POLITECNICO
MILANO 1863

DIPARTIMENTO DI ELETTRONICA
INFORMAZIONE E BIOINGEGNERIA

DEIB

2024

Dipartimento di Elettronica, Informazione e Bioingegneria

Computer Graphics

Milano, 2024

Computer Graphics

- Light models

Scan-line rendering

As previously outlined, in both scan-line rendering and ray-casting, the scene is composed by a finite set of light sources. The contributions of all lights l are added together to compute the final color of the pixel.

Initially, we will ignore the possibility of objects to emit small amount of lights, further simplifying the equation.

$$L(x,\omega_r) = \cancel{L_e(x,\omega_r)} + \sum_l L_e\left(l,\overrightarrow{lx}\right) f_r\left(x,\overrightarrow{lx},\omega_r\right)$$

Scan-line rendering

Each term in the summation is the product of the *light model*, that computes the quantity and direction of the considered light source, and the *BRDF* which accounts how the surface reflects the light.

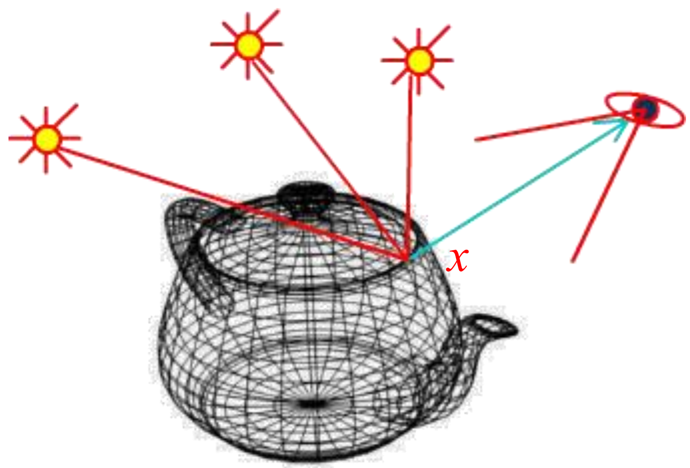
$$L(x, \omega_r) = \sum_l L(l, \overrightarrow{lx}) f_r(x, \overrightarrow{lx}, \omega_r)$$

Light model

BRDF

Light models

A light model describes how light is emitted in the different directions of the space. It takes as input the position of a point *x* of an object. It returns two elements: a vector that represents the *direction* of the light, and a color which accounts for the *intensity* of light received by point *x* for every wavelength.

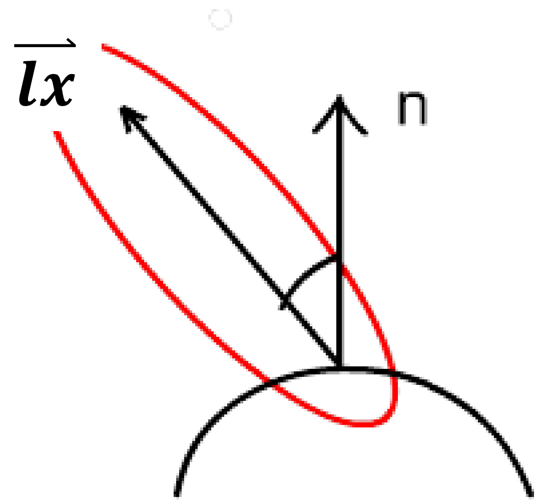


$$L(x, \omega_r) = \sum_l \overset{\text{intensity}}{\boxed{L\left(l, \overrightarrow{lx}\right)}} f_r\left(x, \overset{\text{direction}}{\boxed{\overrightarrow{lx},}} \omega_r\right)$$

Light direction

The light direction can then be specified with a vector $\overrightarrow{\boldsymbol{lx}} = (d_x, d_y, d_z)$: as a convention, the sign of the light direction is chosen to make the ray point *toward the light source*.

Moreover, the direction of the light is a unitary vector: $|\overrightarrow{\boldsymbol{lx}}| = 1$.



Light color

A vector $L(l, \overrightarrow{lx}) = (l_R, l_G, l_B)$ of RGB components defines the light intensity for each wavelength, thus specifying its color.

Components do not necessarily need to be in the 0 ~ 1 range: larger values can model stronger light sources.

Components, however, need to be non-negative.

$L(l, \overrightarrow{lx}) = (0.3, 0.3, 0.3)$

$L(l, \overrightarrow{lx}) = (1, 1, 1)$

$L(l, \overrightarrow{lx}) = (10, 10, 10)$

Notation

As introduced in the previous lessons, the rendering equation must be solved for every color frequency considered (usually, the RGB colors).

Since light color $L(l, \overrightarrow{lx})$ is encoded in a vector, the BRDF function $f_r(x, \overrightarrow{lx}, \omega_r)$ returns a color vector too (we will return on this in the next lesson).

In the following, we will use the * symbol to denote the component-wise product, and a dot · symbol to express the standard scalar product (*dot* product) of two vectors.

$$\boldsymbol{a} * \boldsymbol{b} = (a.R * b.R,\ \ a.G * b.G,\ \ a.B * b.B)$$
$$\boldsymbol{v} \cdot \boldsymbol{u} = v.x * u.x + v.y * u.y + v.z * u.z$$

In GLSL, component wise product is also denoted with symbol **`a*b`**, while dot product is computed with the **`dot(v,u)`** function.

Light models

In this course we will present the three basic direct light models for real time graphics:

- Direct light
- Point light
- Spot light

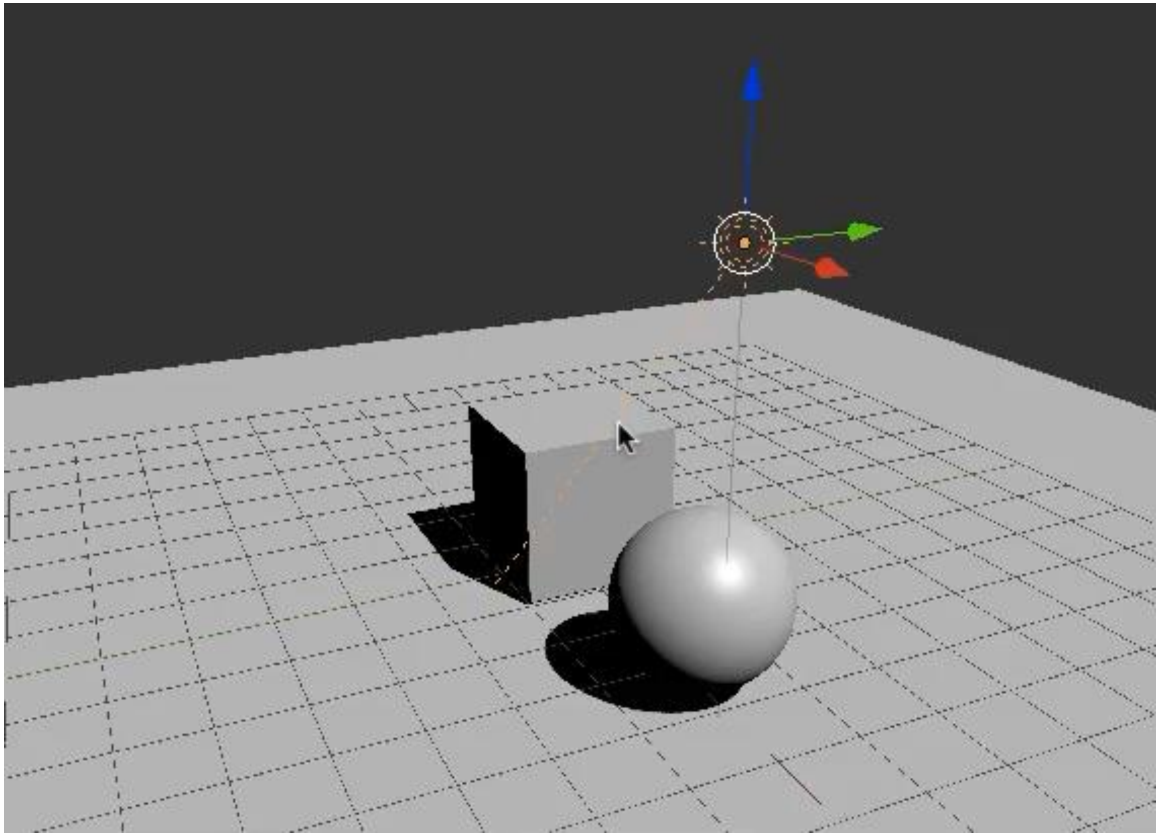
We will also briefly introduce other types of lights with special purposes.

