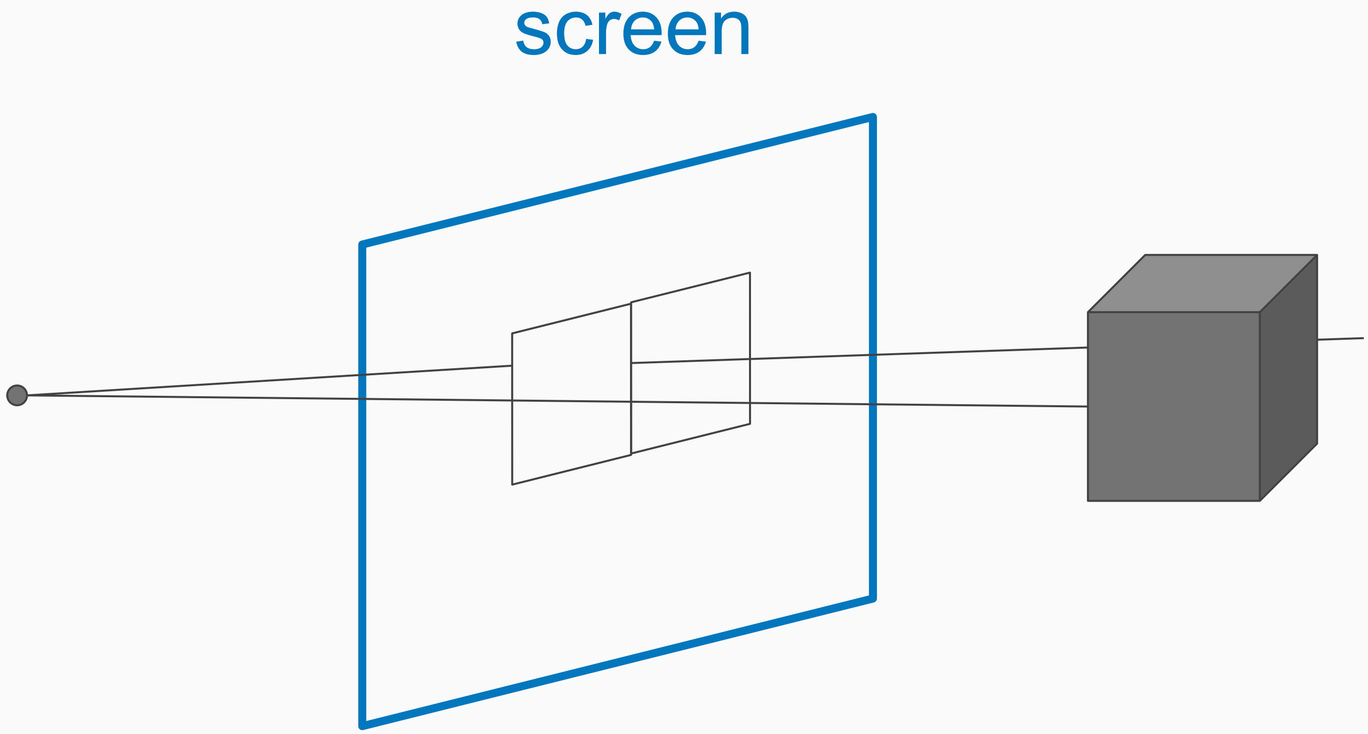
Raytracing from first principles

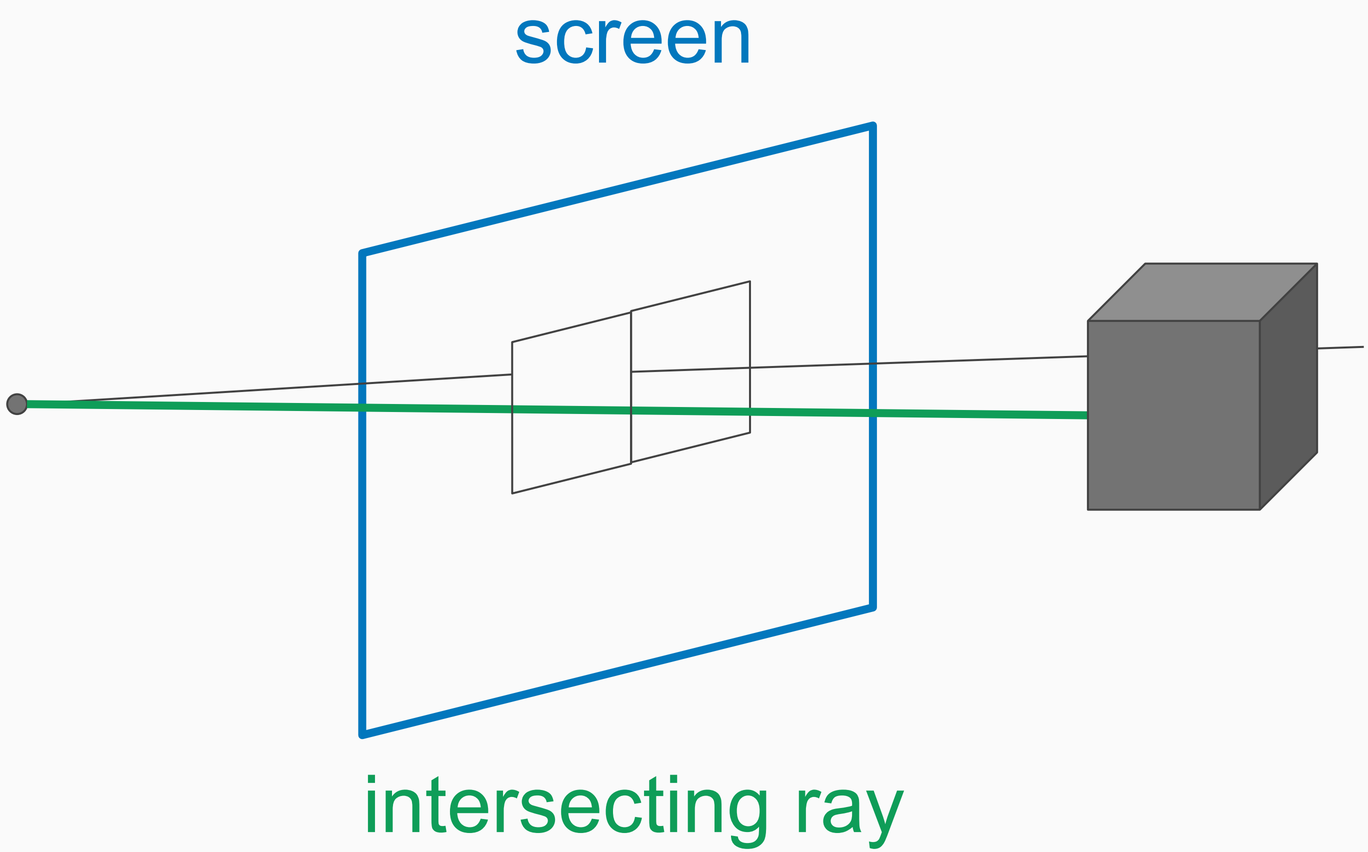
Alexander Vandenbroucke

Raytracing

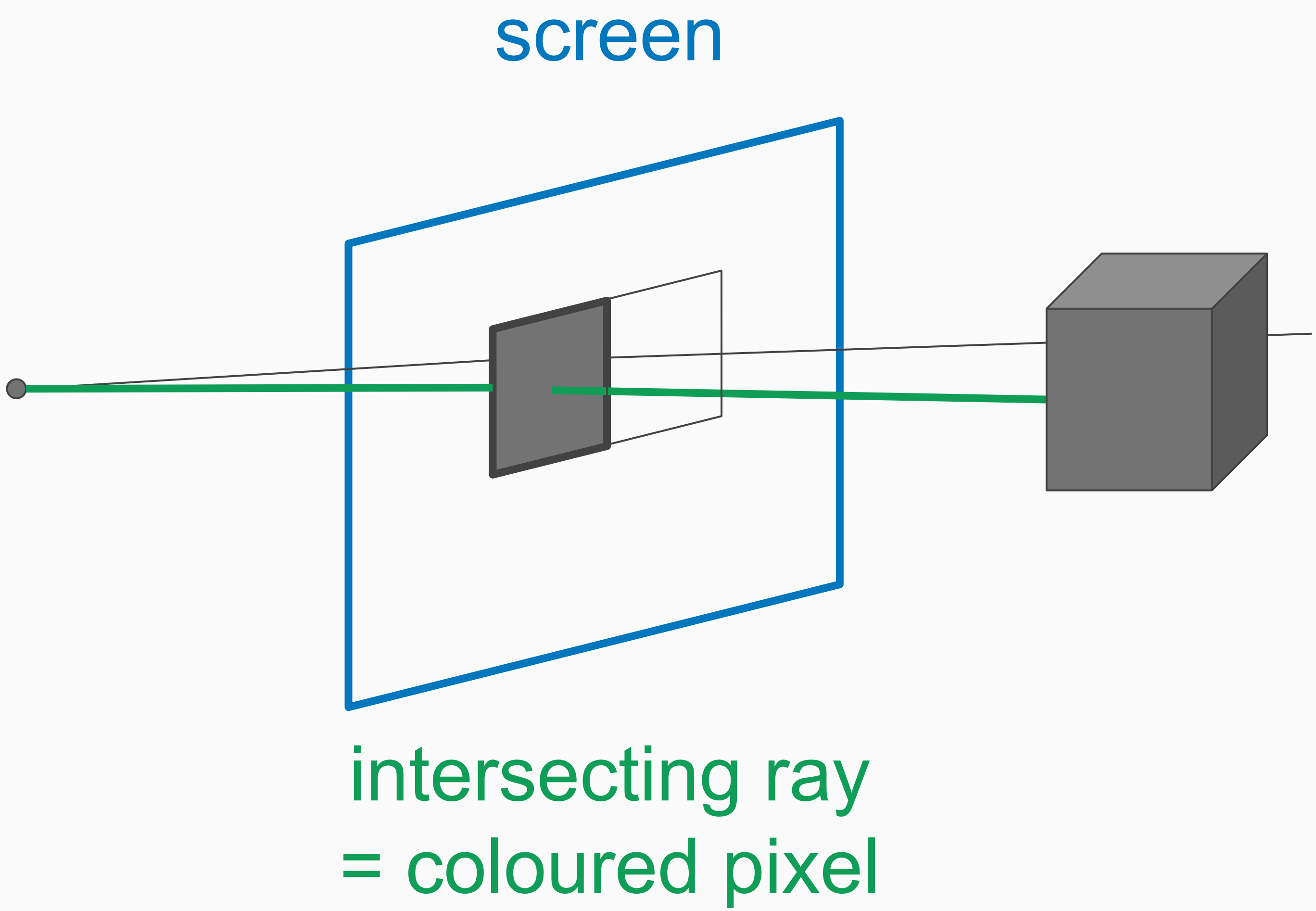
