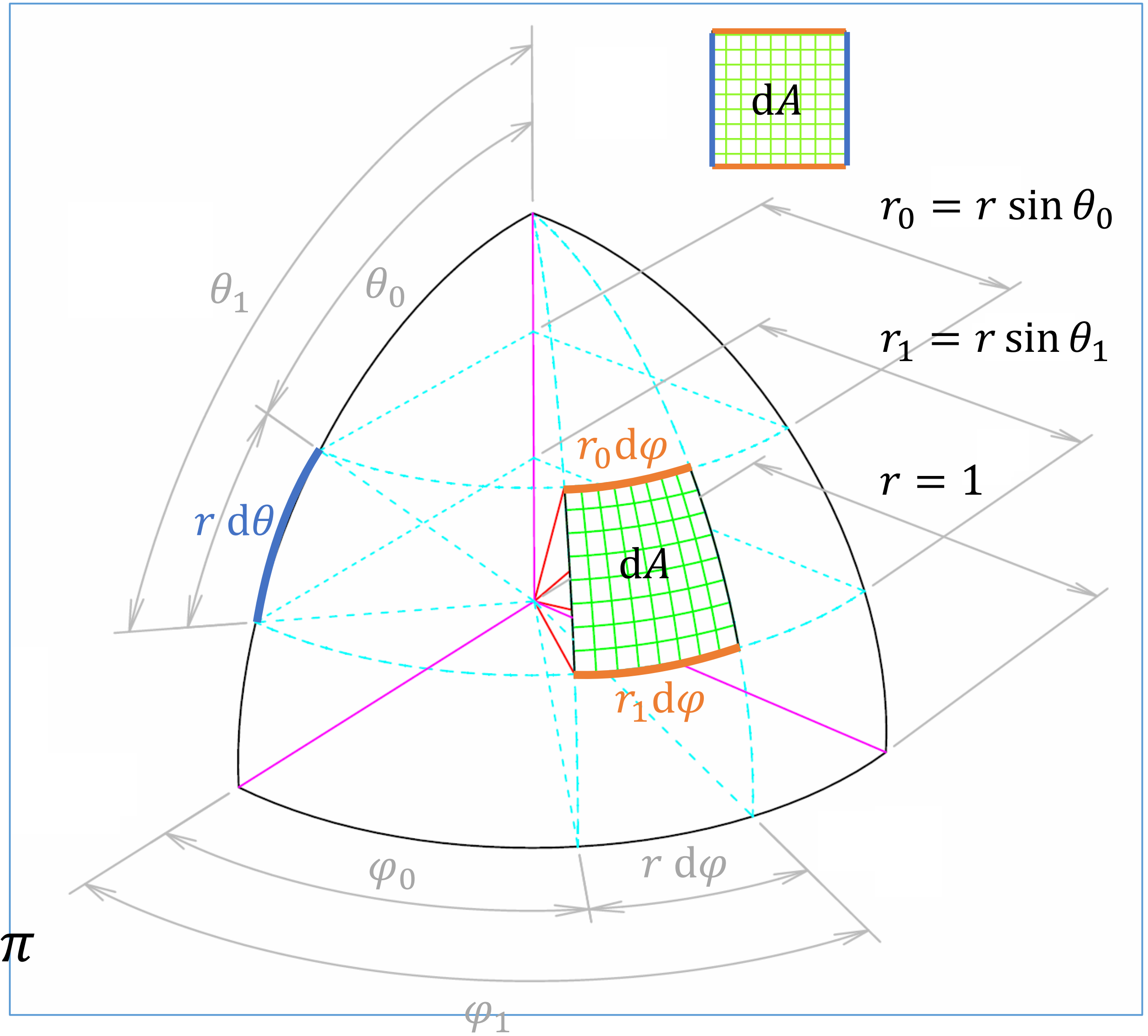
- $\mathrm{d}\omega = \frac{\mathrm{d}A}{r^2}$
- As $\mathrm{d}\theta \to 0$ and $\mathrm{d}\varphi \to 0$

$$\mathrm{d}A = (r\,\mathrm{d}\theta)(r_{\{0,1\}}\mathrm{d}\varphi)$$
$$= (r\,\mathrm{d}\theta)(r\sin\theta_{\{0,1\}}\,\mathrm{d}\varphi)$$

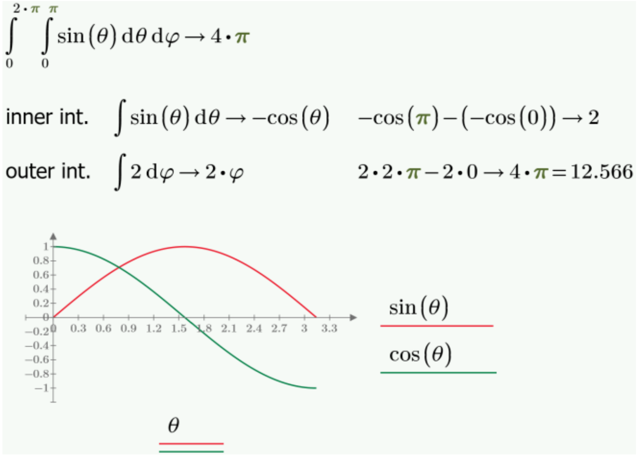
- For unit sphere Ω ($r = 1$) holds that

$$\mathrm{d}\omega = \mathrm{d}A = \sin\theta\,\mathrm{d}\theta\mathrm{d}\varphi$$

- $\int_\Omega \mathrm{d}\omega = \int_{\varphi=0}^{2\pi}\int_{\theta=0}^{\pi}\sin\theta\,\mathrm{d}\theta\mathrm{d}\varphi = 4\pi$



Differential Solid Angle



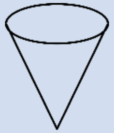
$$\int_0^{2\cdot\pi}\int_0^{\pi}\sin(\theta)\,\mathrm{d}\theta\,\mathrm{d}\varphi \rightarrow 4\cdot\pi$$

inner int. $\int \sin(\theta)\,\mathrm{d}\theta \rightarrow -\cos(\theta)$ $\quad -\cos(\pi)-(-\cos(0)) \rightarrow 2$

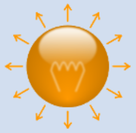
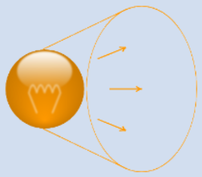
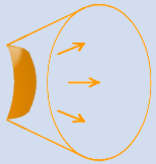
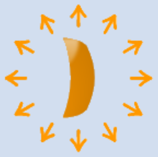
outer int. $\int 2\,\mathrm{d}\varphi \rightarrow 2\cdot\varphi$ $\quad 2\cdot 2\cdot\pi - 2\cdot 0 \rightarrow 4\cdot\pi = 12.566$

```python
from sympy import *

# spherical coordinates (physics and ISO convention)
t = Symbol('t') # theta <0, pi> (polar angle)
p = Symbol('p') # phi <0, 2*pi) (azimuthal angle)

# integral of domega across the sphere
I = integrate(sin(t), (t, 0, pi), (p, 0, 2*pi))
print(I)

Out: 4*pi
```

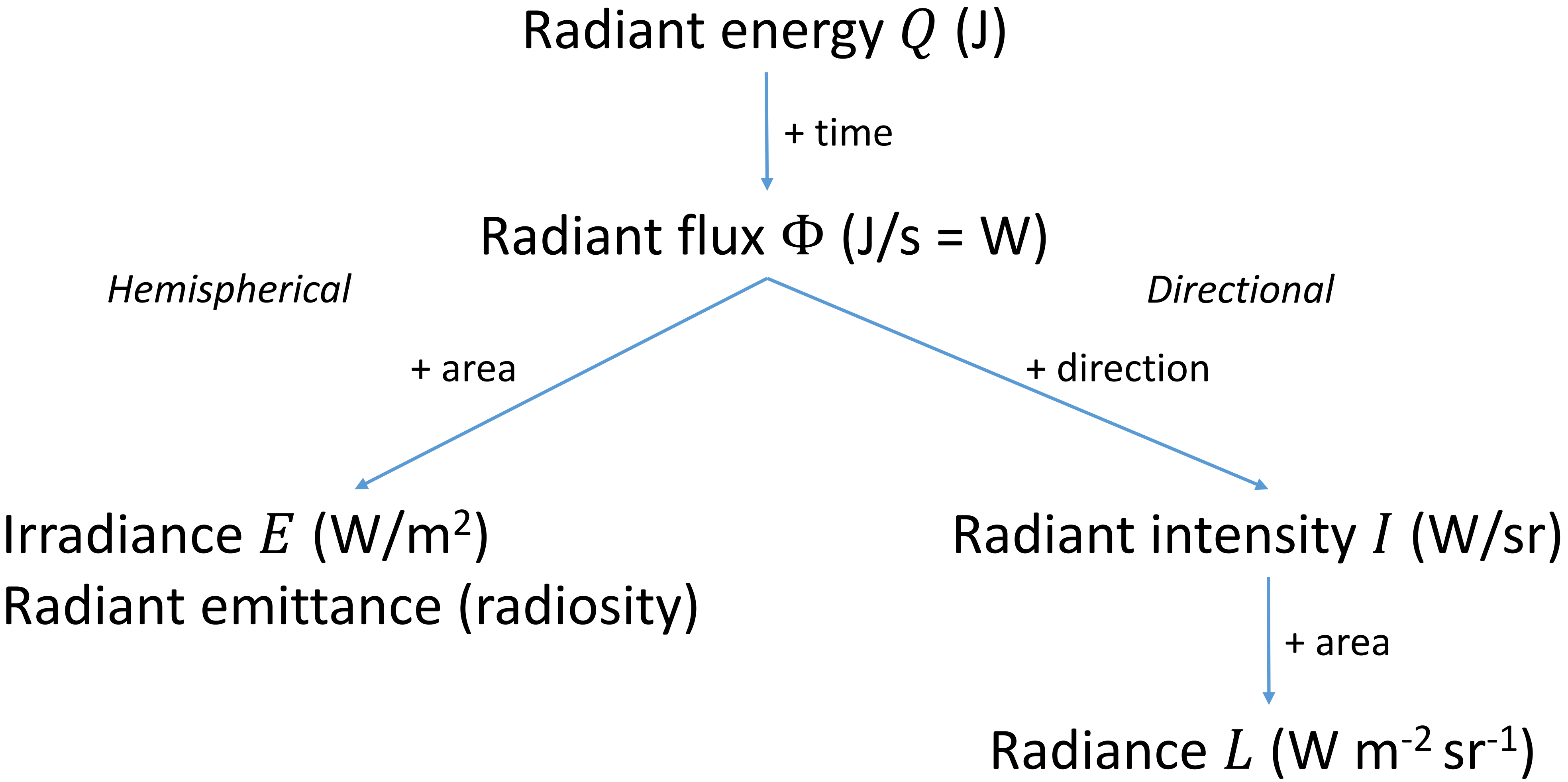
Used Symbols

Representation	Symbol	Meaning
•	x	Point on the real axis
—	$\mathrm{d}x$	Infinitesimally small distance on the real axis
↗	$\omega \equiv \hat{\boldsymbol{d}} = \left(d_x, d_y, d_z\right)^T$	Unit direction
┃	$\mathrm{d}\omega$	Differential solid angle (ray)
	Ω	Solid angle
■	A	Area
•	$\mathrm{d}A$	Differential area

SI Radiometry and Photometry Units

Quantity	Veličina	Definition	Unit	Illustration
Radiant flux	Zářivý tok	$\Phi = \frac{\mathrm{d}Q}{\mathrm{d}t}$	$W = J/s$ (watt)	
Luminous flux	Světelný tok		$lm = cd \cdot sr$ (lumen)	
Radiant intensity	Zářivost	$I = \frac{\mathrm{d}\Phi}{\mathrm{d}\Omega}$	W/sr	
Luminous intensity	Svítivost		$cd = lm/sr$ (candela)	
Radiance	Zář	$L = \frac{\mathrm{d}^2\Phi}{\cos(\theta)\,\mathrm{d}A\,\mathrm{d}\Omega}$	$W/(m^2 \cdot sr)$	
Luminance	Jas		$lm/(m^2 \cdot sr) = cd/m^2$	
Irradiance	Intenzita (o)záření (ozářenost)	$E = \frac{\mathrm{d}\Phi}{\mathrm{d}A}$	W/m^2	
Illuminance	Intenzita osvětlení (osvětlenost)		$lm/m^2 = (cd \cdot sr)/m^2$	

Radiometry



Radiant energy Q (J)

+ time

Radiant flux Φ (J/s = W)

Hemispherical

Directional

+ area

+ direction

Irradiance E (W/m^2)