Direct light models

Directional lights are used to model distant sources such as the sunlight.

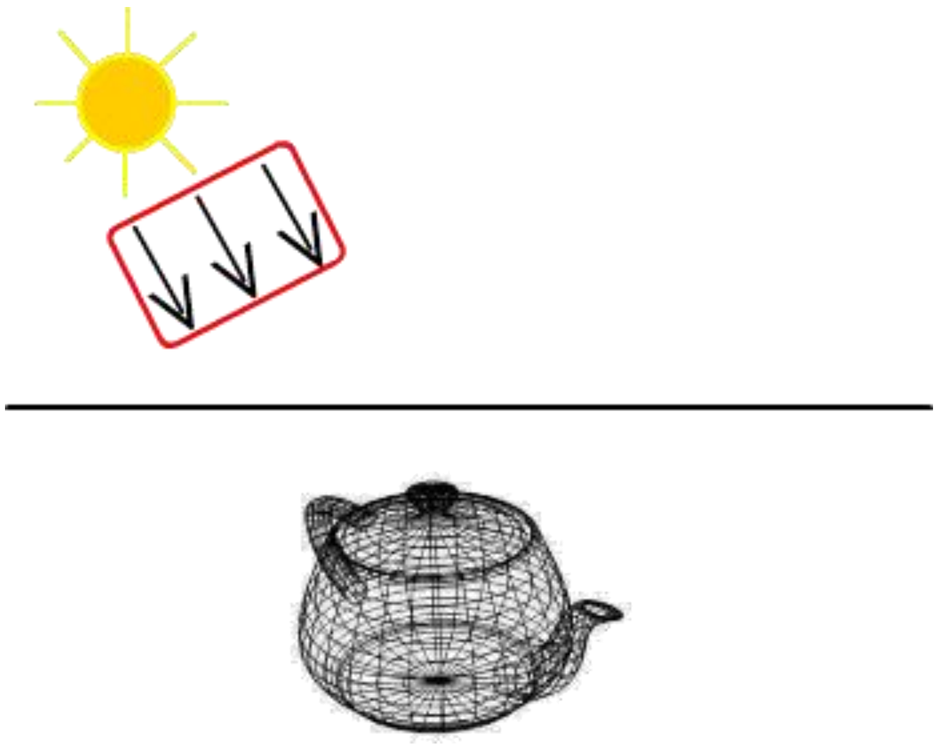
Direct light models

They are sources that are very far away from the objects, so that they uniformly influence the entire scene.



Direct light models

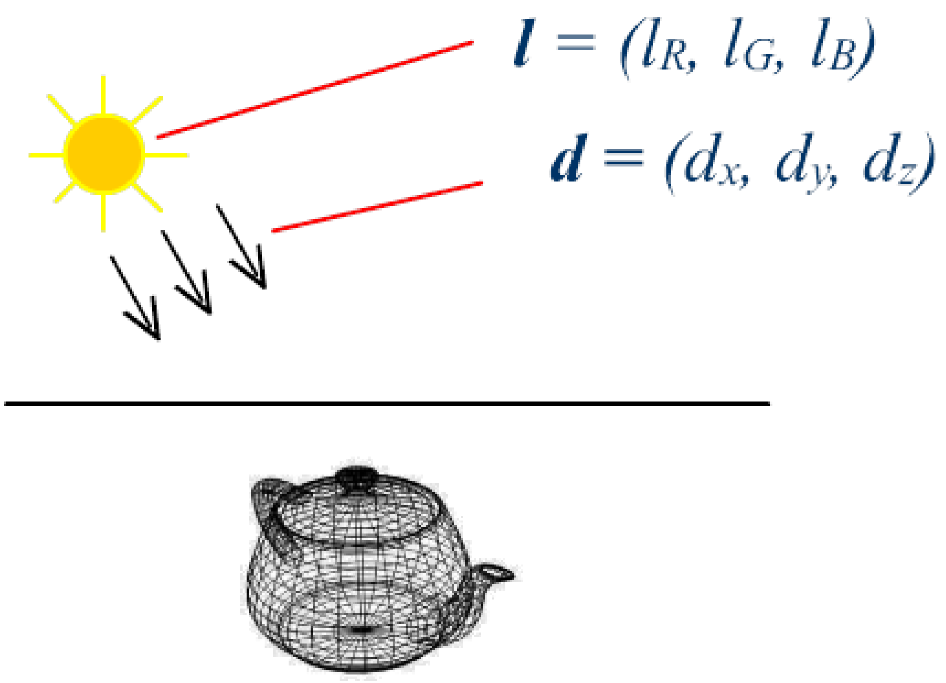
Due to the distance of the source, rays are parallel to each other in all the positions of the space, and constant in color and intensity.



Direct light models

The light direction can then be specified with a constant vector $\boldsymbol{d} = (d_x, d_y, d_z)$ that is independent of the position *x* on the object.

Light color is also specified by a constant vector $\boldsymbol{l} = (l_R, l_G, l_B)$



Direct light models

For every point of an object, the direction of the light and its color are expressed with these two constant values $\boldsymbol{d}$ and $\boldsymbol{l}$:

$$L(x, \omega_r) = \sum_l L(l, \overrightarrow{lx}) * f_r(x, \overrightarrow{lx}, \omega_r)$$

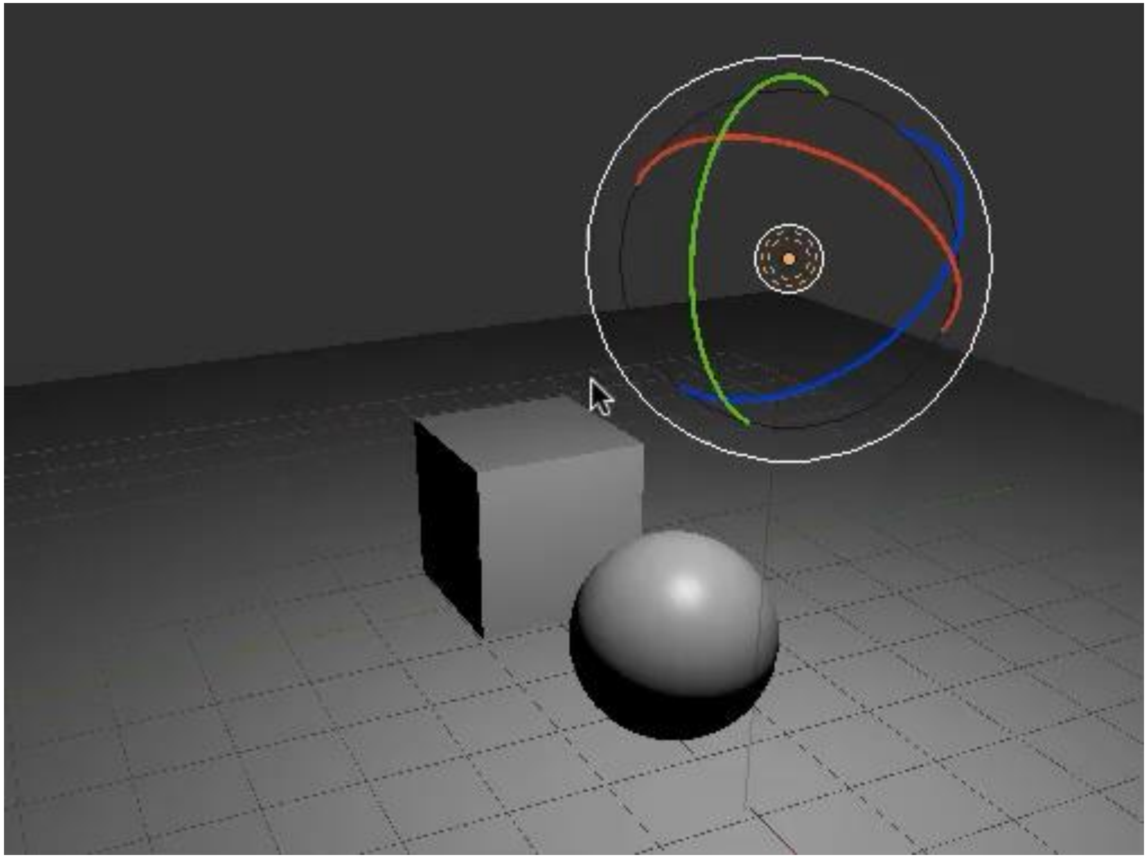
$$\boxed{L(l, \overrightarrow{lx})} = \boldsymbol{l}_{\boldsymbol{l}} \qquad \boxed{\overrightarrow{lx}} = \boldsymbol{d}_{\boldsymbol{l}}$$