Raytracing



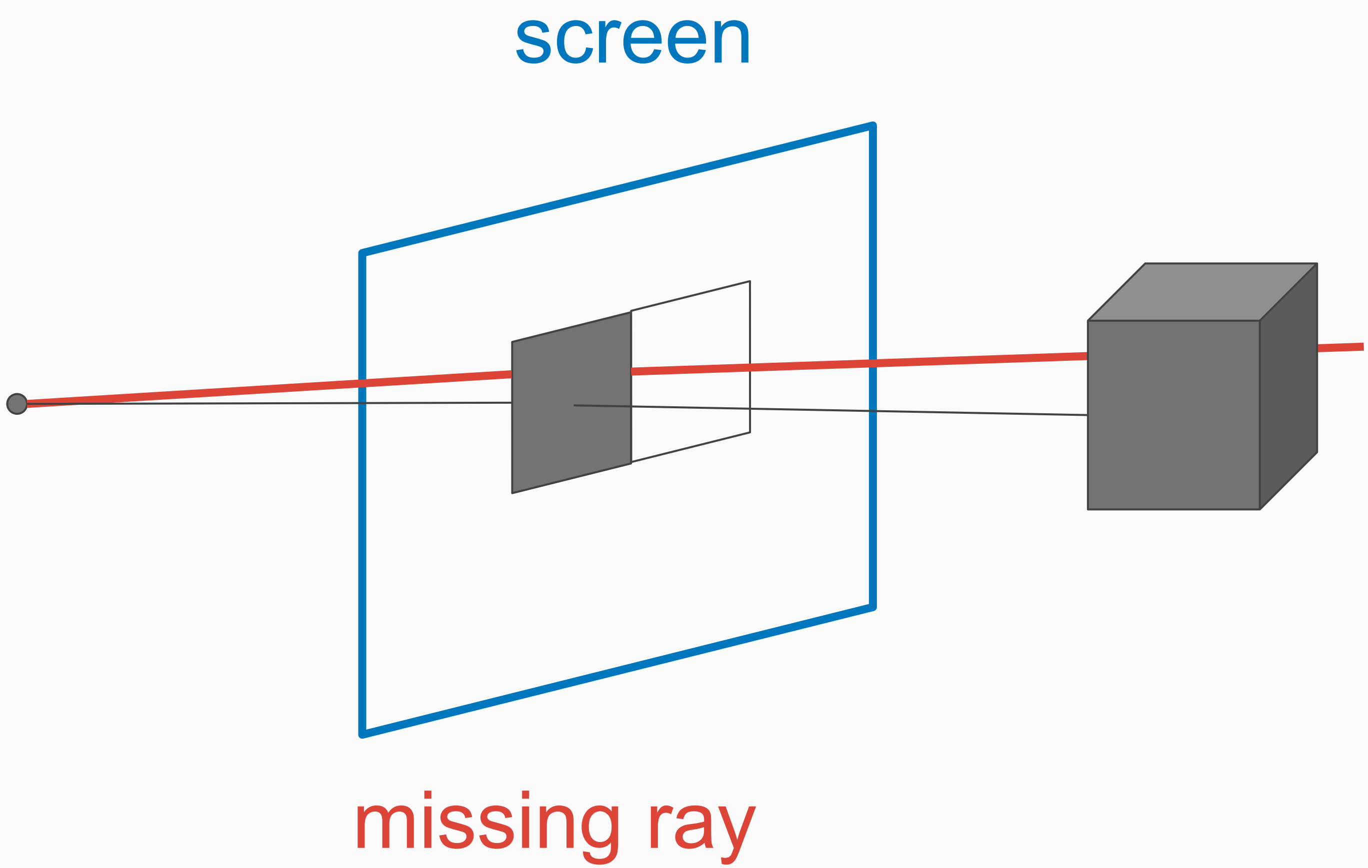
Raytracing



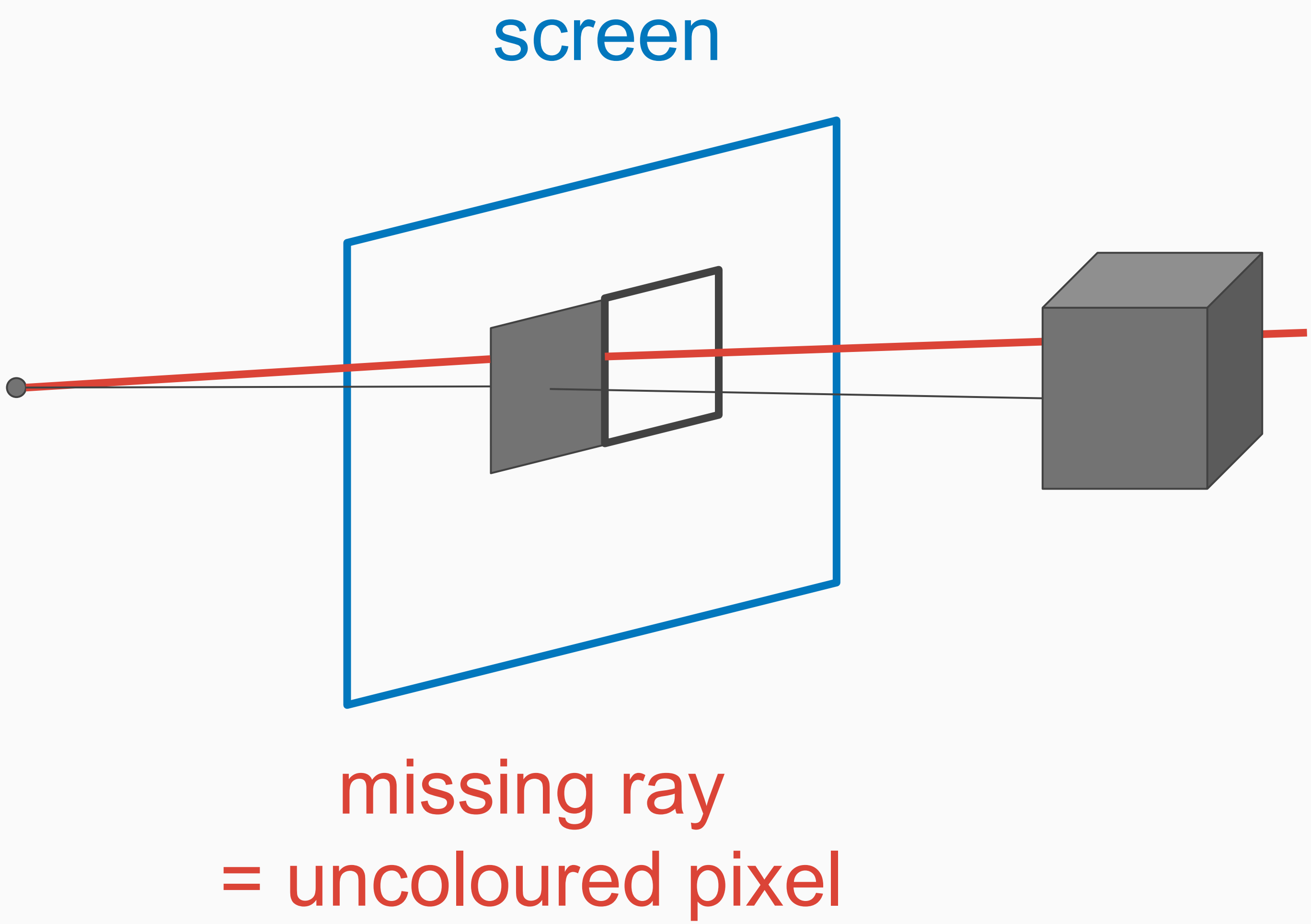
Raytracing



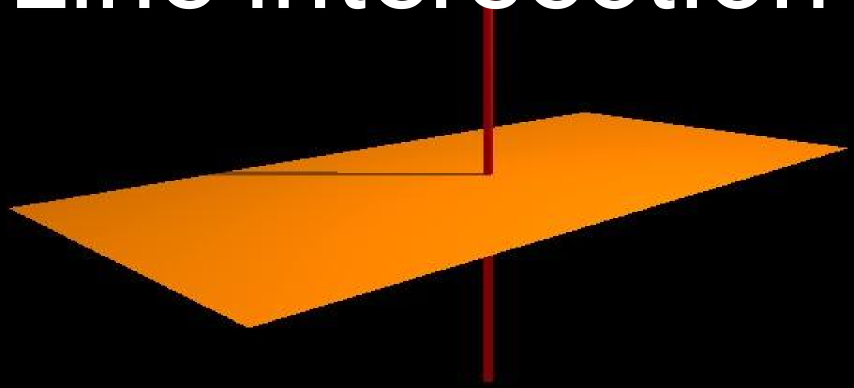
Raytracing



Raytracing