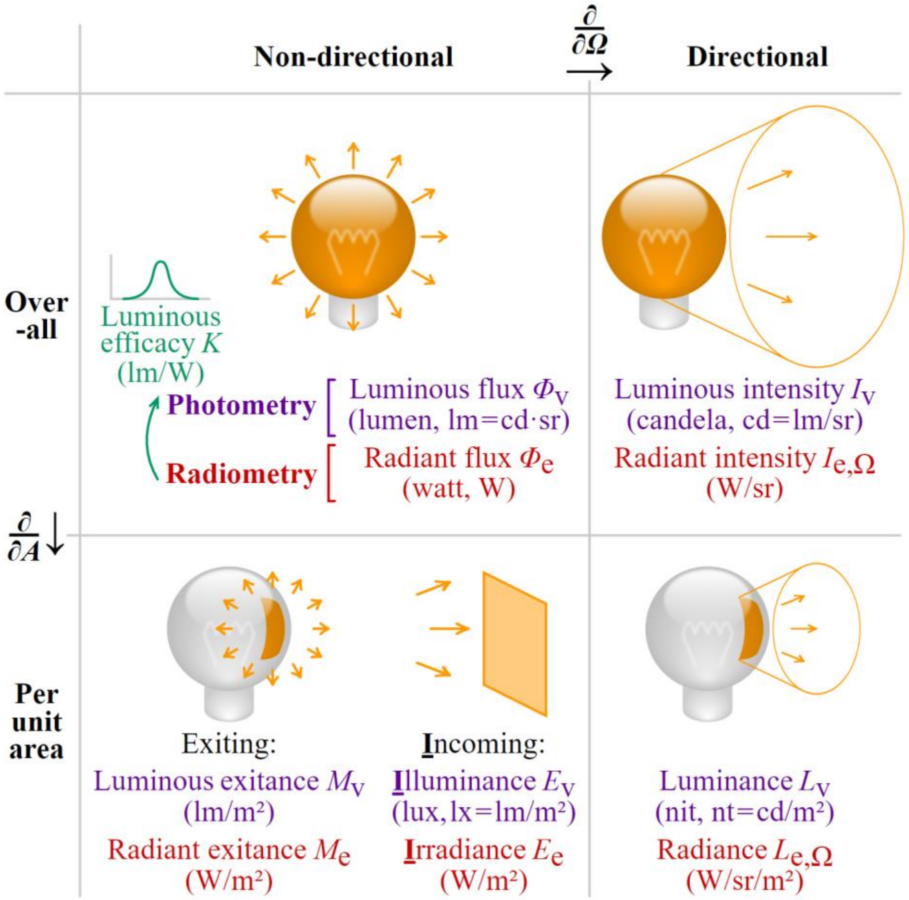
Radiant emittance (radiosity)

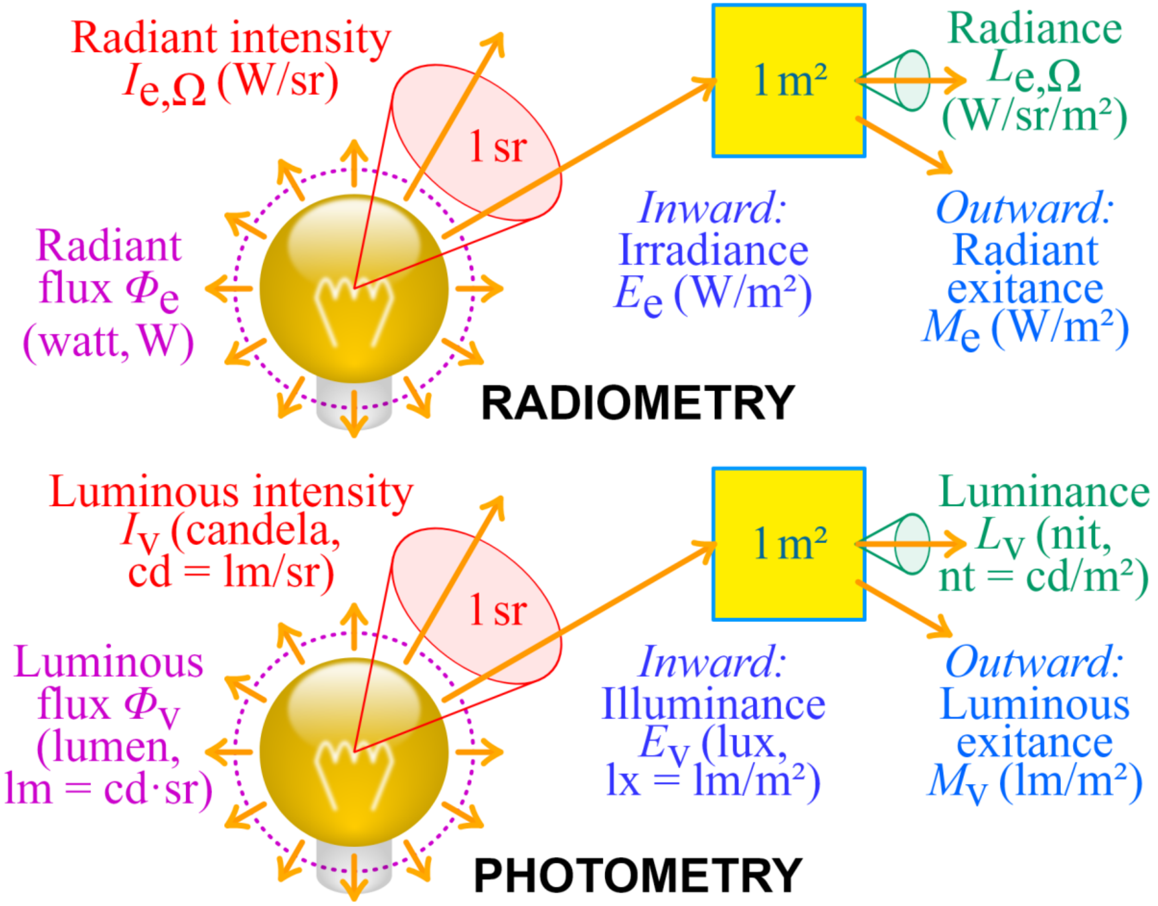
Radiant intensity I (W/sr)

+ area

Radiance L (W m^{-2} sr^{-1})

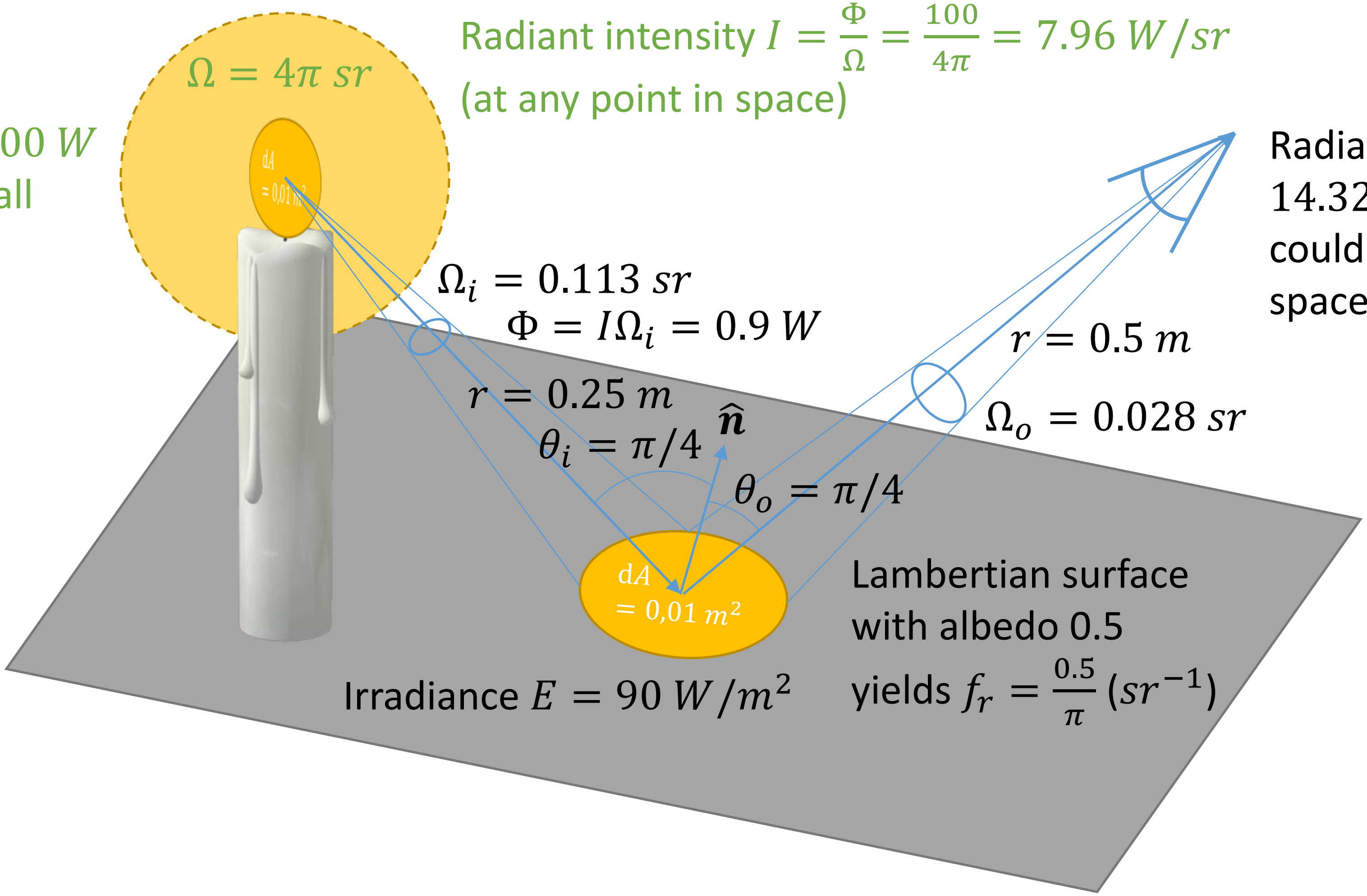
Radiometry Units





Source: https://commons.wikimedia.org/wiki/File:Photometry_radiometry_units.svg

Example (Radiometric Quantities)



Total radiant flux $\Phi = 100\ W$ emitted isotropically in all directions

$\Omega = 4\pi\ sr$

Radiant intensity $I = \frac{\Phi}{\Omega} = \frac{100}{4\pi} = 7.96\ W/sr$ (at any point in space)

Radiance $L = f_r E = 14.324\ W\ m^{-2} sr^{-1}$ (eye could be at any point in space)

$\Omega_i = 0.113\ sr$

$\Phi = I\Omega_i = 0.9\ W$

$r = 0.25\ m$

$\theta_i = \pi/4$

$\hat{\boldsymbol{n}}$

$\theta_o = \pi/4$

$r = 0.5\ m$

$\Omega_o = 0.028\ sr$

Lambertian surface with albedo 0.5 yields $f_r = \frac{0.5}{\pi}\ (sr^{-1})$

Irradiance $E = 90\ W/m^2$

Radiant Flux (W, lm)

- Radiant flux (also flux or power) Φ is the quantity of energy emitted (or received) by an object per unit of time in all directions
- The rate (speed) of electrical energy conversion to light energy.

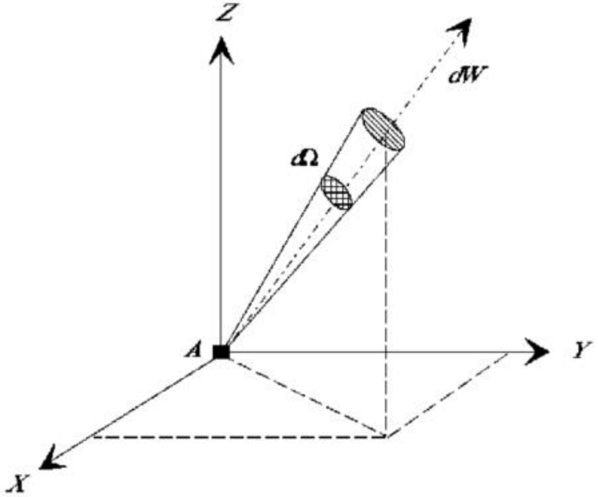
$$\Phi = \frac{\mathrm{d}Q}{\mathrm{d}t} = \int_A \int_\Omega L(\boldsymbol{x}, \omega) \cos\theta \, \mathrm{d}\omega \, \mathrm{d}A$$

- For scenes in steady state holds that terms energy and radiant flux may be used interchangeably

Radiant Intensity (W sr^{-1})