In case of a single direct light, the rendering equation reduces to:

$$L(x, \omega_r) = \boldsymbol{l} * f_r(x, \boldsymbol{d}, \omega_r)$$

Point light models

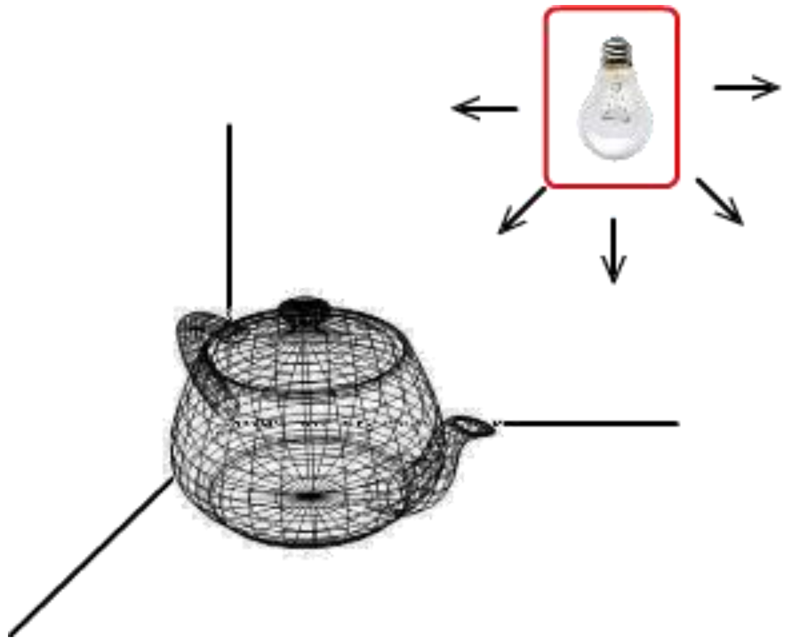
Point lights are sources that emit light from fixed positions in the space, and do not have a specific direction.



Point light models

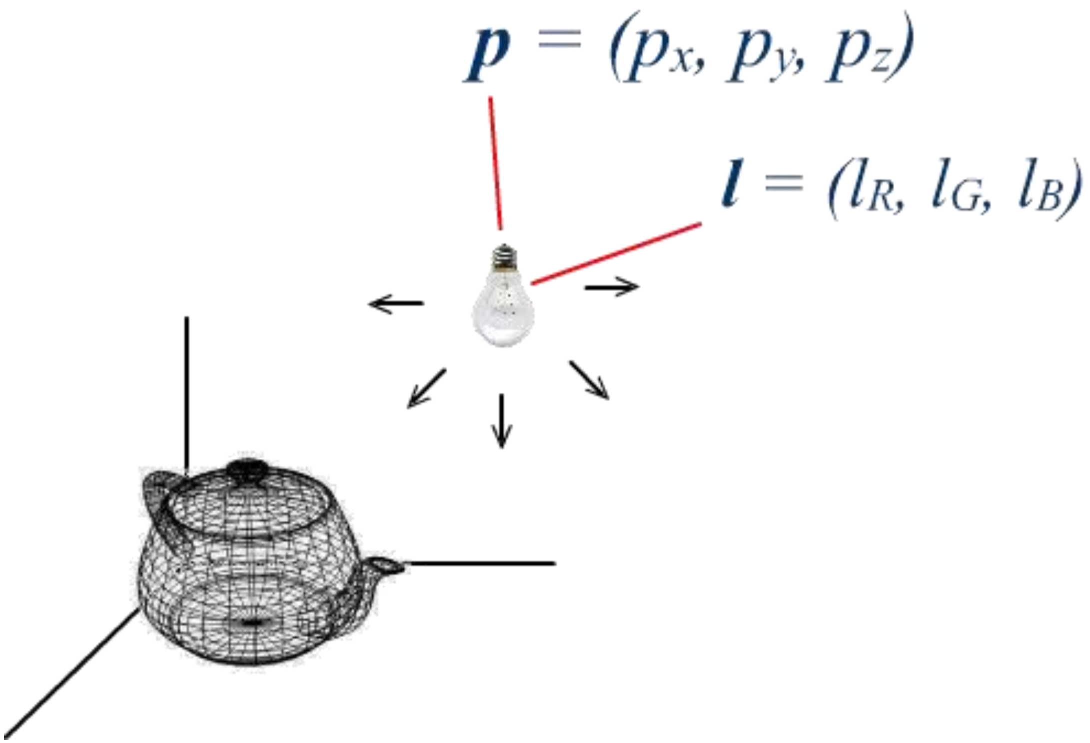
They are used to model sources that emit light in all directions, starting from a given position in the scene.

For example, they can reproduce lamps, bulbs, candles and other omnidirectional light sources.



Point light models

The position $\boldsymbol{p} = (p_x, p_y, p_z)$ and the color $\boldsymbol{l} = (l_R, l_G, l_B)$ characterize a point light.

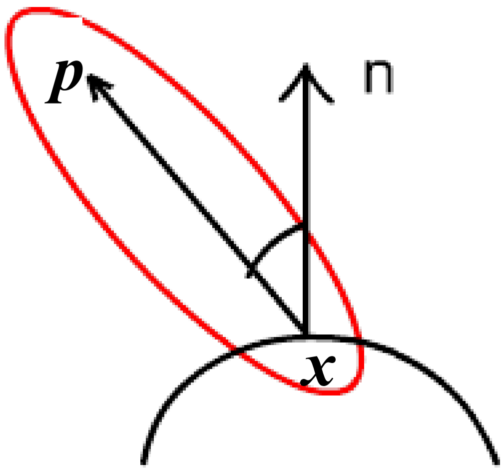


Point light models

The direction goes from point *x* to the center of the light, varying on the surface of the object that it is illuminating.