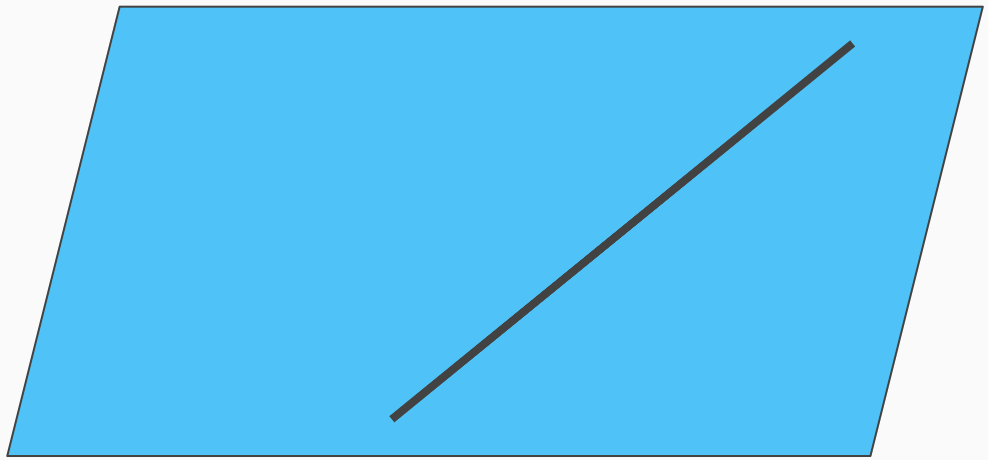


Plane - Line intersection



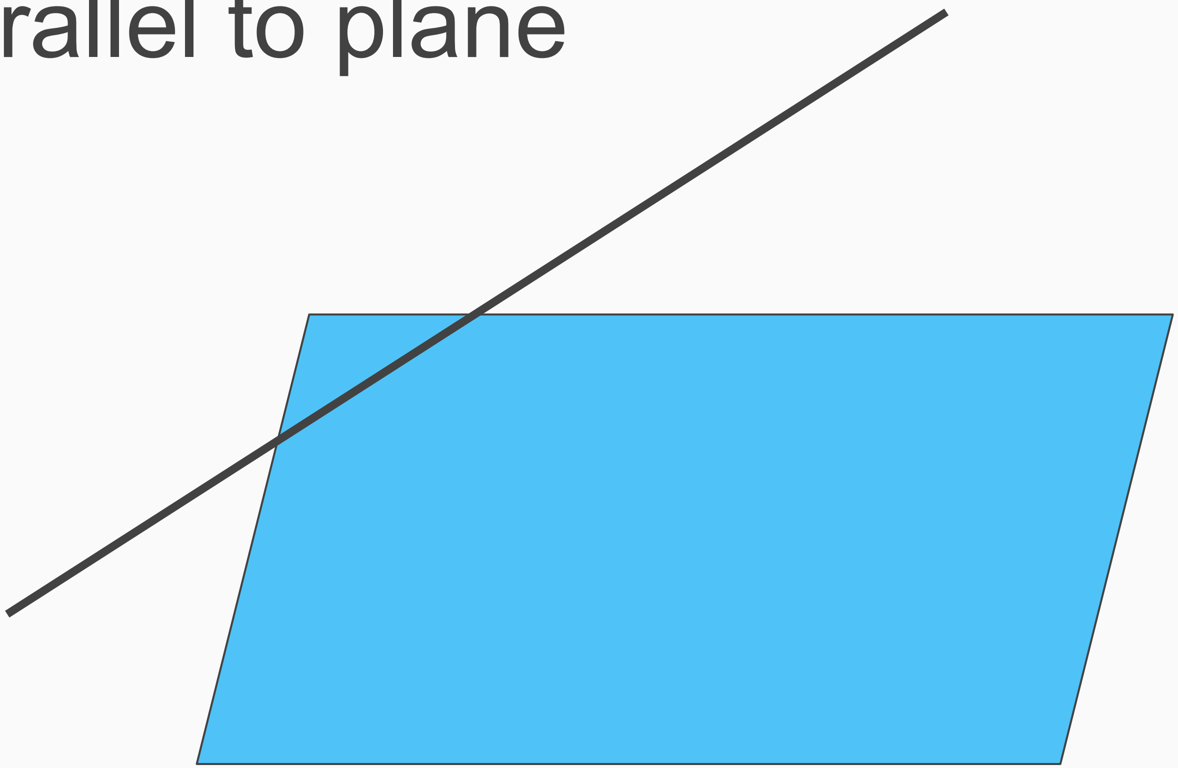
Plane - Line Intersection

Line in the plane



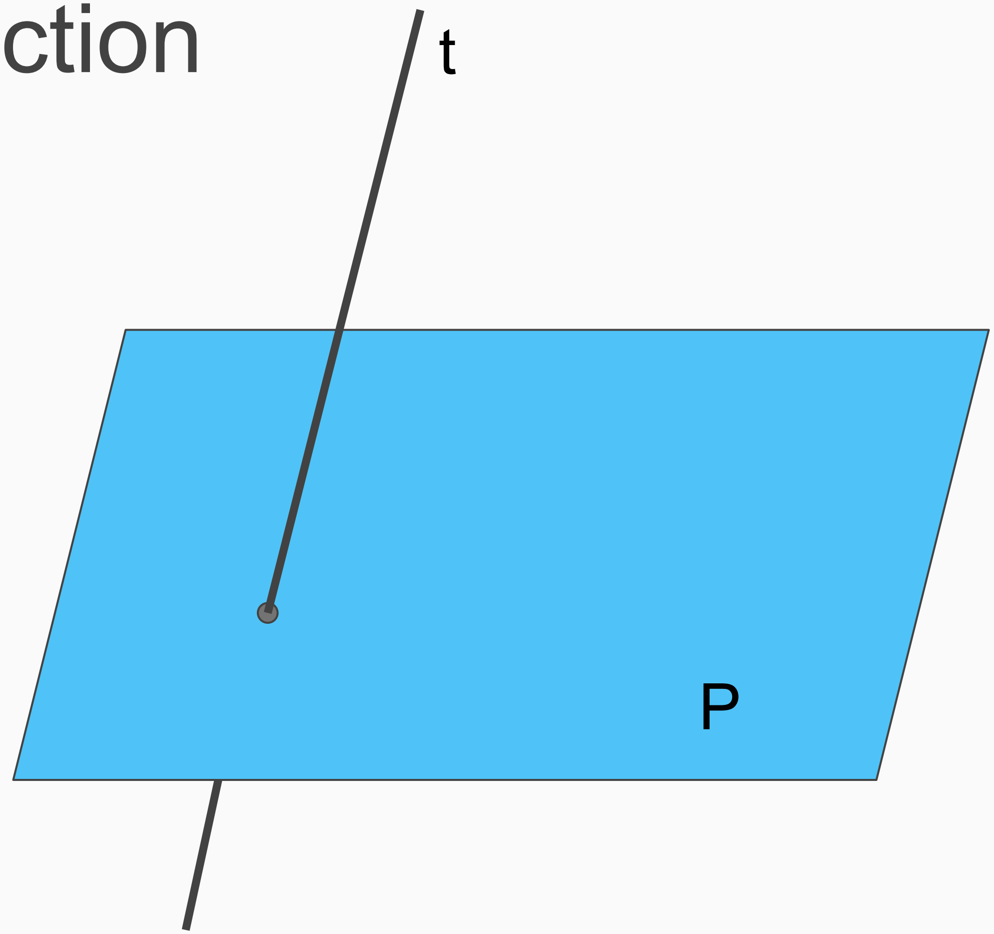
Plane - Line Intersection

Line parallel to plane



Plane - Line Intersection

One Intersection



Plane - Line Intersection

A line through (x_1,y_1,z_1), along (u,v,w)