- We can define the radiant intensity I as the radiant flux $\mathrm{d}\Phi$ emitted by a point light source per solid angle $\mathrm{d}\Omega$

$$I = \frac{\mathrm{d}\Phi}{\mathrm{d}\Omega}$$

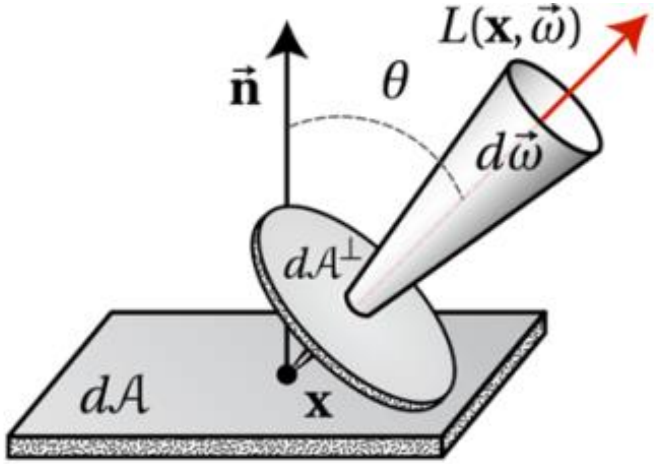


- Describes the directional distribution of light
- Meaningful only for point light sources

Radiance (W m^{-2} sr^{-1})

- Radiance is defined as the radiant flux $\mathrm{d}\Phi$ travelling at some point in a specified direction ω, per unit area $\mathrm{d}A^{\perp}$ perpendicular to a direction of travel, per solid angle $\mathrm{d}\omega$

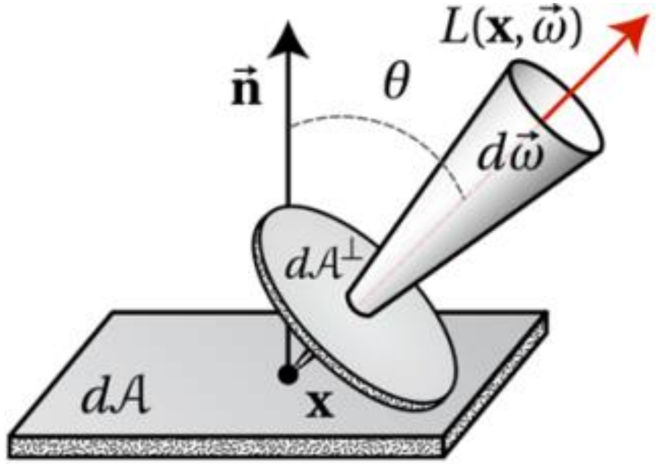
$$L(\boldsymbol{x}, \omega) = \frac{\mathrm{d}^2\Phi}{\cos\theta \, \mathrm{d}A \, \mathrm{d}\omega}$$



Radiance (W m^{-2} sr^{-1})

- Radiance represents the response of eye or sensor
- Radiance is independent of viewing distance
- Radiance measures the reflected sum of the lights' irradiance onto point $\boldsymbol{x}$ as viewed from ω
- Radiance is the irradiance on a surface normal to the beam per solid angle traveling in particular direction

$$L(\boldsymbol{x}, \omega) = \frac{\mathrm{d}E_{\perp}}{\mathrm{d}\omega} = \frac{\mathrm{d}^2\Phi}{\cos\theta\,\mathrm{d}A\,\mathrm{d}\omega}$$

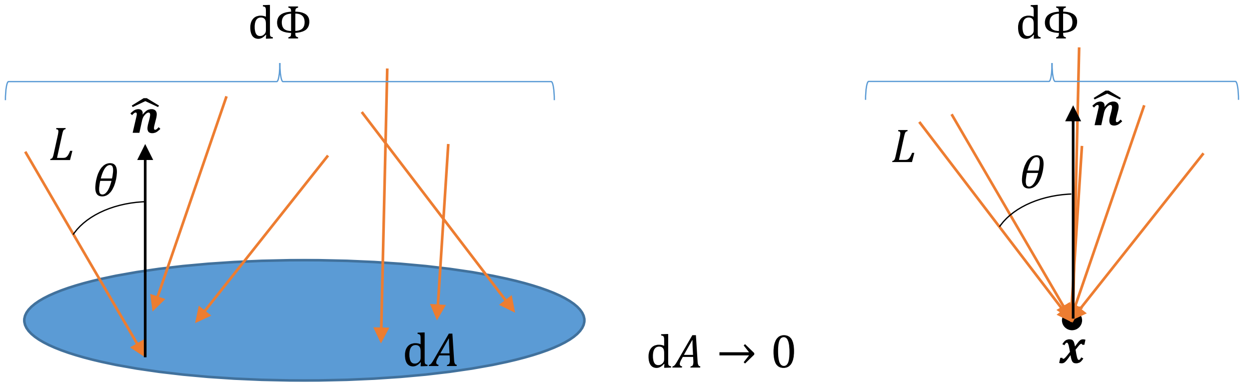


Irradiance, Exitance (or Radiant Flux Density) ($\mathrm{W\ m^{-2}}$, lx)

$$E = \frac{I\,\mathrm{d}\omega}{dA} = \frac{I\,\frac{\cos(\theta)\,dA}{r^2}}{dA} = \frac{I\cos(\theta)}{r^2}$$

- Radiant flux Φ per unit area, or radiant flux density

$$E(\boldsymbol{x}) = \frac{\mathrm{d}\Phi}{\mathrm{d}A}; \frac{\Phi}{A} = \frac{1}{A}\int_A \int_\Omega L(A,\omega)\cos\theta\,\mathrm{d}\omega\,\mathrm{d}A = \int_\Omega L(\boldsymbol{x},\omega)\cos\theta\,\mathrm{d}\omega$$



Side note: Irradiance measurements should be made facing the source, if possible. The irradiance will vary with respect to the cosine of the angle between the optical axis and the normal to the detector

- If we just reverse the light direction we get radiant emittance or radiant exitance

Why is Radiance Constant Along Ray?

- Energy conservation law (in vacuum): flux leaving $\mathrm{d}A_1$ in the direction $\boldsymbol{x}_1 \to \boldsymbol{x}_2$ must be equal to the flux arriving at $\mathrm{d}A_2$ from the direction $\boldsymbol{x}_1 \to \boldsymbol{x}_2$

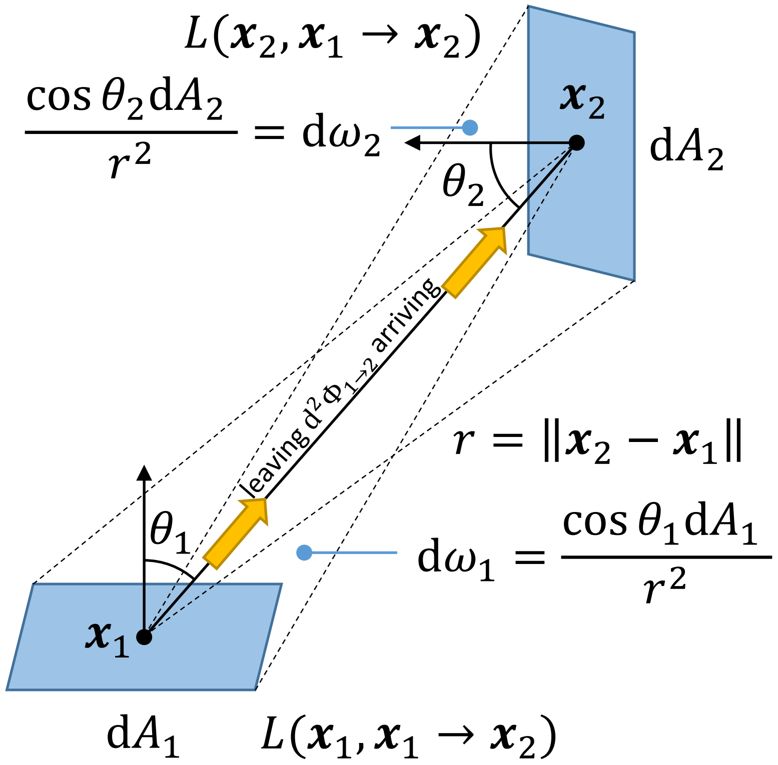
  → leaving flux $\mathrm{d}^2\Phi_{1\to2}$ = arriving flux $\mathrm{d}^2\Phi_{1\to2}$

- Leaving flux $\mathrm{d}^2\Phi_{1\to2} = L(\boldsymbol{x}_1, \boldsymbol{x}_1 \to \boldsymbol{x}_2)\cos\theta_1 \mathrm{d}A_1 \mathrm{d}\omega_2 =$

$$= L(\boldsymbol{x}_1, \boldsymbol{x}_1 \to \boldsymbol{x}_2)\frac{\cos\theta_1 \mathrm{d}A_1 \cos\theta_2 \mathrm{d}A_2}{r^2}$$

- Arriving flux $\mathrm{d}^2\Phi_{1\to2} = L(\boldsymbol{x}_2, \boldsymbol{x}_1 \to \boldsymbol{x}_2)\cos\theta_2 \mathrm{d}A_2 \mathrm{d}\omega_1 =$

$$= L(\boldsymbol{x}_2, \boldsymbol{x}_1 \to \boldsymbol{x}_2)\frac{\cos\theta_2 \mathrm{d}A_2 \cos\theta_1 \mathrm{d}A_1}{r^2}$$



Why is Radiance Constant Along Ray?

- The previous slide shows that if the two equations are equal and the fractions on the right sides of these equations are also equal, then the radiances $L(\boldsymbol{x}_1, \boldsymbol{x}_1 \to \boldsymbol{x}_2)$ and $L(\boldsymbol{x}_2, \boldsymbol{x}_1 \to \boldsymbol{x}_2)$ must also be equal.

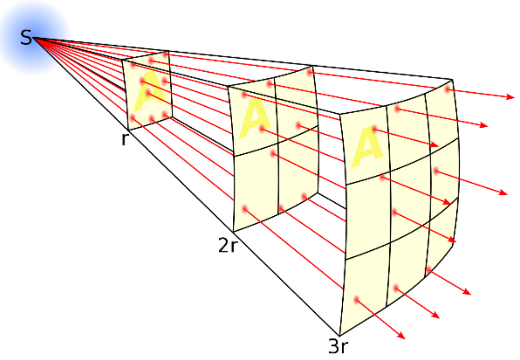
- If $L(\boldsymbol{x}_1, \boldsymbol{x}_1 \to \boldsymbol{x}_2) \frac{\cos\theta_1 \mathrm{d}A_1 \cos\theta_2 \mathrm{d}A_2}{r^2} = L(\boldsymbol{x}_2, \boldsymbol{x}_1 \to \boldsymbol{x}_2) \frac{\cos\theta_2 \mathrm{d}A_2 \cos\theta_1 \mathrm{d}A_1}{r^2}$ then also $L(\boldsymbol{x}_1, \boldsymbol{x}_1 \to \boldsymbol{x}_2) = L(\boldsymbol{x}_2, \boldsymbol{x}_1 \to \boldsymbol{x}_2)$

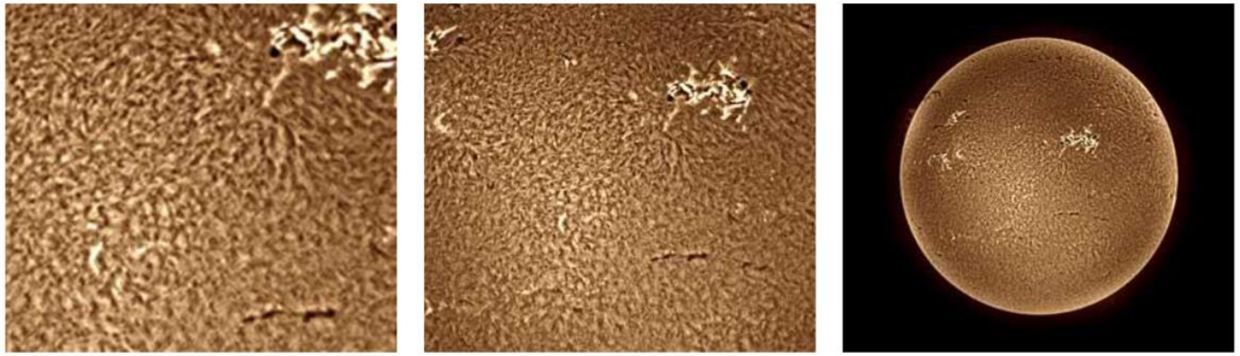
- Therefore, radiance L is invariant along straight paths and does not attenuate with distance.

Corollary of Constant Radiance

- Radiance in a pixel does not depend on the distance from the source
- Partial irradiance $\mathrm{d}E(\boldsymbol{x}, \mathrm{d}\omega) = \mathrm{d}\Phi / \mathrm{d}A$ of a pixel received from $\mathrm{d}A$ decreases as the square of distance from the source (i.e. is proportional to ${}^{1}/{}_{r^2}$)
- Area $\mathrm{d}A$ on the source captured by a pixel in directions $\mathrm{d}\omega$ increases as the square of distance (i.e. is proportional to r^2)

Inverse square law $E_1 r_1^2 = E_2 r_2^2$





J. J. Condon and S. M. Ransom:
"Essential Radio Astronomy", National Radio Astronomy Observatory

Inverse Square Law

Irradiance is often called intensity, but this term is avoided in radiometry where such usage leads to confusion with radiant intensity.