Note that the light direction should be normalized to make it an unitary vector.

$$\overrightarrow{lx} = \frac{\boldsymbol{p} - \boldsymbol{x}}{|\boldsymbol{p} - \boldsymbol{x}|}$$

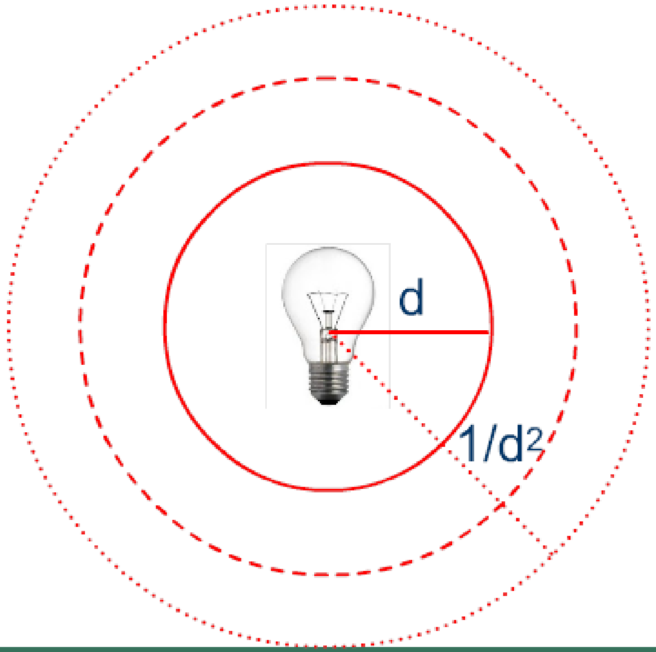


Also note that we write *p* – *x* because the ray is oriented from the object to the light source, as for the direct light case.

Point light models

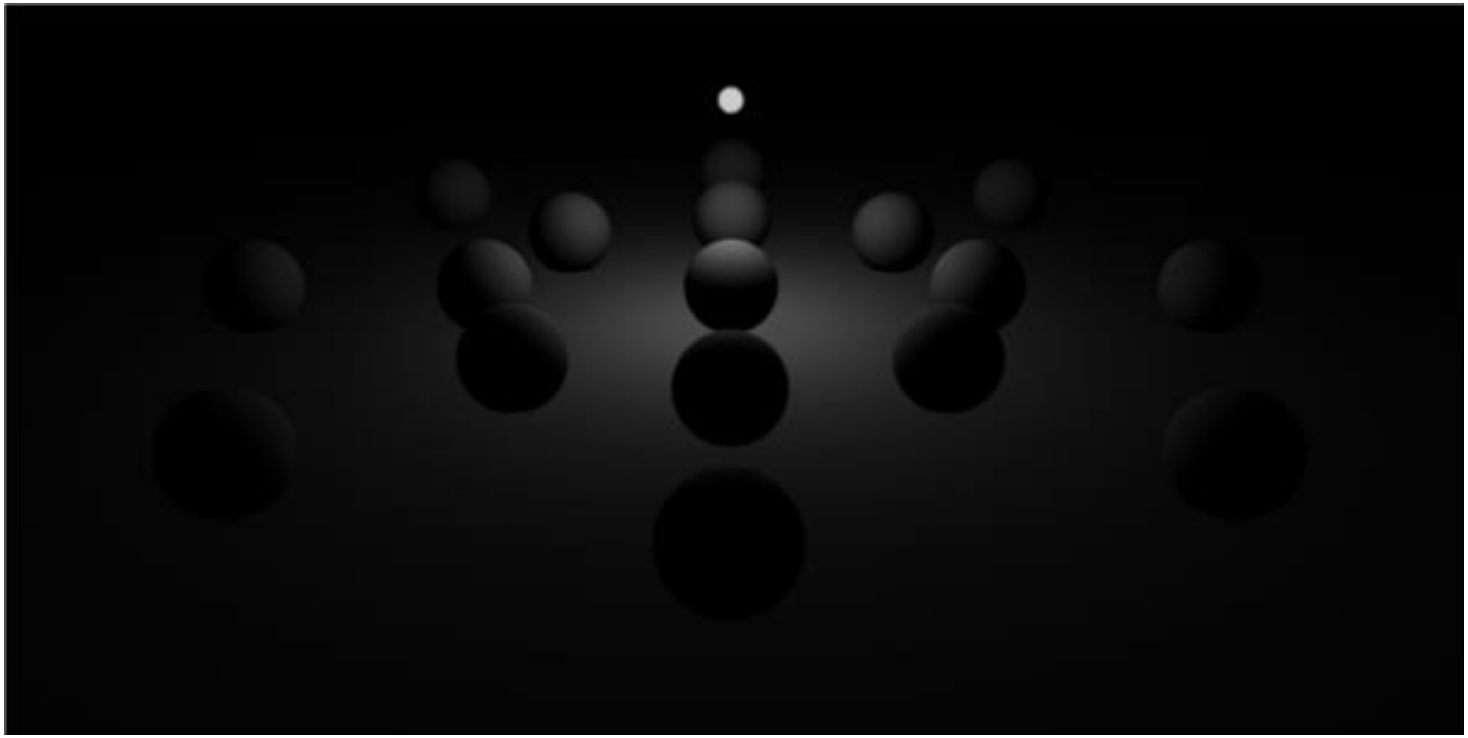
To reproduce the physical properties of light sources, point lights are characterized by a *decay factor*.

Physically, the intensity of a point light reduces at a rate that is proportional to the inverse of the square of the distance.



Point light models

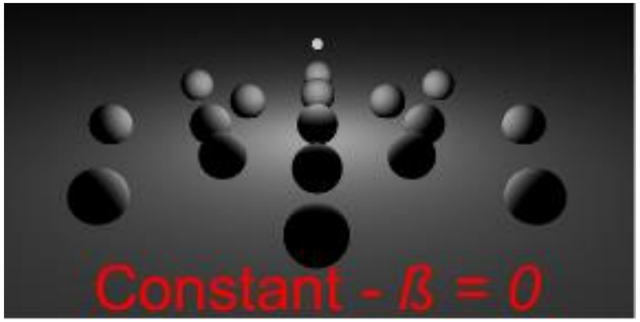
However this might lead to images that are too dark.

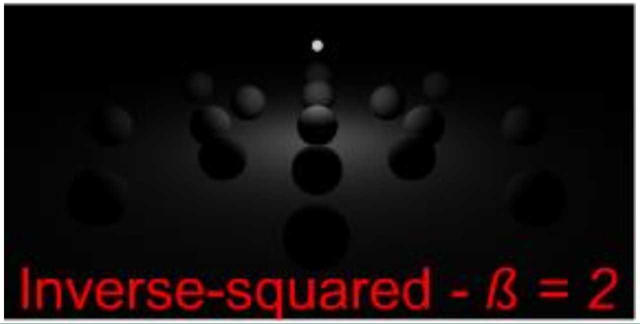


Point light models

For this reason, light models usually allow the user to specify a decay factor *ß* that is either constant, inverse-linear or inverse-squared.

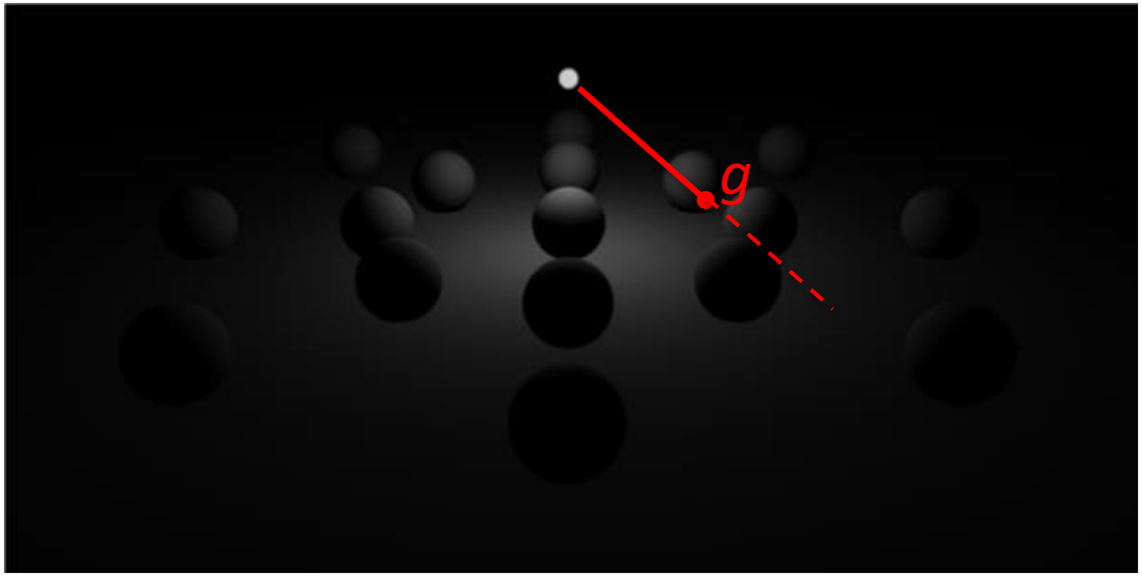
$$L(l, \overrightarrow{lx}) = \left(\frac{g}{|\boldsymbol{p} - \boldsymbol{x}|}\right)^{\beta} \boldsymbol{l}$$