$$t \leftrightarrow (x - x_1)/u = (y - y_1)/v = (z - z_1)/w$$

Plane - Line Intersection

A line through (x_1,y_1,z_1), along (u,v,w)

$$t \leftrightarrow (x - x_1)/u = (y - y_1)/v = (z - z_1)/w$$

A plane through (x_2,y_2,z_2), along **u** and **v**

$$P \leftrightarrow ax + by + cz + d = 0 \textit{ where}$$

$$(a,b,c) = \mathbf{u} \times \mathbf{v}$$

$$-d = ax_2 + by_2 + cz_2$$

Plane - Line Intersection

A line through (x_1,y_1,z_1), along (u,v,w)

$$t \leftrightarrow (x - x_1)/u = (y - y_1)/v = (z - z_1)/w$$

A plane through (x_2,y_2,z_2), along **u** and **v**

$P \leftrightarrow ax + by + cz + d = 0$ *where*

$(a,b,c) = \mathbf{u} ✖ \mathbf{v}$

$-d = ax_2 + by_2 + cz_2$

cross product: the vector perpendicular to **u** and **v**

Plane - Line Intersection

Now solve for x,y,z:

$$\begin{cases} (x - x_1)/u = (y - y_1)/v = (z - z_1)/w \\ ax + by + cz + d = 0 \end{cases}$$

$$\begin{cases} x = u(z - z_1)/w + x_1; \quad y = v(z - z_1)/w + y_1 \\ ax + by + cz - d = 0 \end{cases}$$

...

$$\begin{cases} \underline{x} = u(z - z_1)/w + x_1; \quad \underline{y} = v(z - z_1)/w + y_1 \\ \underline{z} = (-d - ax_1 - by_1 + (\alpha - c)z_1) / \alpha \\ \alpha = (au + bv + cw) / w \end{cases}$$

Plane - Line Intersection

Now solve for x,y,z:

$$\begin{cases} (x - x_1)/u = (y - y_1)/v = (z - z_1)/w \\ ax + by + cz + d = 0 \end{cases}$$

$x = u(z - z_1)/w + x_1$; $y = v(z - z_1)/w + y_1$

$\alpha = (a,b,c) \, . \, (u,v,w) \, / \, w$

$= \text{normal} \, . \, \text{ray direction} \, / \, w$

$z_1)/w + y_1$

$z = (-d - ax_1 - by_1 + (\alpha - c)z_1) \, / \, \alpha$

$\alpha = (au + bv + cw) \, / \, w$

Plane - Line Intersection

Now solve for x,y,z:

$$\begin{cases} (x - x_1)/u = (y - y_1)/v = (z - z_1)/w \\ ax + by + cz + d = 0 \end{cases}$$

$x = u(z - z_1)/w + x_1$; $y = v(z - z_1)/w + y_1$

$\alpha = (a,b,c) . (u,v,w) / w$
$= \text{normal . ray direction} / w$
$= 0$

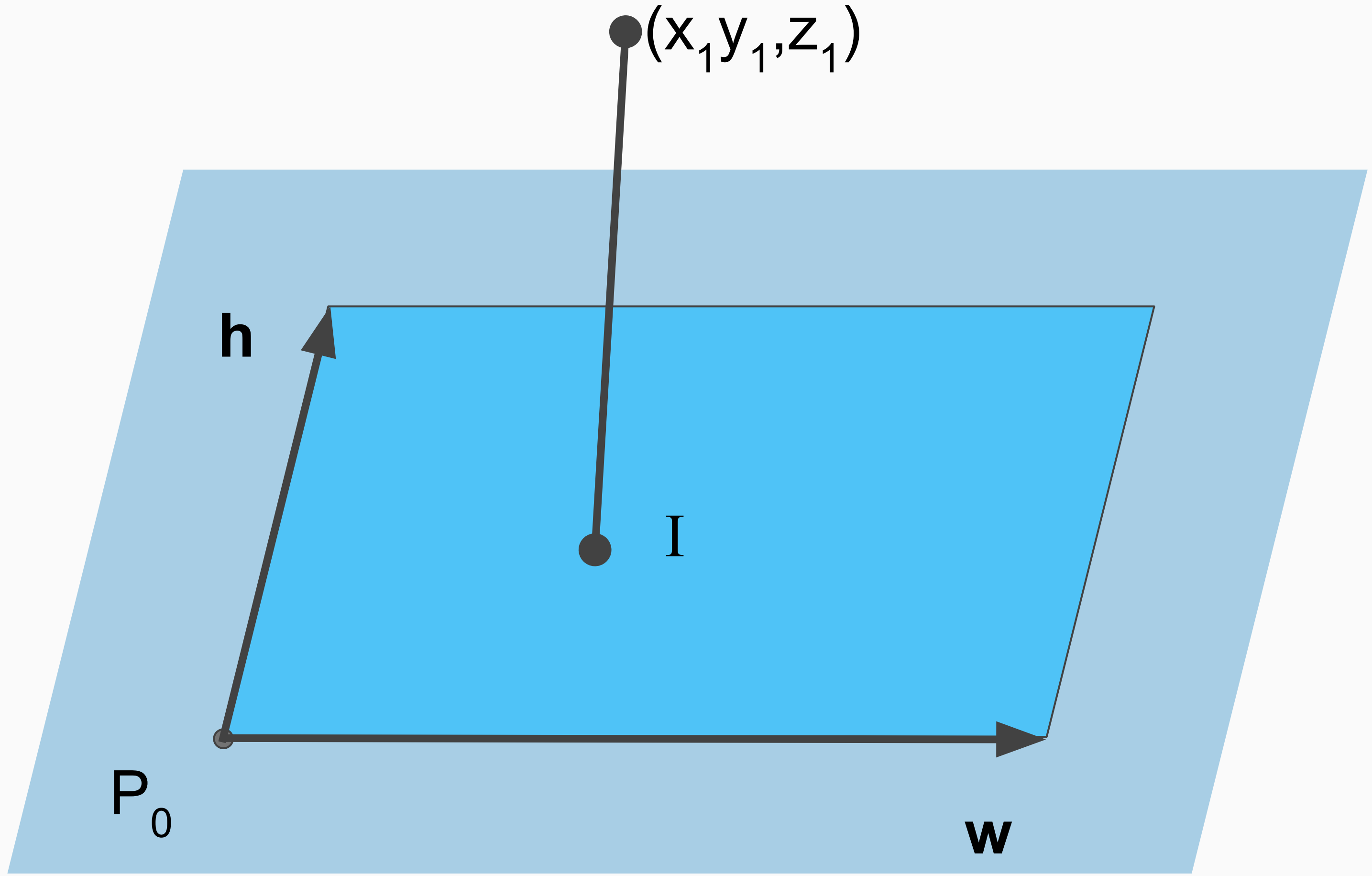
$\Leftrightarrow$ normal $\perp$ ray direction