I = radiant intensity = zářivost
E = irradiance = intenzita (o)záření (ozářenost)

- Why $\frac{E_1}{E_2} = \frac{r_2^2}{r_1^2}$ and not $\frac{I_1}{I_2} = \frac{r_2^2}{r_1^2}$? ($E$ is irradiance/radiosity and I is radiant intensity)

$\Phi_1 = \Phi_2$ (conservation law)

$$E = \frac{\mathrm{d}\Phi}{\mathrm{d}A} \Rightarrow \mathrm{d}\Phi = E\,\mathrm{d}A$$

$$E_1 \mathrm{d}A_1 = E_2 \mathrm{d}A_2$$

$$E_1 4\pi r_1^2 R = E_2 4\pi r_2^2 R$$

$E_1 r_1^2 = E_2 r_2^2$
⇒ product of irradiance and the square of distance from the source is a constant

Auxiliary formulas:

$$\mathrm{d}\Omega = \frac{\mathrm{d}A}{r^2} = \frac{4\pi r^2 R}{r^2}$$

R represents a fraction of the area of the sphere we are dealing with, and without loss of generality can be equal to one.

$$\mathrm{d}\Omega_1 = \mathrm{d}\Omega_2$$

$$\frac{dA_1}{r_1^2} = \frac{dA_2}{r_2^2} \Rightarrow dA_2 = \frac{r_2^2}{r_1^2} dA_1$$

$\Phi_1 = \Phi_2$

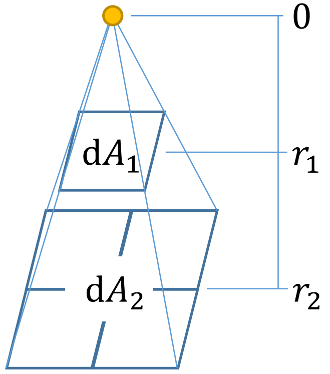
$$I = \frac{\mathrm{d}\Phi}{\mathrm{d}\Omega} \Rightarrow \mathrm{d}\Phi = I\,\mathrm{d}\Omega$$

$$I_1 \mathrm{d}\Omega_1 = I_2 \mathrm{d}\Omega_2$$

$$I_1 \frac{\mathrm{d}A_1}{r_1^2} = I_2 \frac{\mathrm{d}A_2}{r_2^2}$$

$$I_1 \frac{\mathrm{d}A_1}{r_1^2} = I_2 \frac{r_2^2}{r_1^2} \frac{\mathrm{d}A_1}{r_2^2}$$

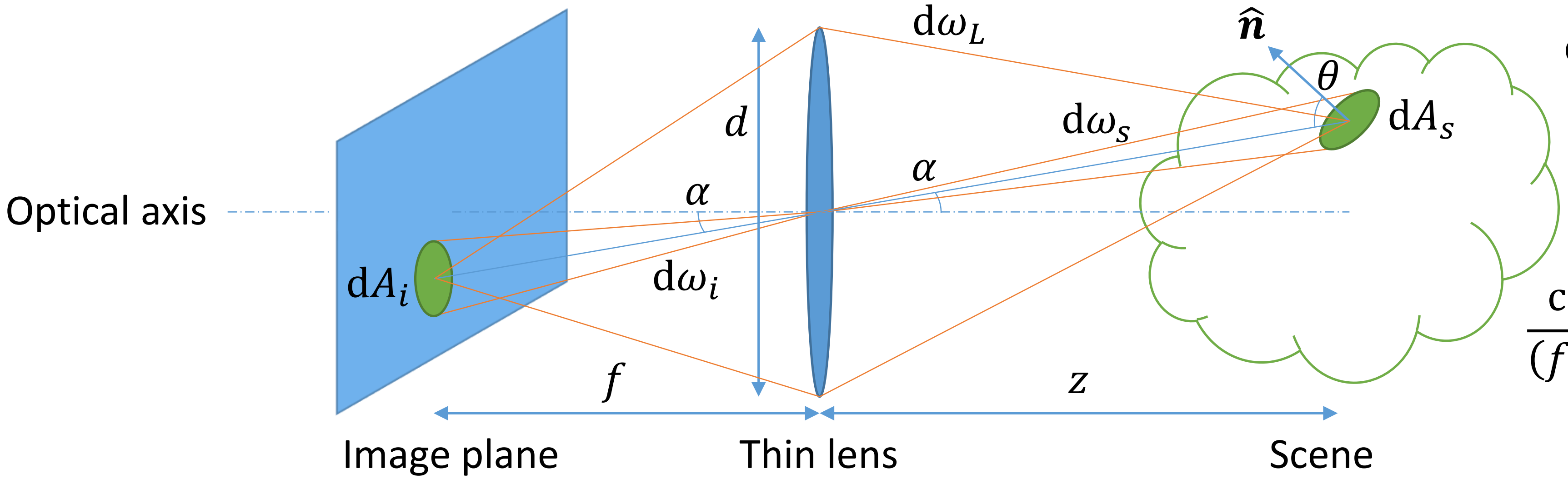
$I_1 = I_2$ ⇒ radiant intensity is the same regardless of r



$\mathrm{d}\Omega_1 = \mathrm{d}\Omega_2$
We assume that $\mathrm{d}A$ is a patch on the spherical surface of radius r directly facing the point source

Scene Brightness vs. Image Brightness

- What is the relation between the brightness of the scene point (radiance L) and the brightness on the image point (irradiance E)?



$$\mathrm{d}\omega_L = \frac{\pi d^2}{4}\frac{\cos(\alpha)}{(z/\cos(\alpha))^2}$$

$$\mathrm{d}\omega_i = \mathrm{d}\omega_s$$

$$\frac{\cos(\alpha)\mathrm{d}A_i}{(f/\cos(\alpha))^2} = \frac{\cos(\theta)\mathrm{d}A_s}{(z/\cos(\alpha))^2}$$

- From the energy conservation law:

flux $\mathrm{d}\Phi_s$ received by the lens of diam. d from $\mathrm{d}A_s$ $=$ flux $\mathrm{d}\Phi_i$ projected onto $\mathrm{d}A_i$

Scene Brightness vs. Image Brightness

From the previous slide, we also know that

$\mathrm{d}\omega_L = \frac{\pi d^2}{4}\frac{\cos(\alpha)}{(z/\cos(\alpha))^2}$ and $\frac{\mathrm{d}A_s}{\mathrm{d}A_i} = \frac{\cos(\alpha)}{\cos(\theta)}\left(\frac{z}{f}\right)^2$

- Scene radiance $L = \frac{\mathrm{d}^2\Phi}{\cos(\theta)\mathrm{d}A_s\mathrm{d}\omega_L}$
- Thus, flux received by the lens from $\mathrm{d}A_s$ is $\mathrm{d}^2\Phi = L\cos(\theta)\,\mathrm{d}A_s\mathrm{d}\omega_L$

- Image irradiance $E = \frac{\mathrm{d}\Phi}{\mathrm{dA_i}}$
- Thus, flux projected onto $\mathrm{d}A_i$ is $\mathrm{d}\Phi = E\ \mathrm{dA_i}$

We can conculde that image irradiance E is proportional to scene radiance L and image brightness falls off from image center as $(\cos(\alpha))^4$ and do not vary with scene depth

- $\mathrm{d}^2\Phi_s = \mathrm{d}\Phi_i \Rightarrow L\cos(\theta)\,\mathrm{d}A_s\mathrm{d}\omega_L = E\ \mathrm{dA_i} \Rightarrow$

$$E = L\cancel{\cos(\theta)}\frac{\pi d^2}{4}\frac{\cos(\alpha)}{(\cancel{z}/\cos(\alpha))^2}\frac{\cos(\alpha)}{\cancel{\cos(\theta)}}\left(\frac{\cancel{z}}{f}\right)^2 = \boxed{L\frac{\pi}{4}\left(\frac{d}{f}\right)^2(\cos(\alpha))^4}$$

Phong Reflection Model

- Designed by Bui Tuong Phong in 1975
- Baseline shading method for many rendering applications

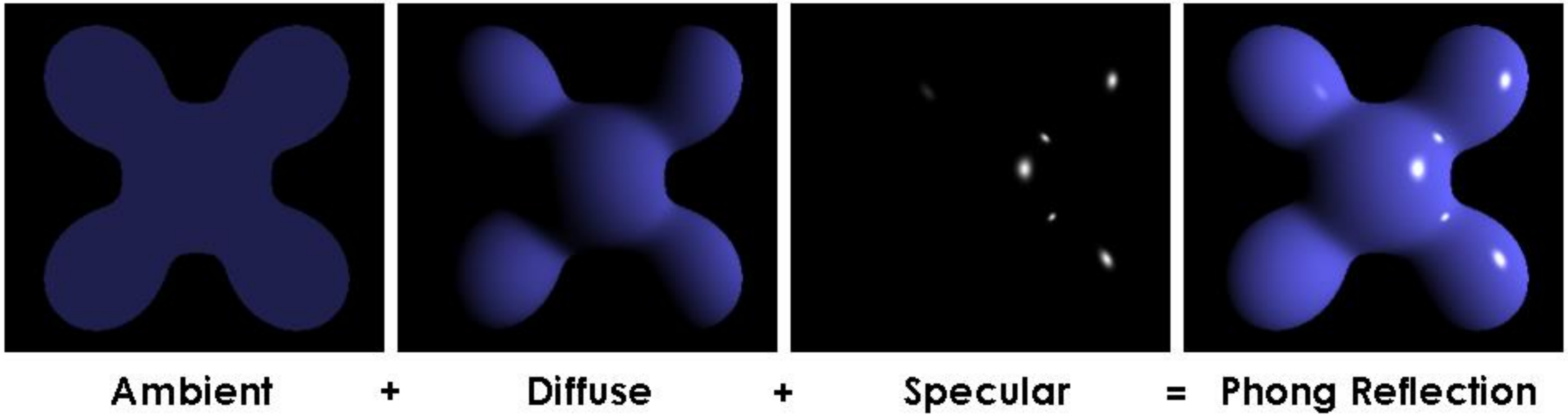
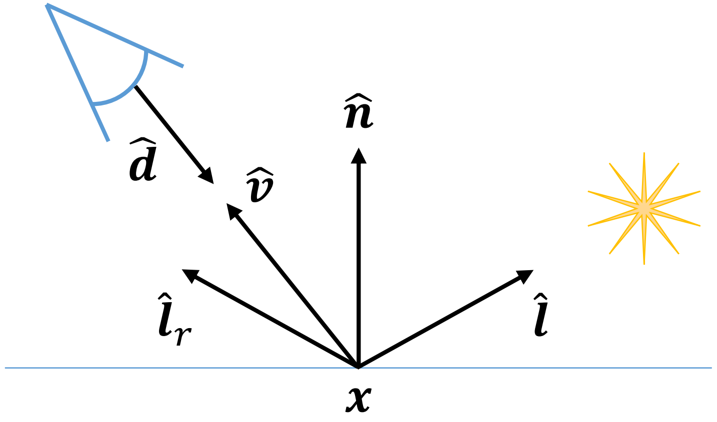


Illustration of the components of the Phong reflection model (Ambient, Diffuse and Specular reflection)
Source: Brad Smith

Phong Reflection Model



- Original definition

$$\boldsymbol{C} = \boldsymbol{I}_a \boldsymbol{m}_a + \sum\nolimits_{visible\ lights} \left(\boldsymbol{I}_d \boldsymbol{m}_d (\hat{\boldsymbol{n}} \cdot \hat{\boldsymbol{l}}) + \boldsymbol{I}_s \boldsymbol{m}_s (\hat{\boldsymbol{v}} \cdot \hat{\boldsymbol{l}}_r)^{\gamma} \right) \text{ where}$$

$$\hat{\boldsymbol{l}}_r = 2(\hat{\boldsymbol{n}} \cdot \hat{\boldsymbol{l}})\hat{\boldsymbol{n}} - \hat{\boldsymbol{l}}$$

For a white source set $\boldsymbol{I}_a = \boldsymbol{I}_d = \boldsymbol{I}_s = (1, 1, 1)$
Class Material contains γ as shininess

- Same equation (without ambient light) using radiometric quantities

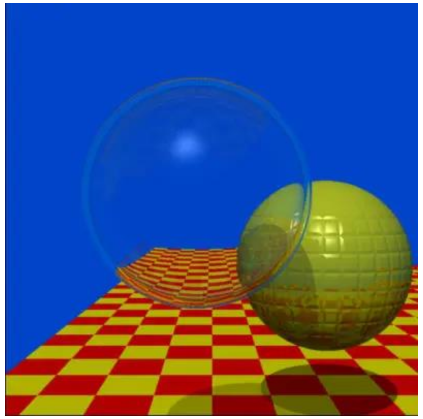
$$\boldsymbol{C} - \boldsymbol{I}_a \boldsymbol{m}_a \equiv \boldsymbol{L}_0(\omega_0) = \boldsymbol{L}_i(\omega_i)(\boldsymbol{m}_d \cos\theta_i + \boldsymbol{m}_s \cos^{\gamma}\theta_r)$$

$$\text{BRDF } f_r^{Phong} = \frac{\boldsymbol{L}_0(\omega_0)}{\boldsymbol{L}_i(\omega_i)\cos\theta_i} = \boldsymbol{m}_d + \frac{\boldsymbol{m}_s \cos^{\gamma}\theta_r}{\cos\theta_i}$$

Physically not correct. Why?