Point light models

The model requires also another value g *that* represents the distance at which the light reduction is exactly 1 : intensity will be higher than $\boldsymbol{l}$ for distances shorter than g, and it will dim for longer distances.



$$L(l, \overrightarrow{lx}) = \left(\frac{g}{|\boldsymbol{p} - \boldsymbol{x}|}\right)^{\beta} \boldsymbol{l}$$

Point light models

To summarize, the direction of the light and the color used in the rendering equations become:

$$L(x, \omega_r) = \sum_{l} L(l, \overrightarrow{lx}) f_r(x, \overrightarrow{lx}, \omega_r)$$

$$\boxed{L(l, \overrightarrow{lx})} = \boldsymbol{l}\left(\frac{g}{|\boldsymbol{p}-\boldsymbol{x}|}\right)^{\beta} \qquad \boxed{\overrightarrow{lx}} = \frac{\boldsymbol{p}-\boldsymbol{x}}{|\boldsymbol{p}-\boldsymbol{x}|}$$

In case of a single point light, the rendering equation for one pixel is:

When no decay is considered

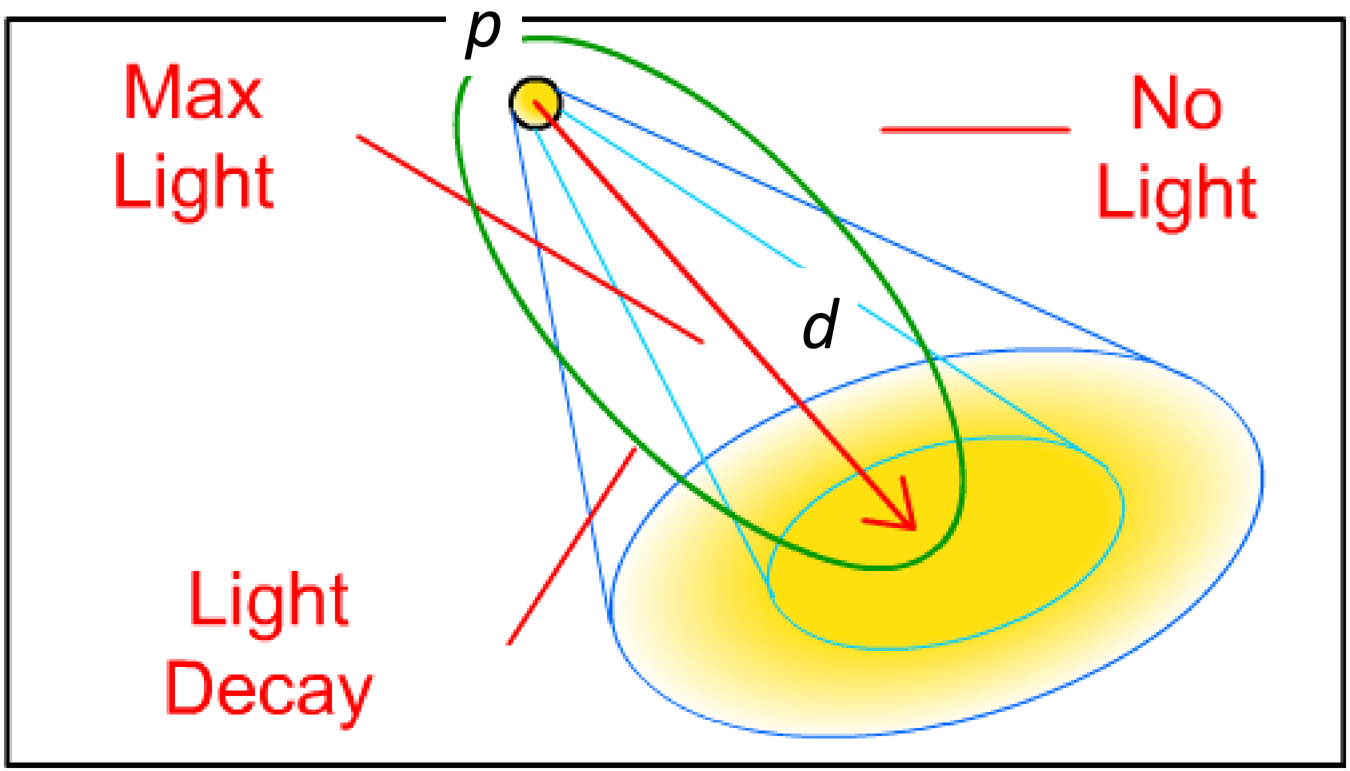
$$L(x, \omega_r) = \boldsymbol{l}\left(\frac{g}{|\boldsymbol{p}-\boldsymbol{x}|}\right)^{\beta} * f_r\left(x, \frac{\boldsymbol{p}-\boldsymbol{x}}{|\boldsymbol{p}-\boldsymbol{x}|}, \omega_r\right) = \boxed{\boldsymbol{l} * f_r\left(x, \frac{\boldsymbol{p}-\boldsymbol{x}}{|\boldsymbol{p}-\boldsymbol{x}|}, \omega_r\right)}$$

Spot light models

Spot lights are special projectors that are used to illuminate specific objects or locations.