$z_1)/w + y_1$

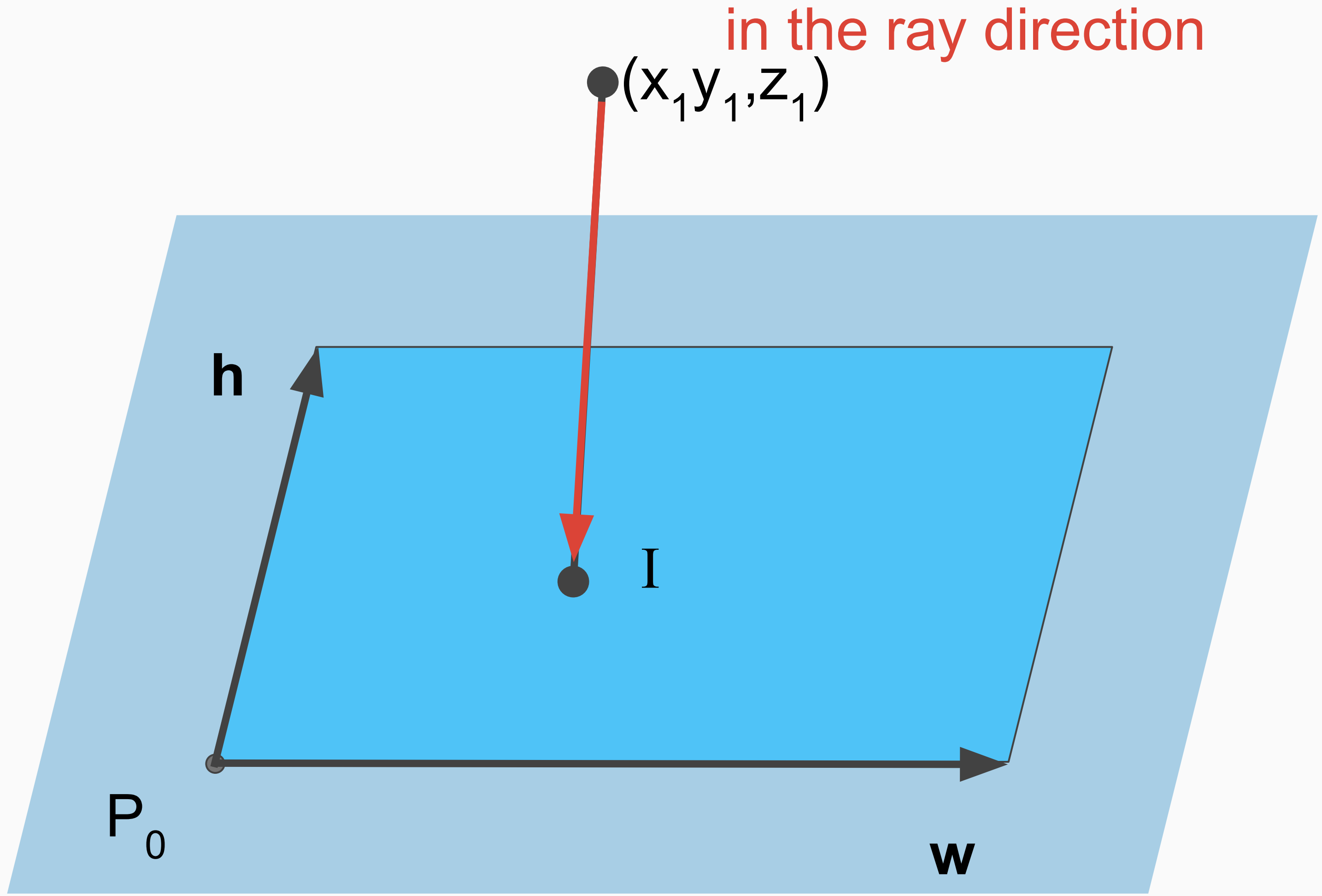
z ... $(-d - ax_1 - by_1 + (\alpha - c)z_1) / \alpha$