Red part is already in the rendering equation
and should not be present in BRDF

Whitted Ray Tracer



- Based on classical ray optics, solves reflection and refraction
- Opaque diffuse surfaces:
  - Cast shadow rays to determine visibility of light sources from the intersection point and use Phong illumination model
  - Shadow ray will affect both diffuse and specular term
- Mirror like surfaces:
  - Only specular reflection with no diffuse component
  - Trace another (secondary) reflection ray at the intersection point
- Transparent surfaces:
  - We will discuss this case in details later
- Missing effects: soft shadows, glossy reflections, diffuse reflections, DoF, …

Whitted Ray Tracer

- Whitted (direct) illumination model

$$\boldsymbol{C} = \boldsymbol{I}_a + \boldsymbol{I}_d \boldsymbol{m}_d \sum_{visible\ lights} (\widehat{\boldsymbol{n}} \cdot \hat{\boldsymbol{l}}) + \boldsymbol{k}_s \boldsymbol{S} + \boldsymbol{k}_t \boldsymbol{T} \text{ where}$$

$\boldsymbol{S}$ and $\boldsymbol{T}$ are intensities of light coming from reflection, resp. refraction,

$\boldsymbol{k}_s$ and $\boldsymbol{k}_t$ are specular, resp. transmission, coefficients

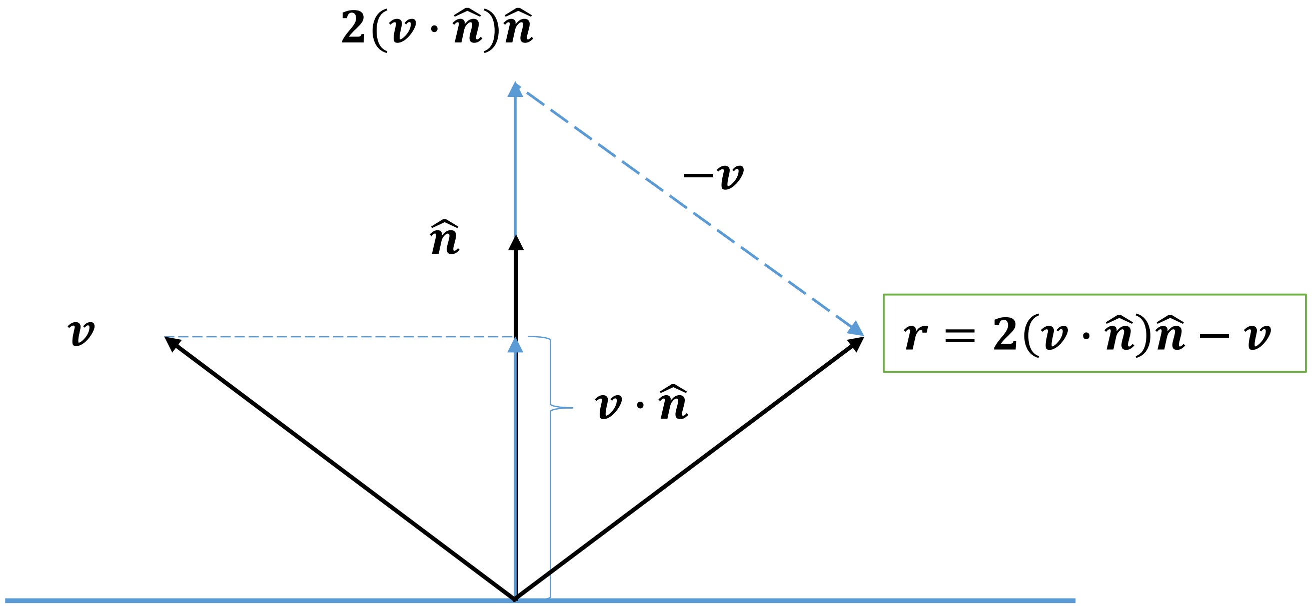
- We may add some attenuation factor to take the distance of light source into account

Whitted Ray Tracer

```
for each pixel:
  primary_ray = GeneratePay( pixel )
  pixel = Trace( primary_ray, 0 )

color Trace( ray, depth, max_depth = 5 ):
  if ( depth < max_depth ):
    normal, material, hit_point = FindNearestIntersection( scene, ray )
    if ( material is diffuse ):
      return material.diffuse * direct_illumination
    if ( material is mirror ):
      return material.specular * Trace( reflected_ray, depth + 1 ) ~~* normal.dot( light_dir )~~
  return background_color
```

Light transport modulation factors

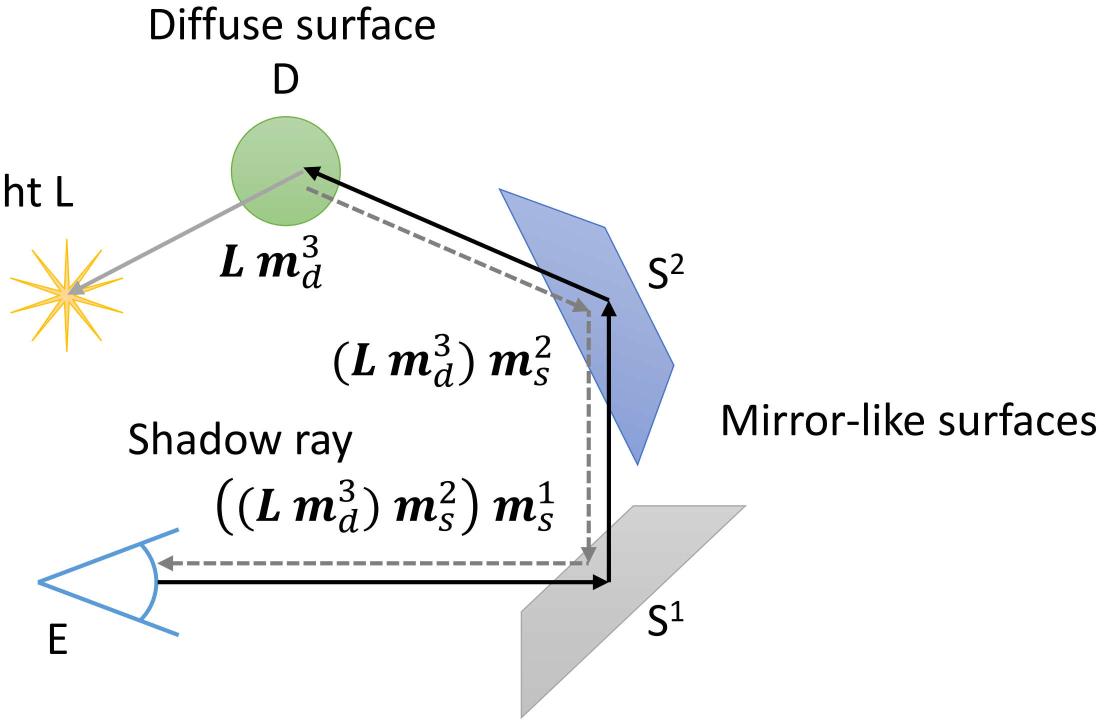
Reflection



$$\boldsymbol{r} = 2(\boldsymbol{v} \cdot \hat{\boldsymbol{n}})\hat{\boldsymbol{n}} - \boldsymbol{v}$$

Whitted Ray Tracer

- Light paths (E S* D L)



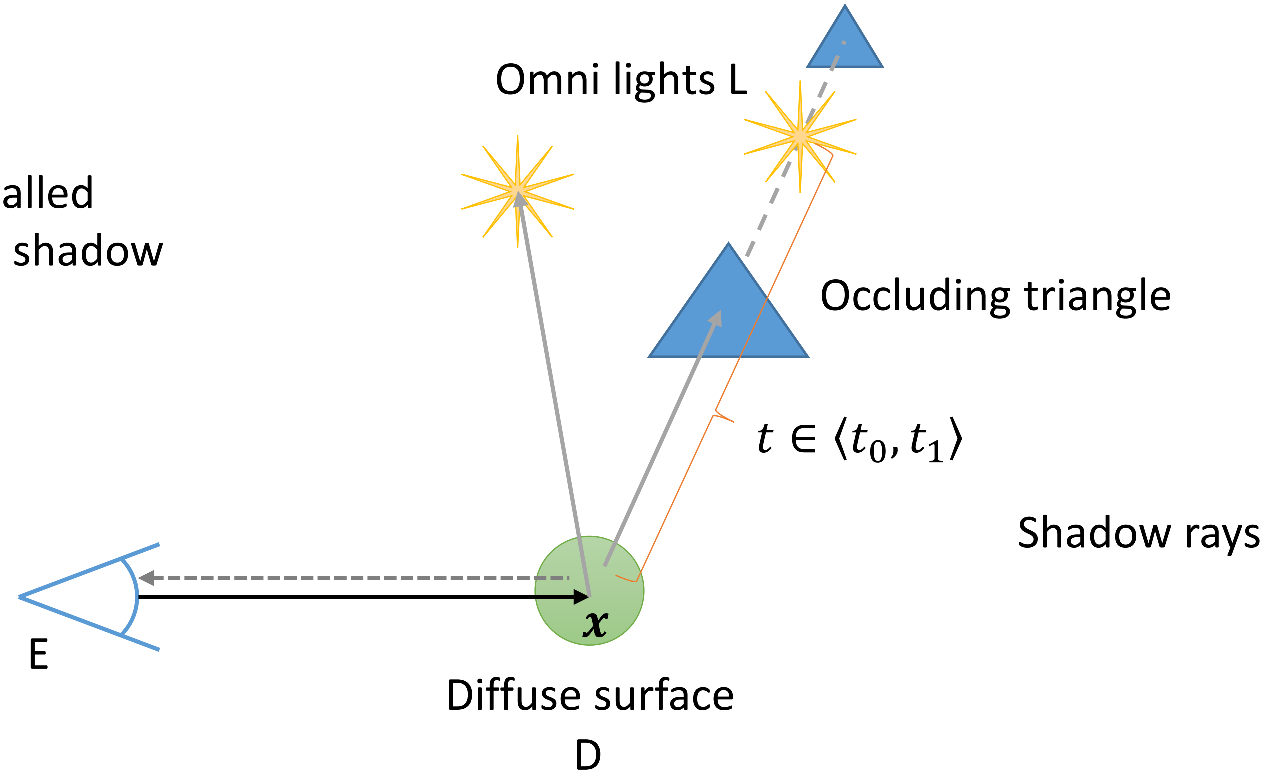
* $0 \ldots n$ hits

(Hard) Shadows

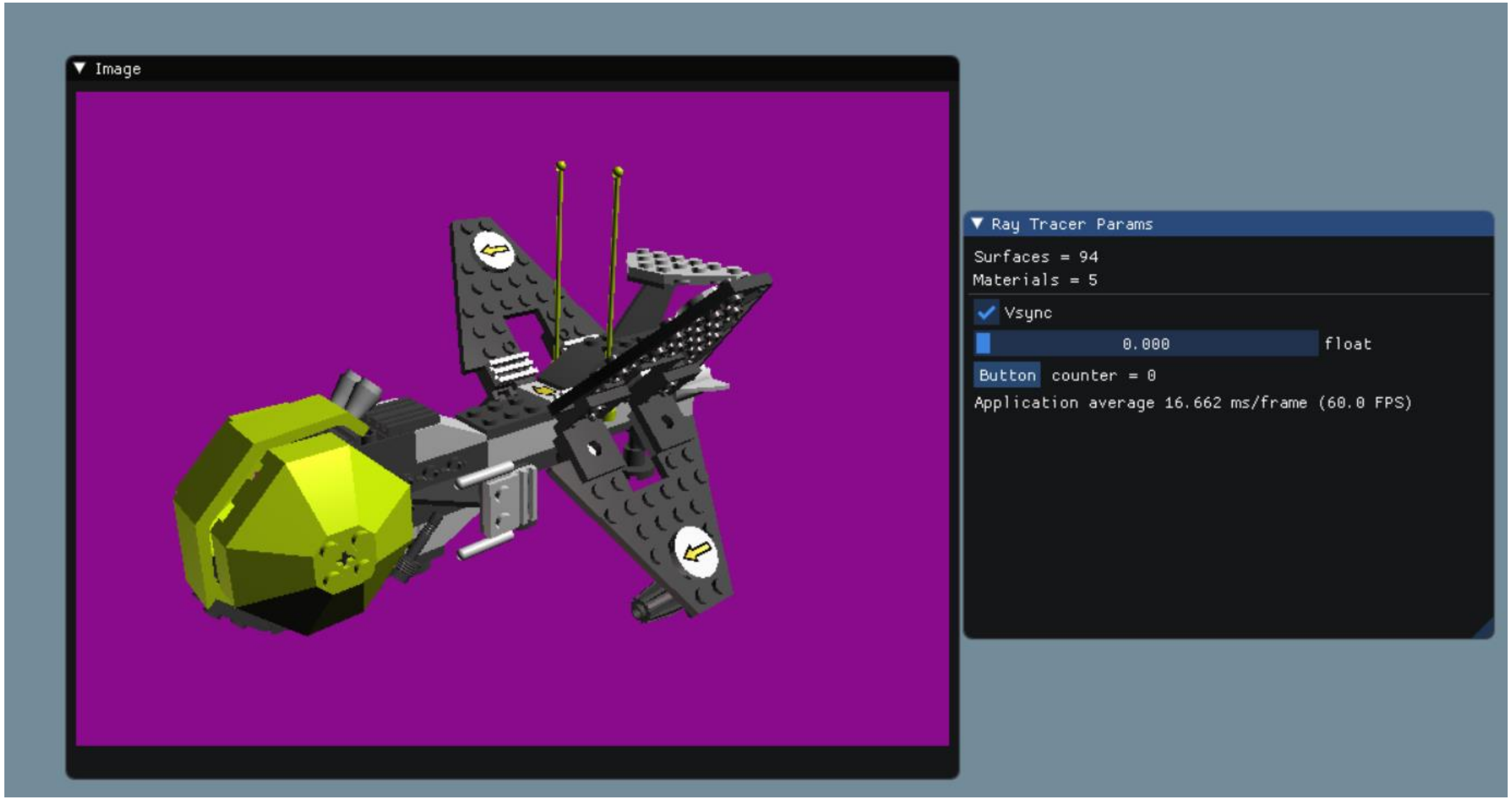
To avoid errorneous self-shadowing called shadow acne, move the origin of the shadow ray a bit further above the surface