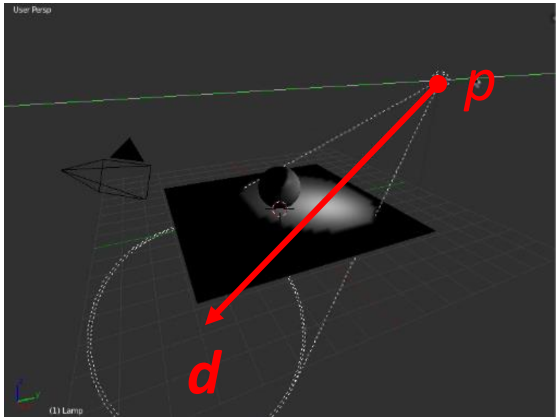
Spot light models

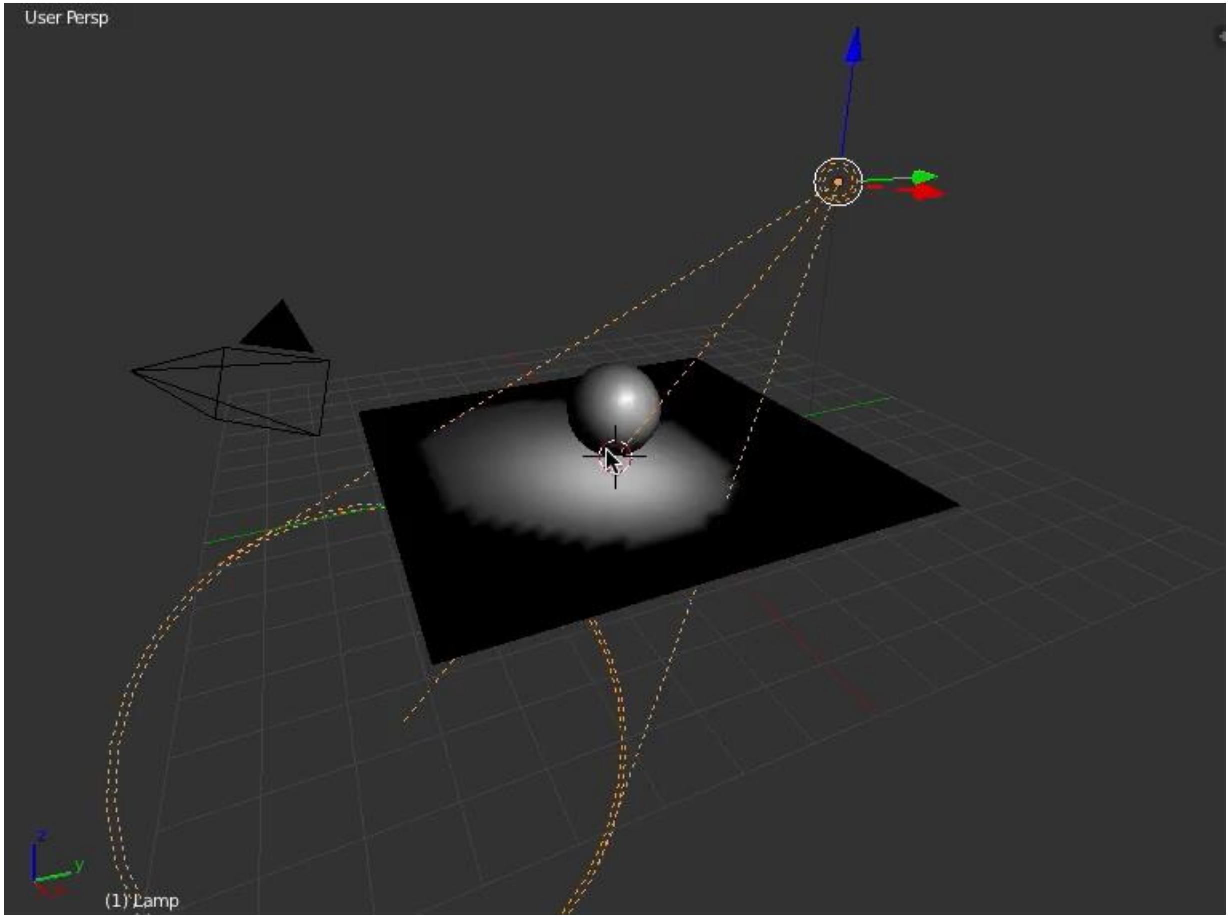
They are conic sources characterized by a direction *d* and a position *p*.



Spot light models

In particular, Spot lights emit in direction *d*, starting from point *p*.

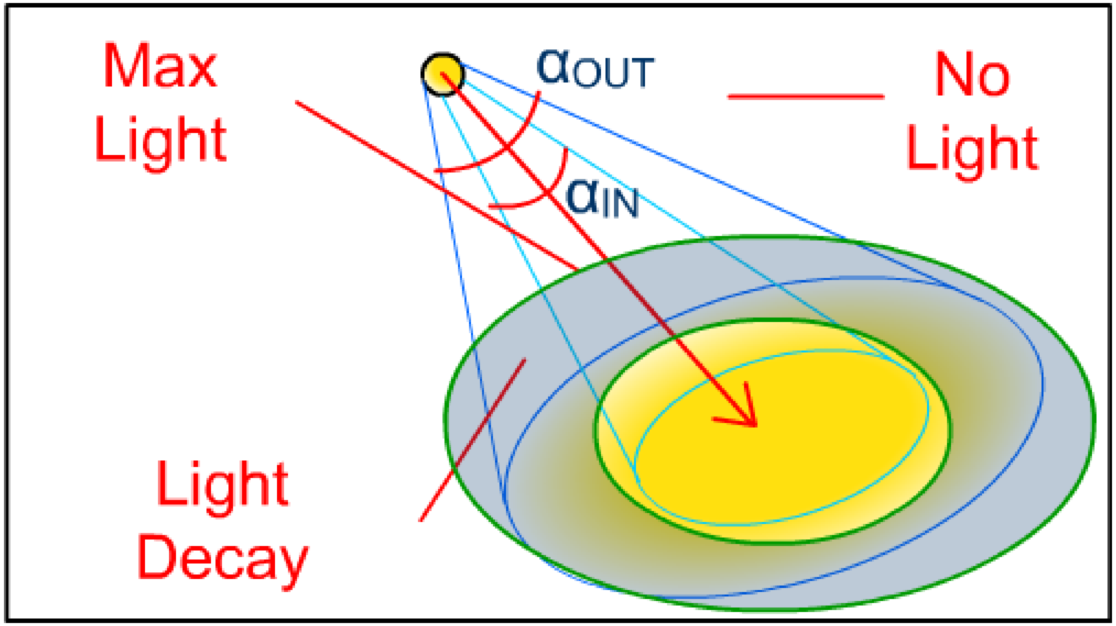




Spot light models

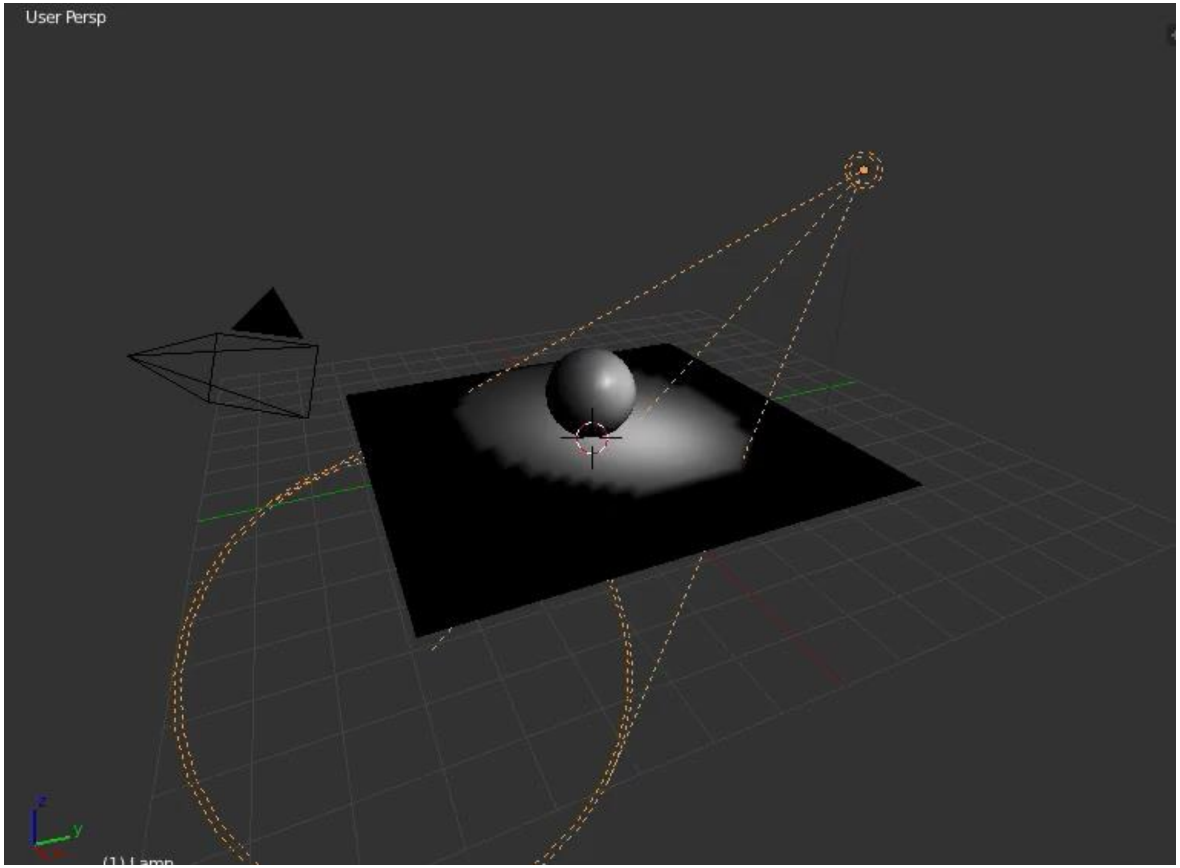
Spot lights are characterized by two angles α_{IN} and α_{OUT} that divide the illuminated area into three zones: constant (inside α_{IN}), decay (between α_{IN} and α_{OUT}) and absent (outside α_{OUT}).