$\alpha = (au + bv + cw) / w$

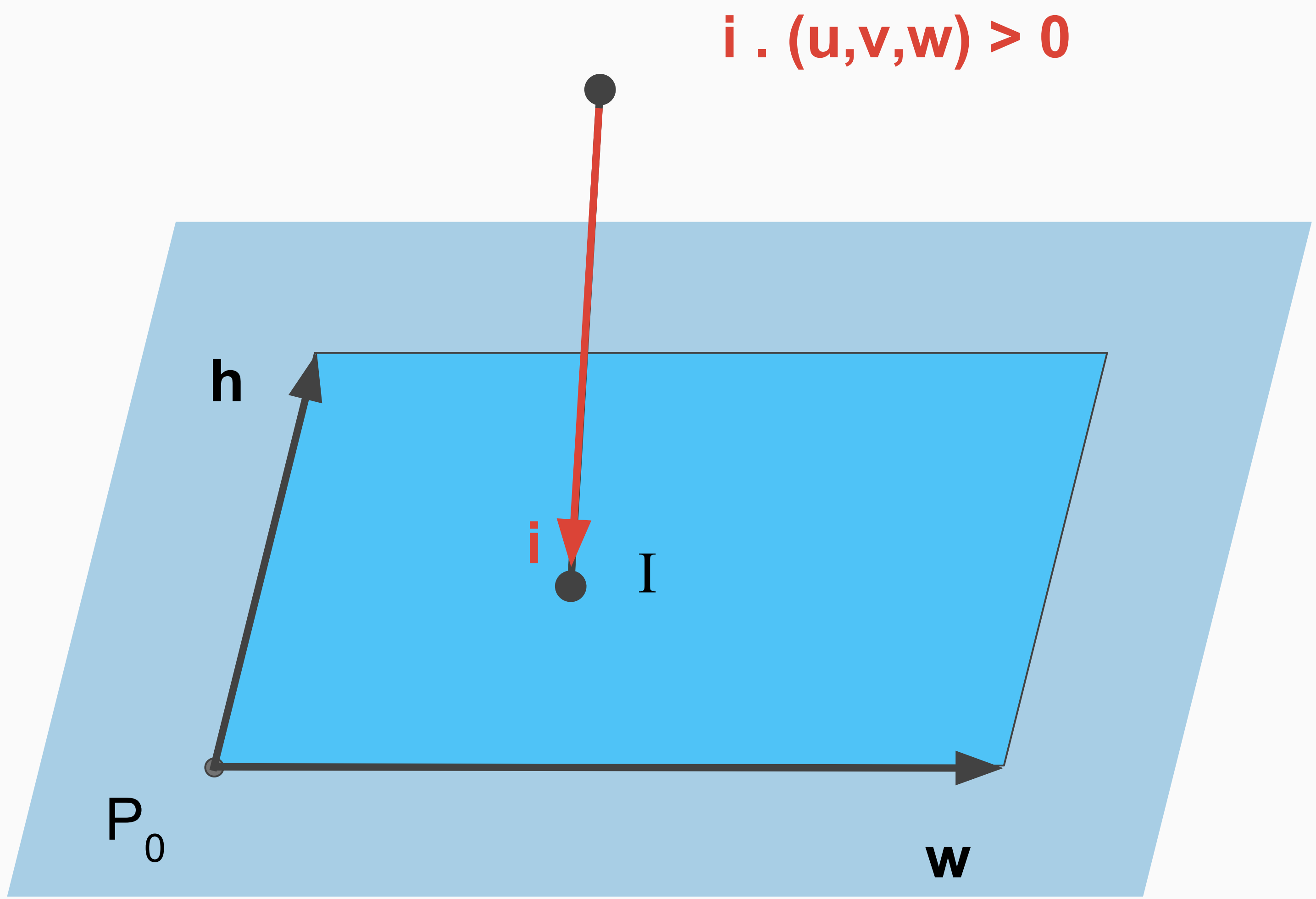
Ray - Rectangle intersection



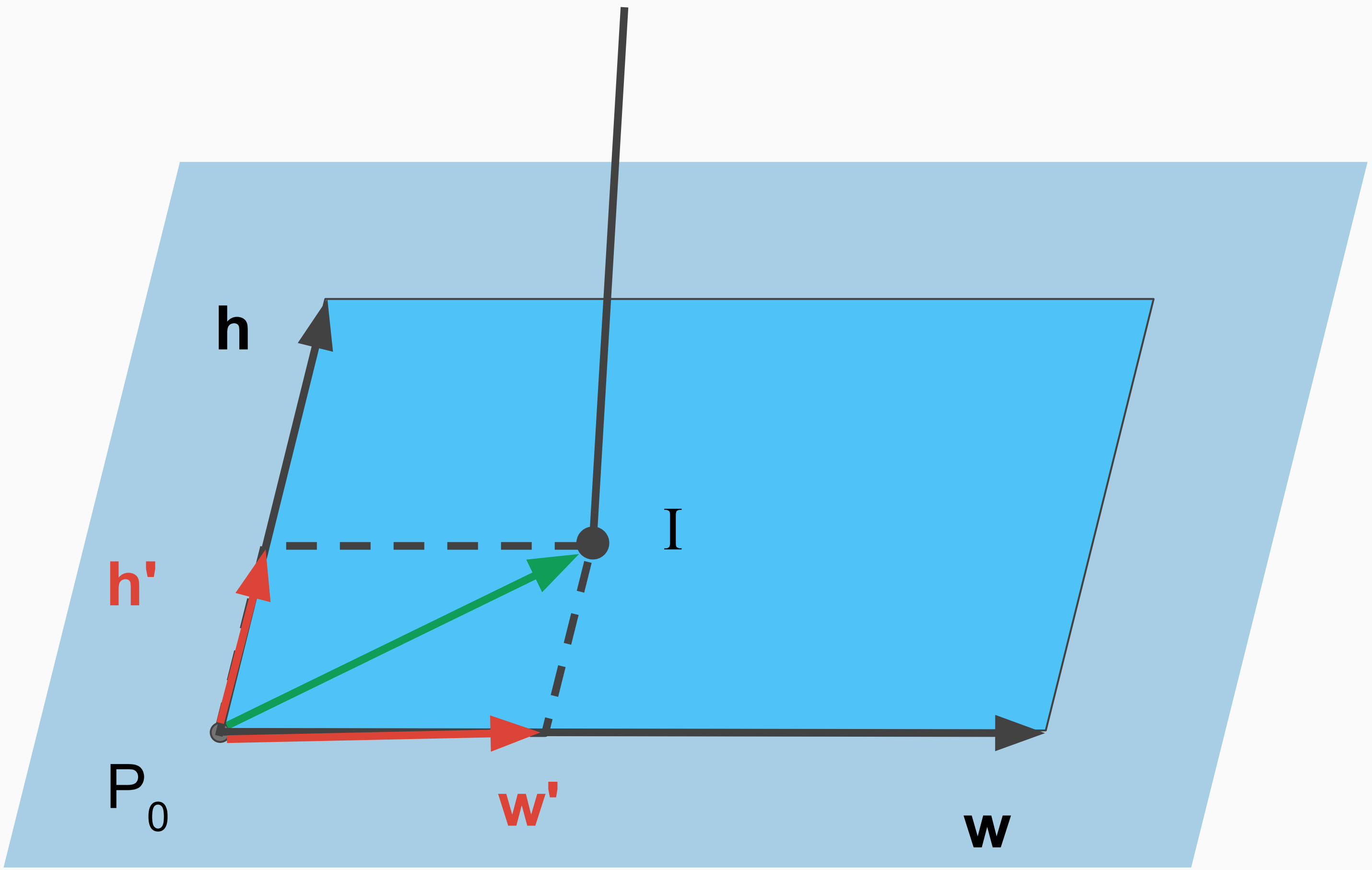
Ray - Rectangle intersection



Ray - Rectangle intersection