View direction offset
Normal offset

```
ray.tnear = 0.001
ray.tfar = ||l||
```



Result of Second Exercise

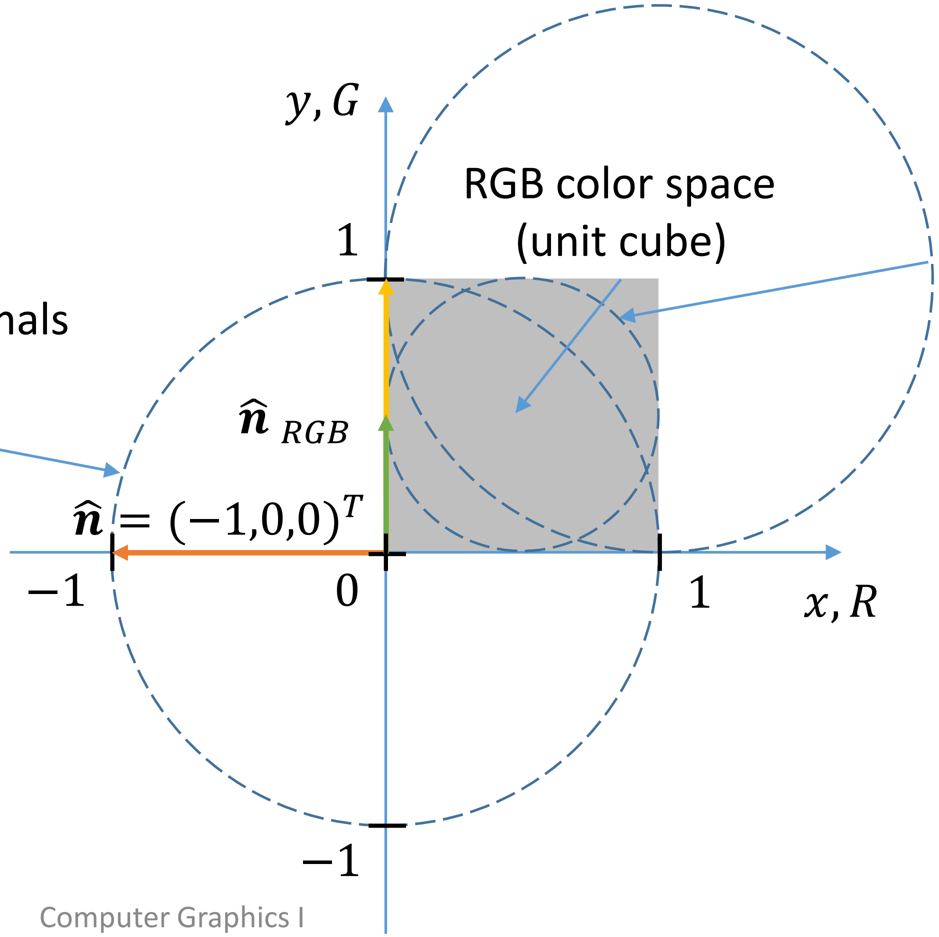


```
Surfaces = 94
Materials = 5
☑ Vsync
0.000 float
counter = 0
Application average 16.662 ms/frame (60.0 FPS)
```

Normal Shader

- We can easily see that normal cannot be represented in RGB space directly

$$\hat{\boldsymbol{n}}_{RGB} = \frac{\hat{\boldsymbol{n}} + (1,1,1)^T}{2}$$



Space of possible normals
(unit sphere)

RGB color space
(unit cube)

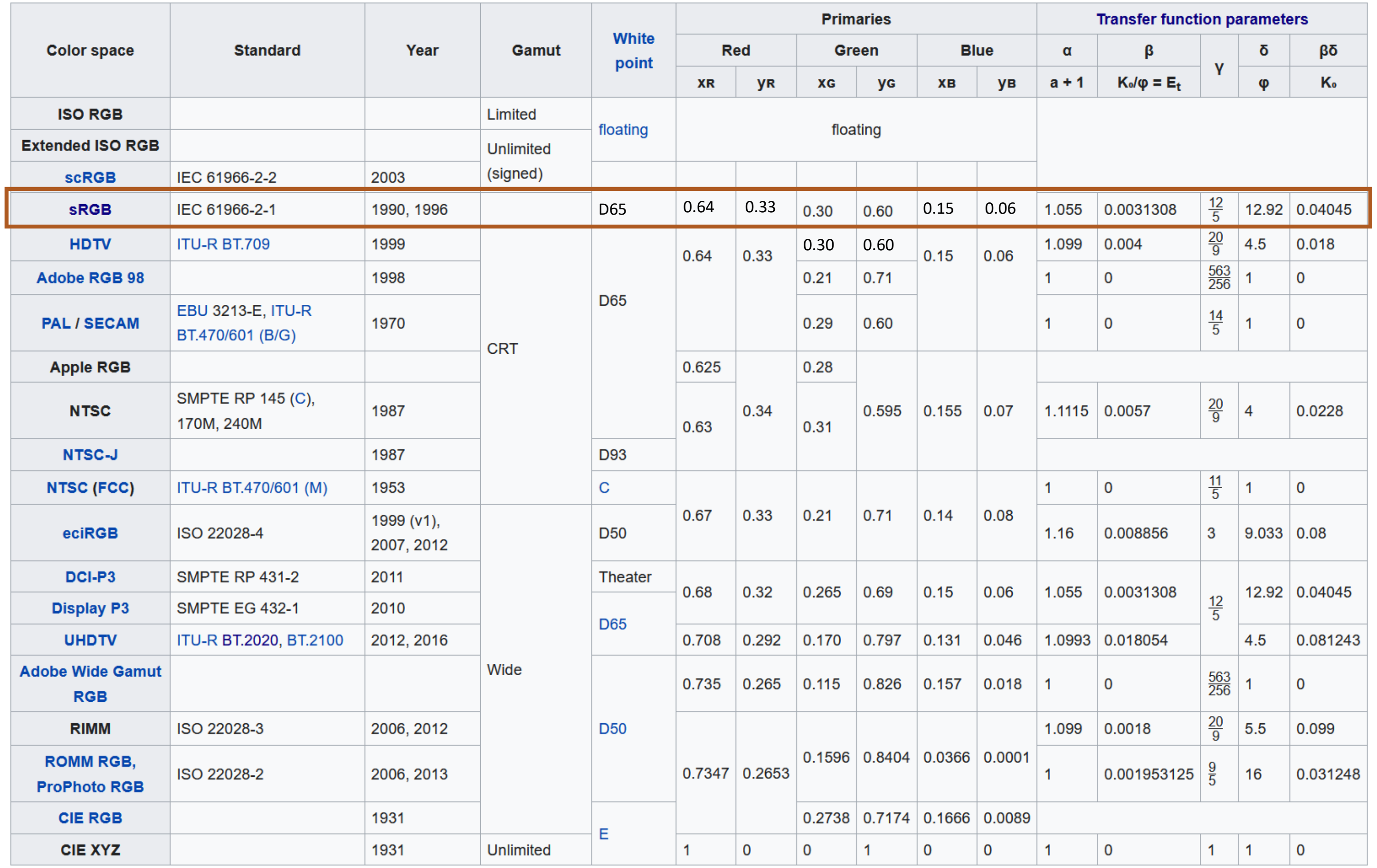
Space of possible
„colorized“ normals
(sphere with unit
diameter and centered
at (0.5, 0.5, 0.5))

Note that the z-axis, resp. the blue color axis, are omitted for the sake of brevity

RGB vs sRGB Color Spaces

- RGB color space is any additive color space based on RGB color model that employs RGB primaries (i.e. red, green, and blue chromacities)
- Primary colors are defined by their CIE 1931 color space chromacity coordinates (x, y)
- Specification of any RGB color space also includes definition of white point and transfer function
- sRGB is one of many (but by far the most commonly used) color spaces for computer displays
- Our renderer will produce sRGB images

Specifications of RGB color spaces



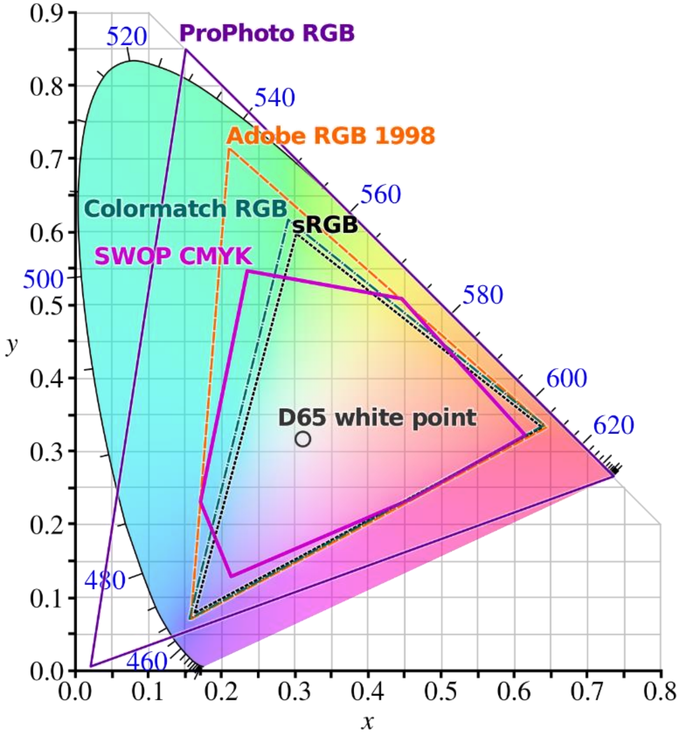
Color space	Standard	Year	Gamut	White point	Primaries						Transfer function parameters				
					Red		Green		Blue		α	β	γ	δ	βδ
					xR	yR	xG	yG	xB	yB	a + 1	$K_0/\varphi = E_t$		φ	K_0
ISO RGB			Limited	floating	floating										
Extended ISO RGB			Unlimited (signed)												
scRGB	IEC 61966-2-2	2003													
sRGB	IEC 61966-2-1	1990, 1996		D65	0.64	0.33	0.30	0.60	0.15	0.06	1.055	0.0031308	$\frac{12}{5}$	12.92	0.04045
HDTV	ITU-R BT.709	1999	CRT	D65	0.64	0.33	0.30	0.60	0.15	0.06	1.099	0.004	$\frac{20}{9}$	4.5	0.018
Adobe RGB 98		1998					0.21	0.71			1	0	$\frac{563}{256}$	1	0
PAL / SECAM	EBU 3213-E, ITU-R BT.470/601 (B/G)	1970					0.29	0.60			1	0	$\frac{14}{5}$	1	0
Apple RGB					0.625	0.34	0.28	0.595	0.155	0.07					
NTSC	SMPTE RP 145 (C), 170M, 240M	1987			0.63		0.31				1.1115	0.0057	$\frac{20}{9}$	4	0.0228
NTSC-J		1987		D93											
NTSC (FCC)	ITU-R BT.470/601 (M)	1953		C	0.67	0.33	0.21	0.71	0.14	0.08	1	0	$\frac{11}{5}$	1	0
eciRGB	ISO 22028-4	1999 (v1), 2007, 2012	Wide	D50							1.16	0.008856	3	9.033	0.08
DCI-P3	SMPTE RP 431-2	2011		Theater	0.68	0.32	0.265	0.69	0.15	0.06	1.055	0.0031308	$\frac{12}{5}$	12.92	0.04045
Display P3	SMPTE EG 432-1	2010		D65											
UHDTV	ITU-R BT.2020, BT.2100	2012, 2016			0.708	0.292	0.170	0.797	0.131	0.046	1.0993	0.018054		4.5	0.081243
Adobe Wide Gamut RGB				D50	0.735	0.265	0.115	0.826	0.157	0.018	1	0	$\frac{563}{256}$	1	0
RIMM	ISO 22028-3	2006, 2012			0.7347	0.2653	0.1596	0.8404	0.0366	0.0001	1.099	0.0018	$\frac{20}{9}$	5.5	0.099
ROMM RGB, ProPhoto RGB	ISO 22028-2	2006, 2013									1	0.001953125	$\frac{9}{5}$	16	0.031248
CIE RGB		1931		E			0.2738	0.7174	0.1666	0.0089					
CIE XYZ		1931	Unlimited		1	0	0	1	0	0	1	0	1	1	0