In the light decay zone between α_{IN} and α_{OUT}, the light intensity decreases linearly from the inner to the outer angle.



Spot light models

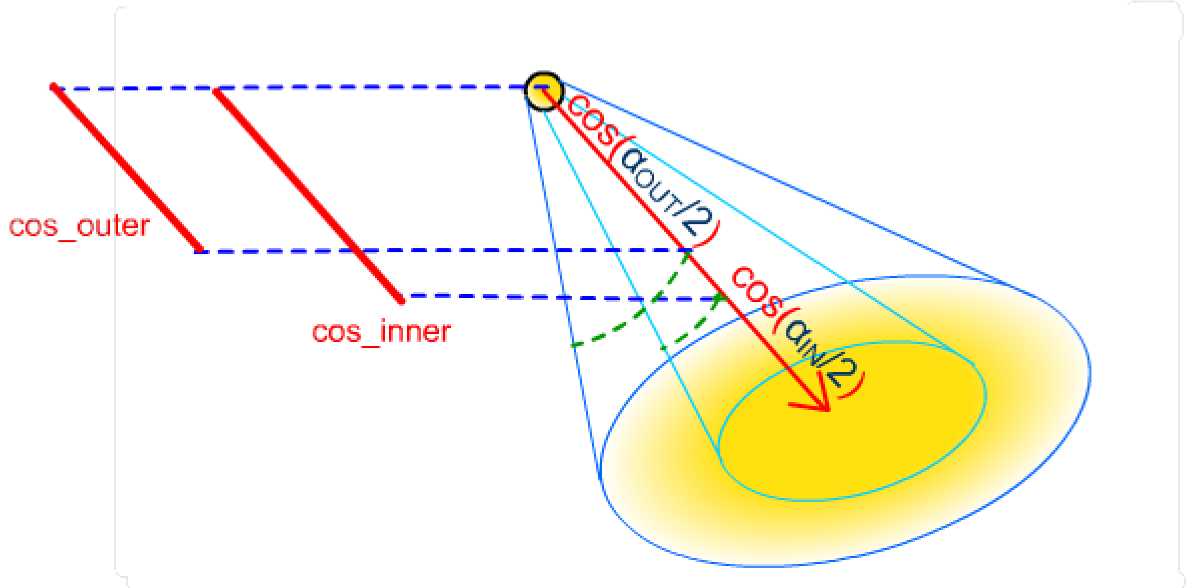
Using these two parameters, the light can be sized to concentrate its effect on a given subject.



Spot light models

For the implementation of the spot lights, usually the *cosine of the half-angles* of the *inner* and *outer cones* c_{in} and c_{out} are used.

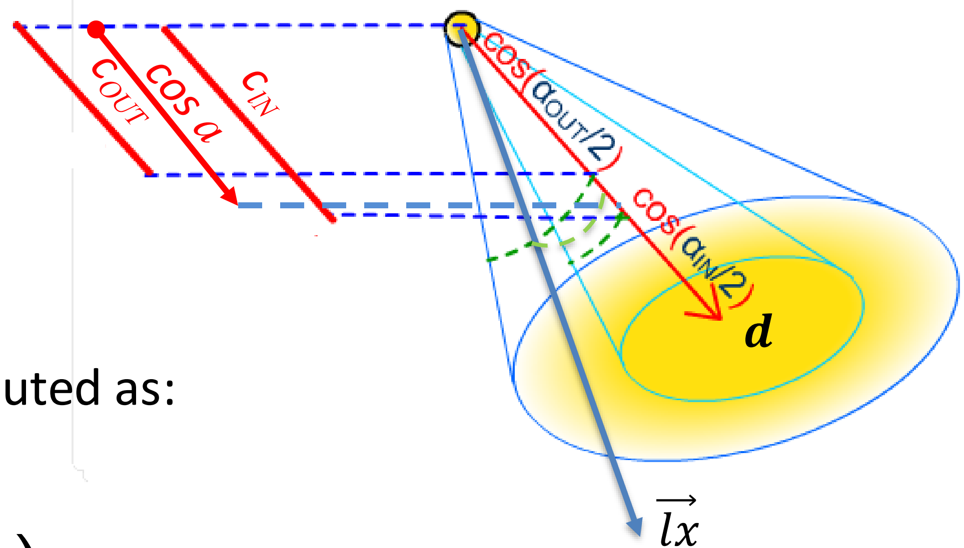
Note that the cosine of the inner angle is greater than the one of the outer angle.



Spot light models

The cosine of between the light direction vector $\overrightarrow{lx}$ and the direction of the spot $\boldsymbol{d}$ can be computed by performing the dot product between the two.

$$\cos\alpha = \overrightarrow{lx} \cdot \boldsymbol{d}$$



The cone dimming effect is computed as:

$$clamp\left(\frac{\cos\alpha - c_{OUT}}{c_{IN} - c_{OUT}}\right)$$

With $clamp(y) = \begin{cases} 0 & y < 0 \\ y & y \in [0,1] \\ 1 & y > 1 \end{cases}$

Spot light models

Spot lights are implemented by confining other light sources with the dimming term just introduced.

In particular, they inherit the light direction $\overrightarrow{lx}_{\mathbf{0}}$ from the model they derive from, and modulate their color $L_0(l,\overrightarrow{lx})$ with the dimming term.

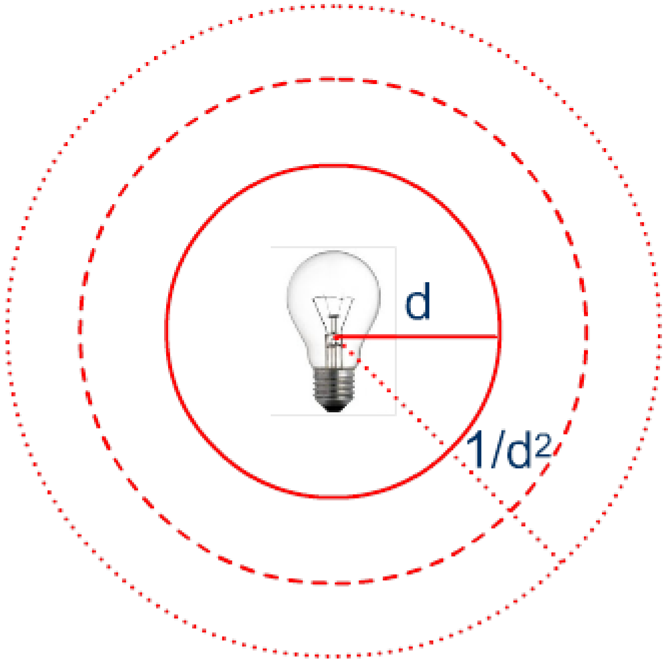
$$L(l,\overrightarrow{lx}) = L_0(l,\overrightarrow{lx}) \cdot clamp\left(\frac{\frac{\boldsymbol{p}-\boldsymbol{x}}{|\boldsymbol{p}-\boldsymbol{x}|}\cdot \boldsymbol{d} - c_{OUT}}{c_{IN}-c_{OUT}}\right) \qquad \overrightarrow{lx} = \overrightarrow{lx}_{\mathbf{0}}$$

The most popular spot-light implementation adds the dimming factor to the point light: in the following we will focus only on this case.

Spot light models

In this case, spot lights are also characterized by the decay factor β, the target distance g and the color vector $\boldsymbol{l}$.

Light direction is then computed as for the point light.



$$\boldsymbol{L_0}(\boldsymbol{l}, \overrightarrow{\boldsymbol{lx}}) = \boldsymbol{l}\left(\frac{g}{|\boldsymbol{p}-\boldsymbol{x}|}\right)^{\beta}$$

$$\overrightarrow{lx}_{\boldsymbol{0}} = \frac{\boldsymbol{p}-\boldsymbol{x}}{|\boldsymbol{p}-\boldsymbol{x}|}$$

Spot light models

To summarize, the direction of the light and the color used in the rendering equations are the following:

$$L(x, \omega_r) = \sum_l L(l, \overrightarrow{lx}) f_r(x, \overrightarrow{lx}, \omega_r)$$

$$\boxed{L(l, \overrightarrow{lx})} = \boldsymbol{l} \left(\frac{g}{|\boldsymbol{p} - \boldsymbol{x}|} \right)^{\beta} \cdot clamp \left(\frac{\frac{\boldsymbol{p} - \boldsymbol{x}}{|\boldsymbol{p} - \boldsymbol{x}|} \cdot \boldsymbol{d} - c_{OUT}}{c_{IN} - c_{OUT}} \right) \qquad \boxed{\overrightarrow{lx}} = \frac{\boldsymbol{p} - \boldsymbol{x}}{|\boldsymbol{p} - \boldsymbol{x}|}$$

Spot light models

In case of a spot light without decay, the rendering equation for one pixel is:

$$L(x, \omega_r) = \boldsymbol{l} \cdot clamp\left(\frac{\frac{\boldsymbol{p}-\boldsymbol{x}}{|\boldsymbol{p}-\boldsymbol{x}|} \cdot \boldsymbol{d} - c_{OUT}}{c_{IN} - c_{OUT}}\right) * f_r\left(x, \frac{\boldsymbol{p}-\boldsymbol{x}}{|\boldsymbol{p}-\boldsymbol{x}|}, \omega_r\right)$$

If we consider a spot-light that extends direct lights, it further simplifies:

$$L(x, \omega_r) = \boldsymbol{l} \cdot clamp\left(\frac{\frac{\boldsymbol{p}-\boldsymbol{x}}{|\boldsymbol{p}-\boldsymbol{x}|} \cdot \boldsymbol{d} - c_{OUT}}{c_{IN} - c_{OUT}}\right) * f_r(x, \boldsymbol{d}, \omega_r)$$

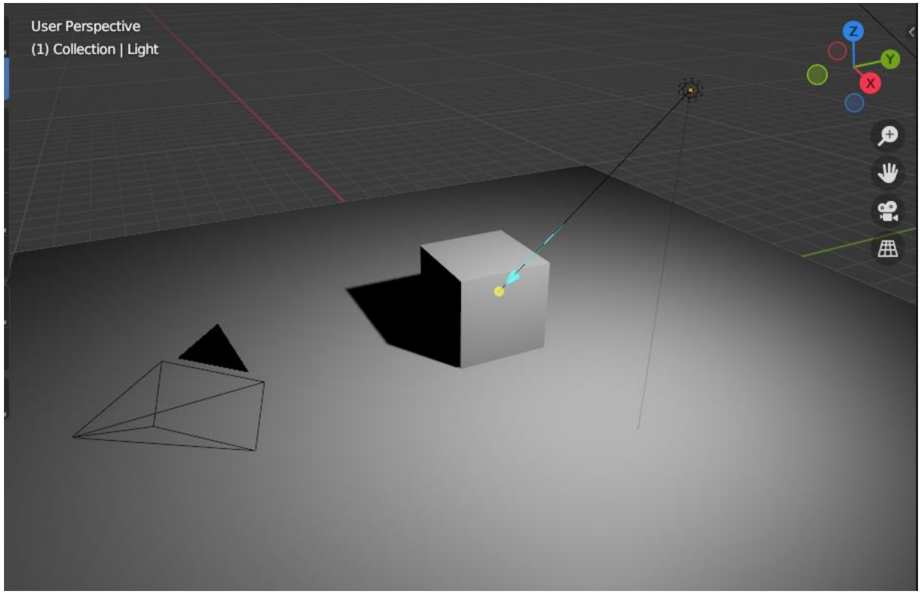
This model can be used, without a visible difference with respect to the one based on the point light, when the outer cone is particularly narrow.

Special light models: the cosine light

When the inner cone reduces to zero, and the outer cone is maximized (i.e. cosine equal to one), the equation become again very simple:

$$L(x, \omega_r) = \boldsymbol{l} \cdot clamp\left(\frac{\boldsymbol{p} - \boldsymbol{x}}{|\boldsymbol{p} - \boldsymbol{x}|} \cdot \boldsymbol{d}\right) * f_r\left(x, \frac{\boldsymbol{p} - \boldsymbol{x}}{|\boldsymbol{p} - \boldsymbol{x}|}, \omega_r\right)$$

This special light model is sometimes called the "cosine" light model. Although being very simple, it produces interesting diffuse lighting effects.



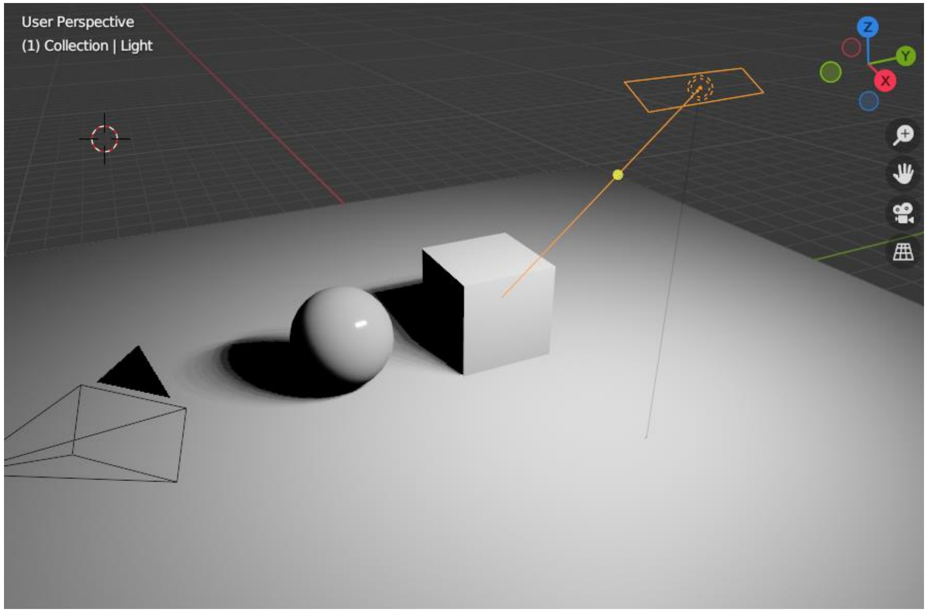
Special light models: area lights

Most of realistic light sources do not have a point origin.

Area lights aim at capturing the shape of the light in the scene.

Unfortunately, due the fact that the light shape must be considered, single sources can no longer be considered, and a full integral should be used even in scanline rendering.

Current solutions for reproducing area lights are then based on specific approximation to the integral, and cannot be decoupled from the BRDF of the surfaces.



$$L(x,\omega_r) = \sum_l L_e(l,\overrightarrow{lx}) f_r(x,\overrightarrow{lx},\omega_r)$$

$$L(x,\omega_r) = \int L_e(l,\overrightarrow{lx}) f_r(x,\overrightarrow{lx},\omega_r)\,dl$$

Marco Gribaudo

Associate Professor

CONTACTS

Tel. +39 02 2399 3568
marco.gribaudo@polimi.it
https://www.deib.polimi.it/eng/home-page

(Remember to use the phone, since mails might require a lot of time to be answered. Microsoft Teams messages might also be faster than regular mails)