Source: https://en.wikipedia.org/wiki/RGB_color_space

Color Gamut

- A color gamut is defined as a range of colors that a particular device is capable of displaying or recording
- It usually appears as a closed area of primary colors in a chromaticity diagram. The missing dimension is the brightness, which is perpendicular to the screen or paper
- Color gamut is displayed as a triangular area enclosed by color coordinates corresponding to the red, green, and blue color

CIE 1931 XYZ color space



Color Gamut

- sRGB - by far the most commonly used color space for computer displays
- NTSC – standard for analog television
- Adobe RGB (1998) - encompass most of the colors achievable on CMYK color printers, but by using RGB primary colors on a device such as a computer display
- DCI-P3 – space for digital movie projection from the American film industry
- EBU – European color space surpassing the PAL standard
- Rec. 709 - shares the sRGB primaries, used in HDTVs
- Rec. 2024 – 4K or 8K resolution at 10 or 12 bits per channel
- Remember that 72% NTSC is not sRGB (which is often claimed). Matching the ratios of the color gamut areas does not necessarily guarantee the ability to achieve the same image (the degree of overlap of the triangles is important and not the ratio of their areas).

Linear sRGB and Gamma Compressed sRGB

- Images displayed on monitors are encoded in nonlinear sRGB color space to compensate the transformation of brightness the monitor does
- Our render has to work in linear space as we are using linear operations with colors
- Every texel and material color have to be processed in linear sRGB color model
- Only the final color values stored in framebuffer are converted back to gamma compensated sRGB values
- Issue with blending two sRGB colors in non-linear color space:

$\boldsymbol{C}_{srgb} = \alpha \boldsymbol{A}_{srgb} + (1-\alpha)\boldsymbol{B}_{srgb}$ doesn work well

$\boldsymbol{C}_{rgb} = \alpha \boldsymbol{A}_{rgb} + (1-\alpha)\boldsymbol{B}_{rgb}$ correct

$\boldsymbol{C}_{srgb} = ToSRGB\left(\alpha\, ToRGB\left(\boldsymbol{A}_{srgb}\right) + (1-\alpha)\, ToRGB\left(\boldsymbol{B}_{srgb}\right)\right)$ correct

Note the undesirable dark silhouettes when mixing two colors directly in sRGB space (created in paint.net)

Gamma Correction

- The human visual system response is logarithmic, not linear, resulting in the ability to perceive an incredible brightness range of over 10 decades
- Gamma characterizes the reproduction of tone scale in an imaging system. Gamma summarizes, in a single numerical parameter, the nonlinear relationship between code value (in an 8-bit system, from 0 through 255) and luminance. Nearly all image coding systems are nonlinear, and so involve values of gamma different from unity
- The main purpose of gamma correction is to code luminance into a perceptually-uniform domain, so as optimize perceptual performance of a limited number of bits in each channel

Source: POYNTON, Charles A. The rehabilitation of gamma, 2000.

sRGB Transfer Functions

- Function returns gamma-expanded (or linear) sRGB values from gamma-compressed (or non-linear) sRGB values

```
float c_linear( float c_srgb, float gamma = 2.4f )
{
  if ( c_srgb <= 0.0f ) return 0.0f;
  else if ( c_srgb >= 1.0f ) return 1.0f;

  assert( ( c_srgb >= 0.0f ) && ( c_srgb <= 1.0f ) );

  if ( c_srgb <= 0.04045f )
  {
    return c_srgb / 12.92f;
  }
  else
  {
    const float a = 0.055f;
    return powf( ( c_srgb + a ) / ( 1.0f + a ), gamma );
  }
}
```

- Function returns gamma-compressed (or non-linear) sRGB values from gamma-expanded (or linear) sRGB values

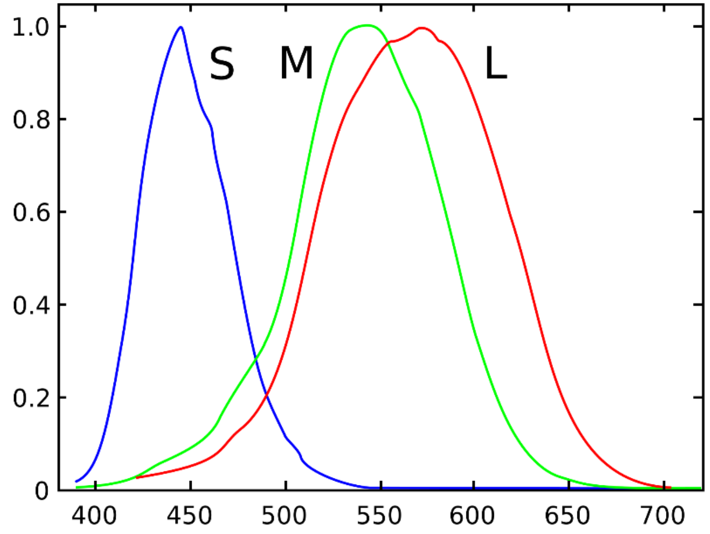
```
float c_srgb( float c_linear, float gamma = 2.4f )
{
  if ( c_linear <= 0.0f ) return 0.0f;
  else if ( c_linear >= 1.0f ) return 1.0f;

  assert( ( c_linear >= 0.0f ) && ( c_linear <= 1.0f ) );

  if ( c_linear <= 0.0031308f )
  {
    return 12.92f * c_linear;
  }
  else
  {
    const float a = 0.055f;
    return ( 1.0f + a )*powf( c_linear, 1.0f / gamma ) - a;
  }
}
```

Spectral Sensitivity of Human Eye

- The human eye with normal vision has three kinds of cone cells (čípek) that sense light (trichromacy vs pentachromacy), having peaks of spectral sensitivity in short ("S", 420 nm – 440 nm), middle ("M", 530 nm – 540 nm), and long ("L", 560 nm – 580 nm) wavelengths
- LMS forms a 3D cone space

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.38971 & 0.68898 & -0.07868 \\ -0.22981 & 1.18340 & 0.04641 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 1.91019 & -1.11214 & 0.20245 \\ 0.37095 & 0.62905 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix}$$

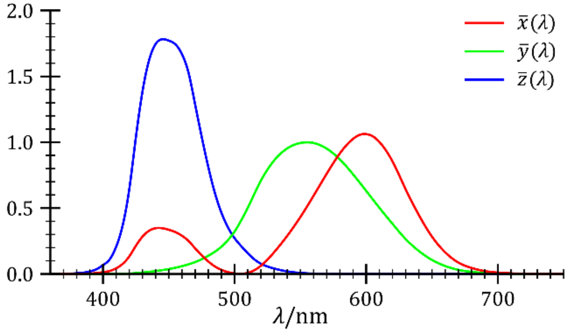


Normalized responsivity spectra of human cone

CIE 1931 Color Space

Can be thought of as the spectral sensitivity curves of three linear light detectors yielding the CIE tristimulus values *X*, *Y* and *Z*. Describe the CIE standard observer.

CIE's Color Matching Functions



- CIE XYZ is a color space deliberately designed so that the Y parameter represents the luminance of a color
- We can compute XYZ values from spectral data $L_{e,\Omega}$ as follows

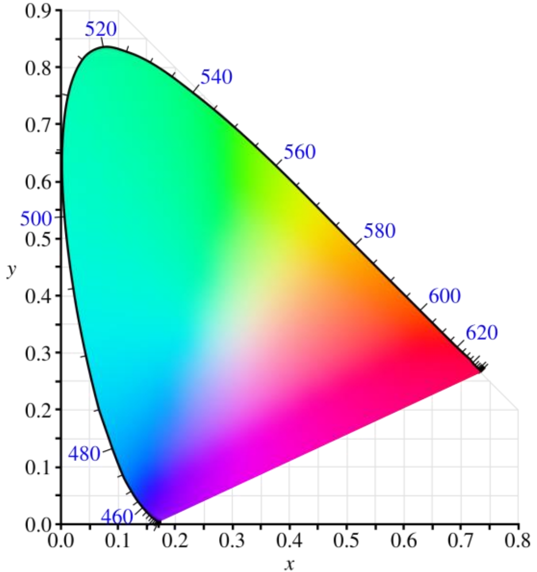
$$\{X,Y,Z\} = \int_{\lambda} L_{e,\Omega}(\lambda)\{\bar{x},\bar{y},\bar{z}\}(\lambda)\,\mathrm{d}\lambda \text{ where } \lambda \in \langle 380, 780\rangle \text{ nm}$$

spectral radiance (W·sr^{-1}·m^{-2}·nm^{-1}) of a reference illuminant

$$X = \frac{Y}{y}x;\ Z = \frac{Y}{y}(1 - x - y)$$

- CIE xyY is a three dimensional color space defined by (x, y) (chromaticity) and Y (luminance) parameters

$$x = \frac{X}{X+Y+Z};\ y = \frac{Y}{X+Y+Z};\ z = \frac{Z}{X+Y+Z} = 1 - x - y$$



CIE chromaticity diagram

sRGB Color Space

- Forward

$$\begin{bmatrix} R_{linear} \\ G_{linear} \\ B_{linear} \end{bmatrix} = \begin{bmatrix} 3.240969942 & -1.537383178 & -0.49861076 \\ -0.969243636 & 1.875967502 & 0.041555057 \\ 0.05563008 & -0.203976959 & 1.056971514 \end{bmatrix} \begin{bmatrix} X_{D65} \\ Y_{D65} \\ Z_{D65} \end{bmatrix}$$

- and backward linear transformation follows

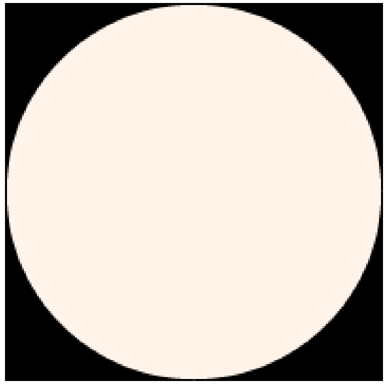
$$\begin{bmatrix} X_{D65} \\ Y_{D65} \\ Z_{D65} \end{bmatrix} = \begin{bmatrix} 0.412390799 & 0.357584339 & 0.180480788 \\ 0.212639006 & 0.715168679 & 0.072192316 \\ 0.019330819 & 0.119194780 & 0.950532152 \end{bmatrix} \begin{bmatrix} R_{linear} \\ G_{linear} \\ B_{linear} \end{bmatrix}$$

- And finally, gamma-compression (or expansion) takes place here

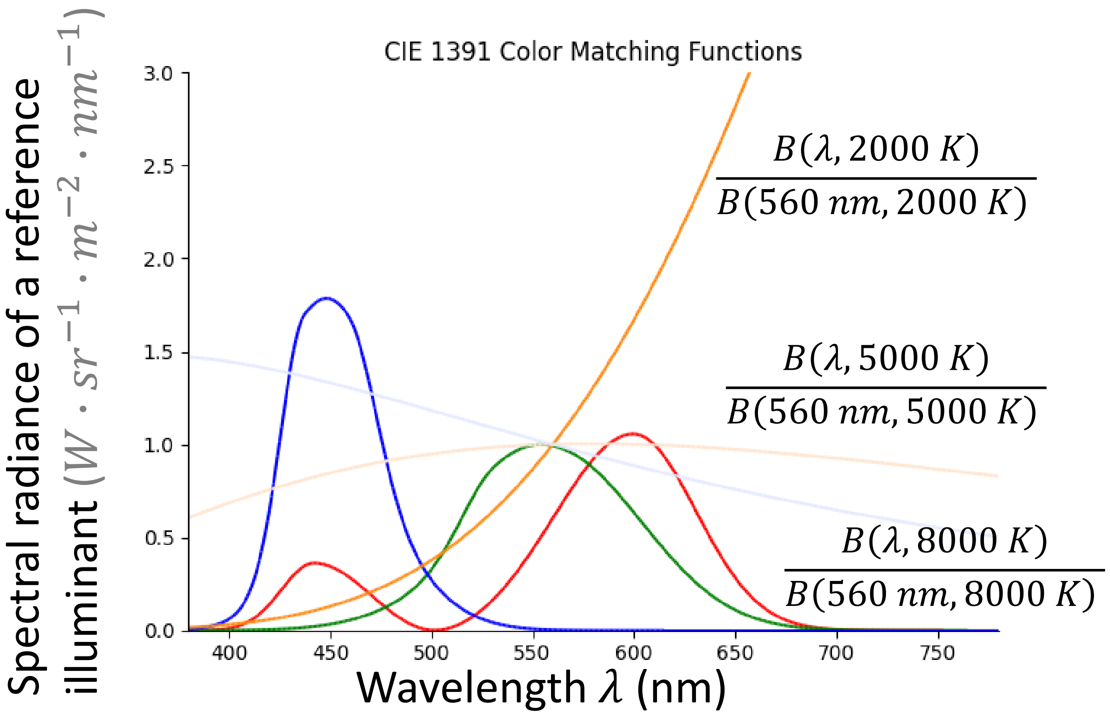
Experiment 1

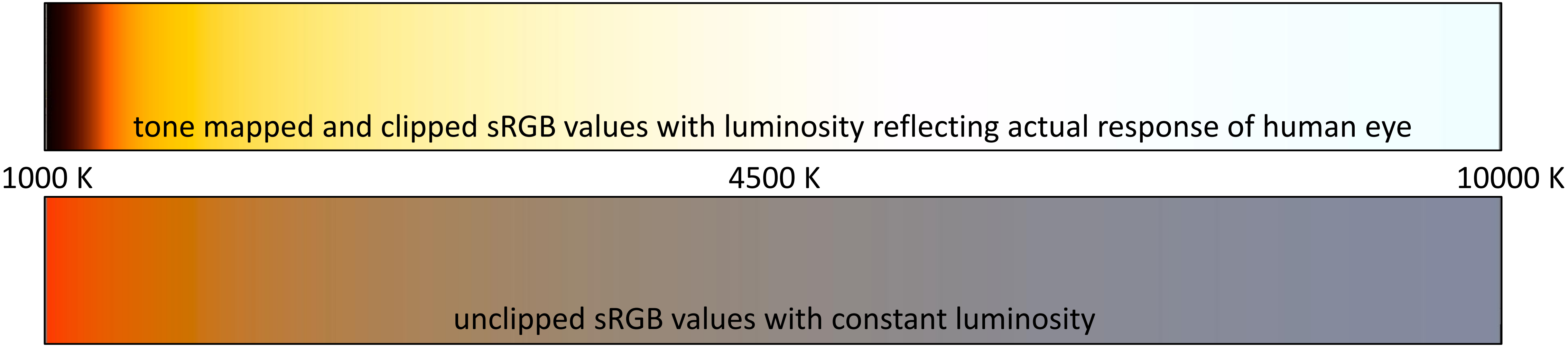
- Blackbody radiation

$$B(\lambda, T) = \frac{2\hbar c^2}{\lambda^5} \frac{1}{e^{\frac{\hbar c}{\lambda k_B T}} - 1}$$

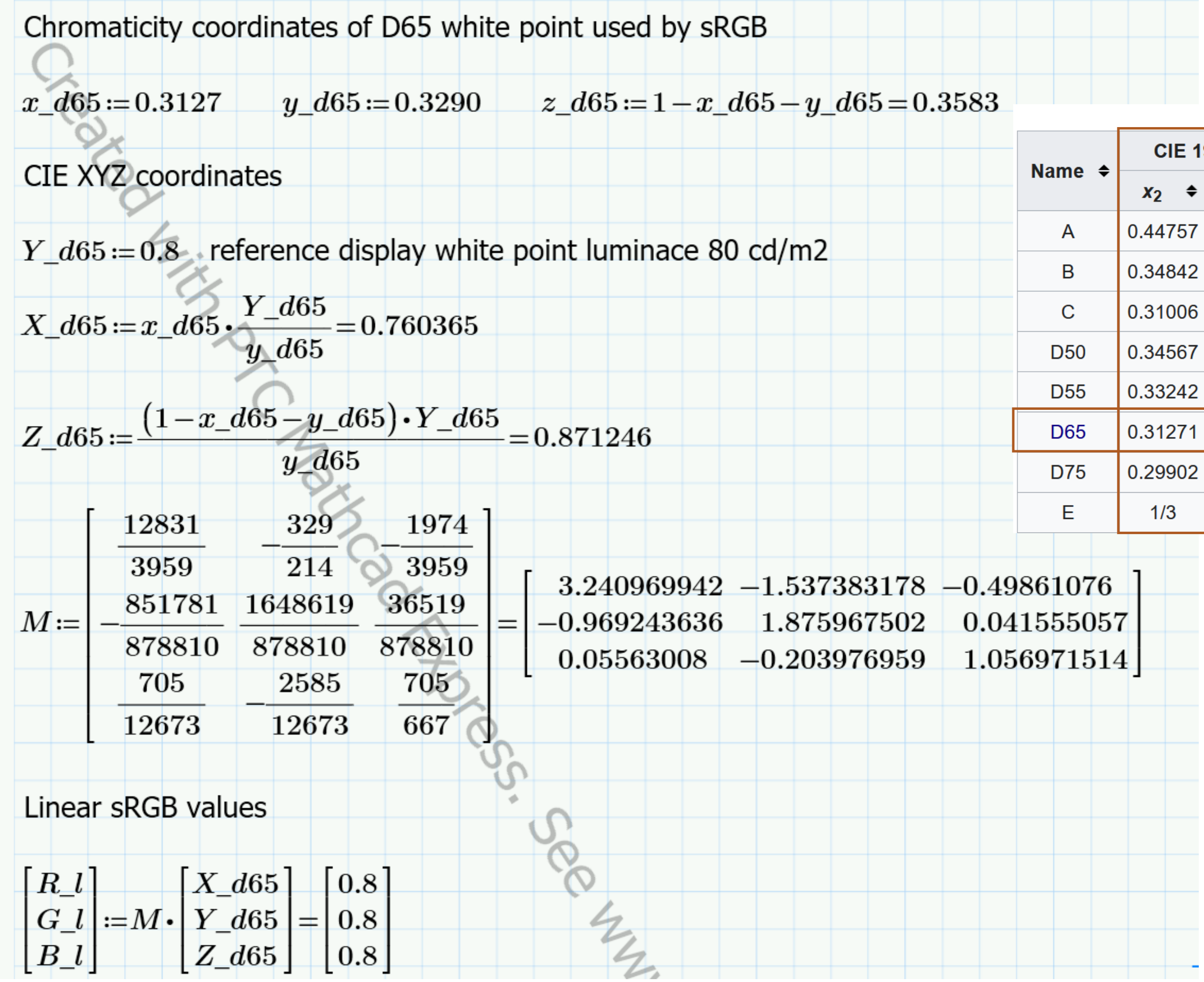


True color of the Sun





Experiment 2



Chromaticity coordinates of D65 white point used by sRGB

$x\_d65 := 0.3127 \quad y\_d65 := 0.3290 \quad z\_d65 := 1 - x\_d65 - y\_d65 = 0.3583$

CIE XYZ coordinates

$Y\_d65 := 0.8$ reference display white point luminace 80 cd/m2

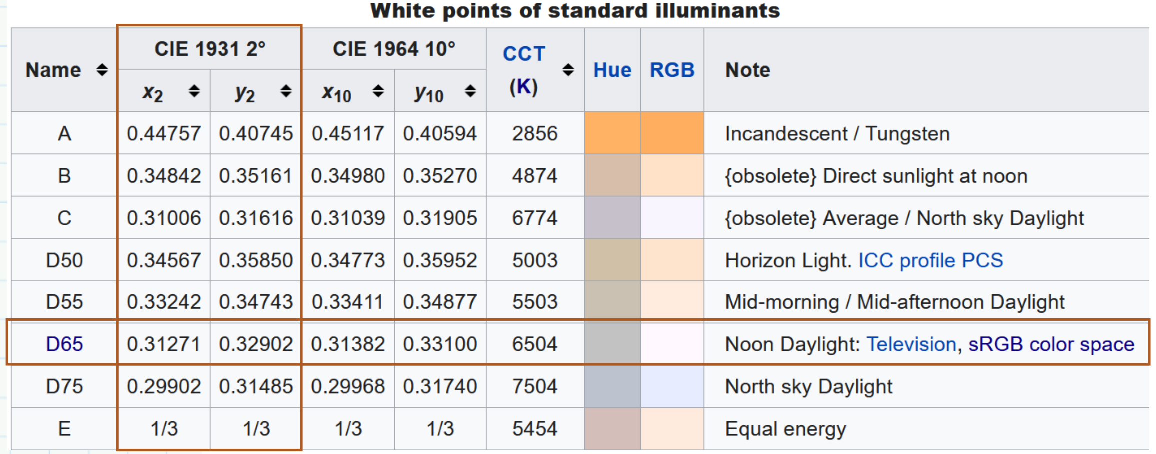
$$X\_d65 := x\_d65 \cdot \frac{Y\_d65}{y\_d65} = 0.760365$$

$$Z\_d65 := \frac{(1 - x\_d65 - y\_d65) \cdot Y\_d65}{y\_d65} = 0.871246$$

$$M := \begin{bmatrix} \frac{12831}{3959} & -\frac{329}{214} & -\frac{1974}{3959} \\ -\frac{851781}{878810} & \frac{1648619}{878810} & \frac{36519}{878810} \\ \frac{705}{12673} & -\frac{2585}{12673} & \frac{705}{667} \end{bmatrix} = \begin{bmatrix} 3.240969942 & -1.537383178 & -0.49861076 \\ -0.969243636 & 1.875967502 & 0.041555057 \\ 0.05563008 & -0.203976959 & 1.056971514 \end{bmatrix}$$

Linear sRGB values

$$\begin{bmatrix} R\_l \\ G\_l \\ B\_l \end{bmatrix} := M \cdot \begin{bmatrix} X\_d65 \\ Y\_d65 \\ Z\_d65 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 0.8 \\ 0.8 \end{bmatrix}$$



White points of standard illuminants

Name	CIE 1931 2°		CIE 1964 10°		CCT (K)	Hue	RGB	Note
	x_2	y_2	x_{10}	y_{10}				
A	0.44757	0.40745	0.45117	0.40594	2856	■	■	Incandescent / Tungsten
B	0.34842	0.35161	0.34980	0.35270	4874	■	■	{obsolete} Direct sunlight at noon
C	0.31006	0.31616	0.31039	0.31905	6774	■	■	{obsolete} Average / North sky Daylight
D50	0.34567	0.35850	0.34773	0.35952	5003	■	■	Horizon Light. ICC profile PCS
D55	0.33242	0.34743	0.33411	0.34877	5503	■	■	Mid-morning / Mid-afternoon Daylight
D65	0.31271	0.32902	0.31382	0.33100	6504	■	■	Noon Daylight: Television, sRGB color space
D75	0.29902	0.31485	0.29968	0.31740	7504	■	■	North sky Daylight
E	1/3	1/3	1/3	1/3	5454	■	■	Equal energy

Source: https://en.wikipedia.org/wiki/Standard_illuminant

Experiment 3

- Direct mixing of two gamma-compressed sRGB colors

- Mixing of two gamma-expanded sRGB colors

Code for Experiment 3

```
def _compress(u):
  if u <= 0:
    return 0
  if u >= 1:
    return 1
  if u <= 0.0031308:
    return 12.92 * u
  else:
    return (1.055 * u**(1 / 2.4) - 0.055)

def _expand(u):
  if u <= 0:
    return 0
  if u >= 1:
    return 1
  if u <= 0.04045:
    return u / 12.92
  else:
    return ( ( u + 0.055 ) / 1.055 )**2.4
```

```
def compress(c_in):
  c_out = []
  for i in range(3):
    c_out.append(_compress(c_in[i]))
  return tuple(c_out)

def expand(c_in):
  c_out = []
  for i in range(3):
    c_out.append(_expand(c_in[i]))
  return tuple(c_out)

def mix_linear(c0, c1, alpha):
  c_out = []
  for i in range(3):
    c_out.append(alpha * c0[i] + (1 - alpha) * c1[i])
  return tuple(c_out)

def mix_srgb(c0, c1, alpha):
  return compress(mix_linear(expand(c0), expand(c1), alpha))
```

Tone-mapping

- A process which maps an input image of high dynamic range (HDR) to a limited low dynamic range (LDR)
- Typical output devices such as LCD monitors have LDR (i.e. accept values with a very narrow range of $\langle 0, 1 \rangle$)
- A ray tracer produces linear HDR outputs with a potentially unlimited range of $\langle 0, \infty)$
- We need to scale down these HDR values into LDR somehow
- We can use various (tone)-mapping operators (or functions) to do so