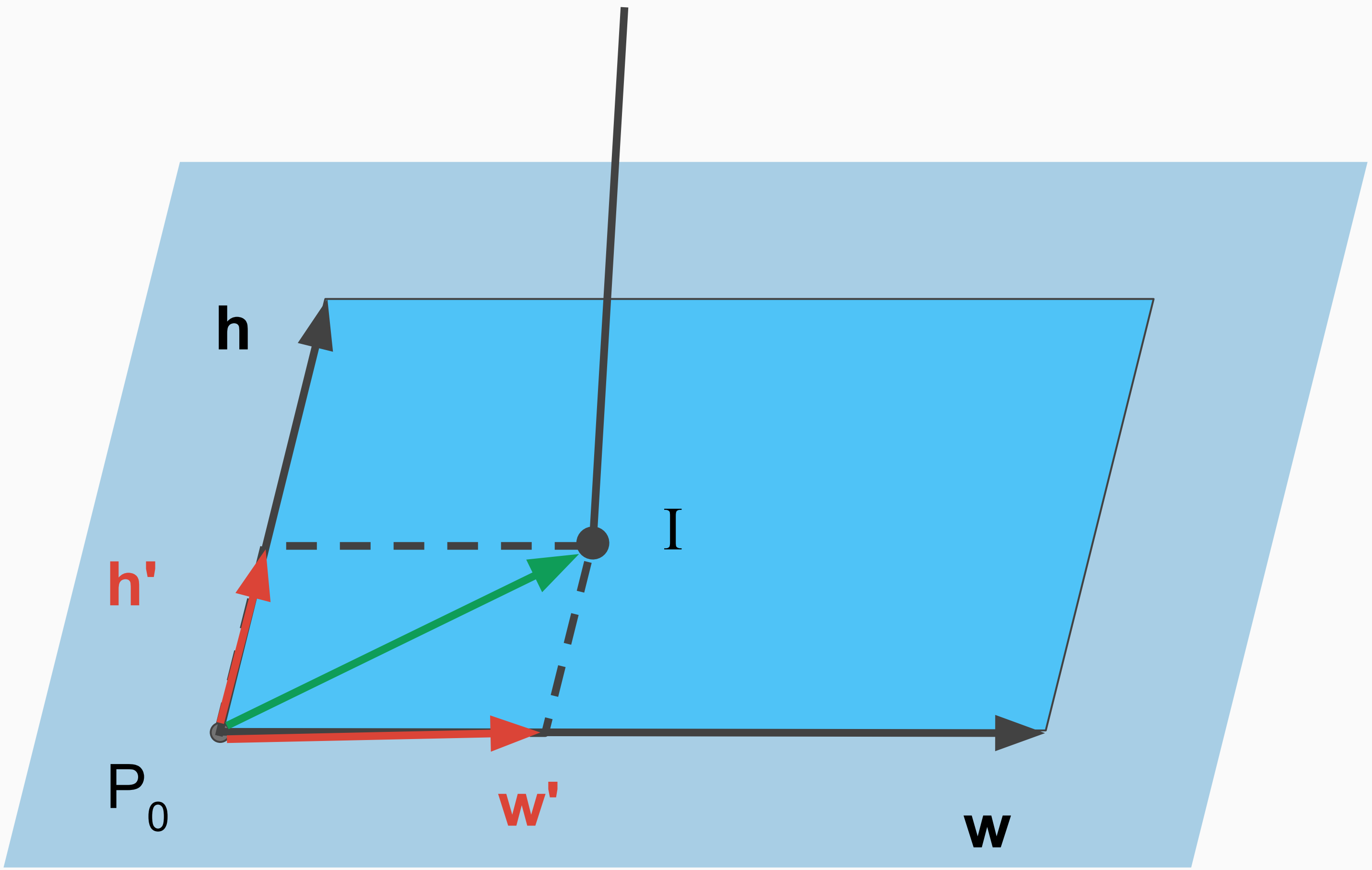


Ray - Rectangle intersection



$0 \leq ||h'|| \leq ||h||$ and $0 \leq ||w'|| \leq ||w||$

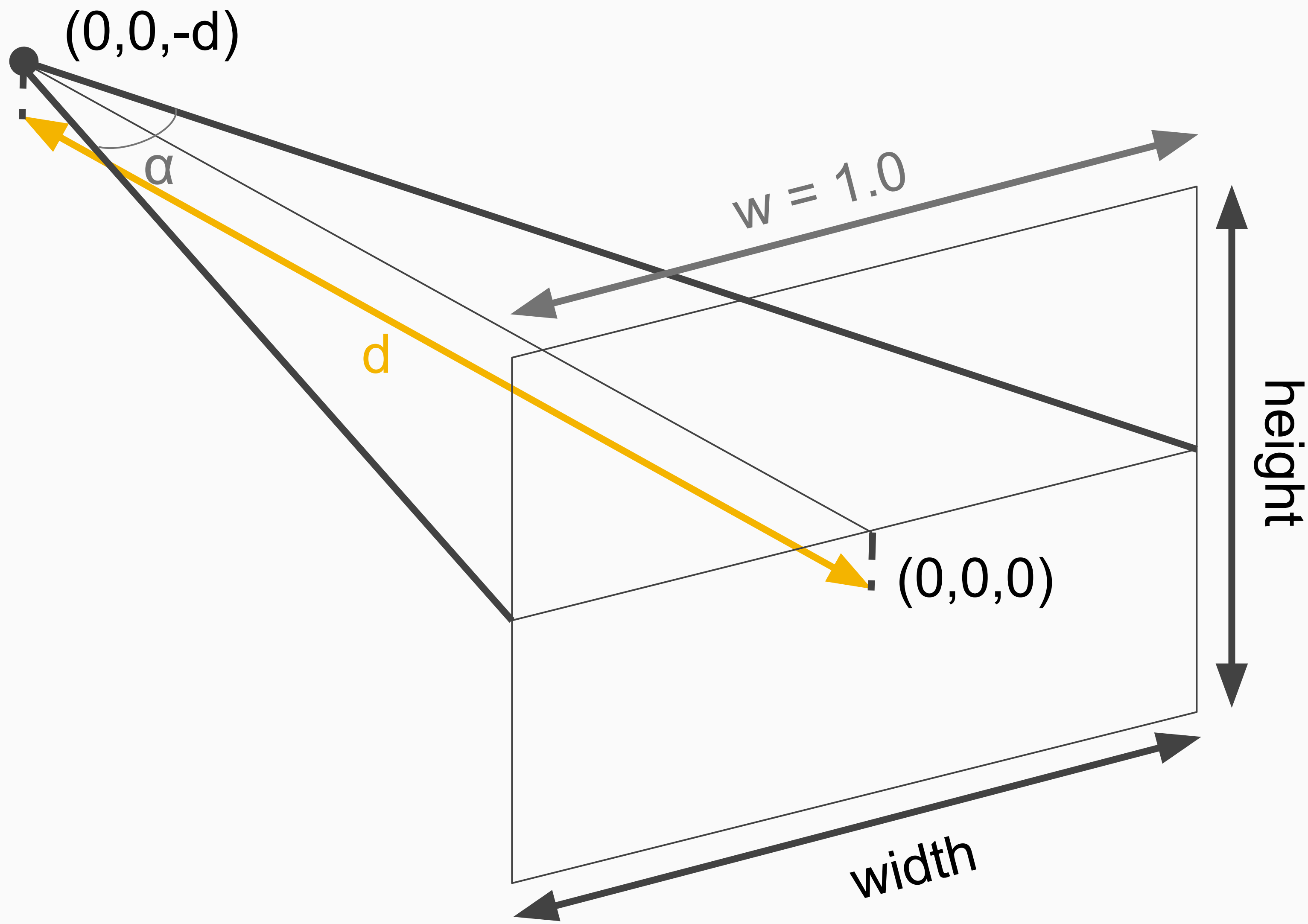
Ray - Rectangle intersection



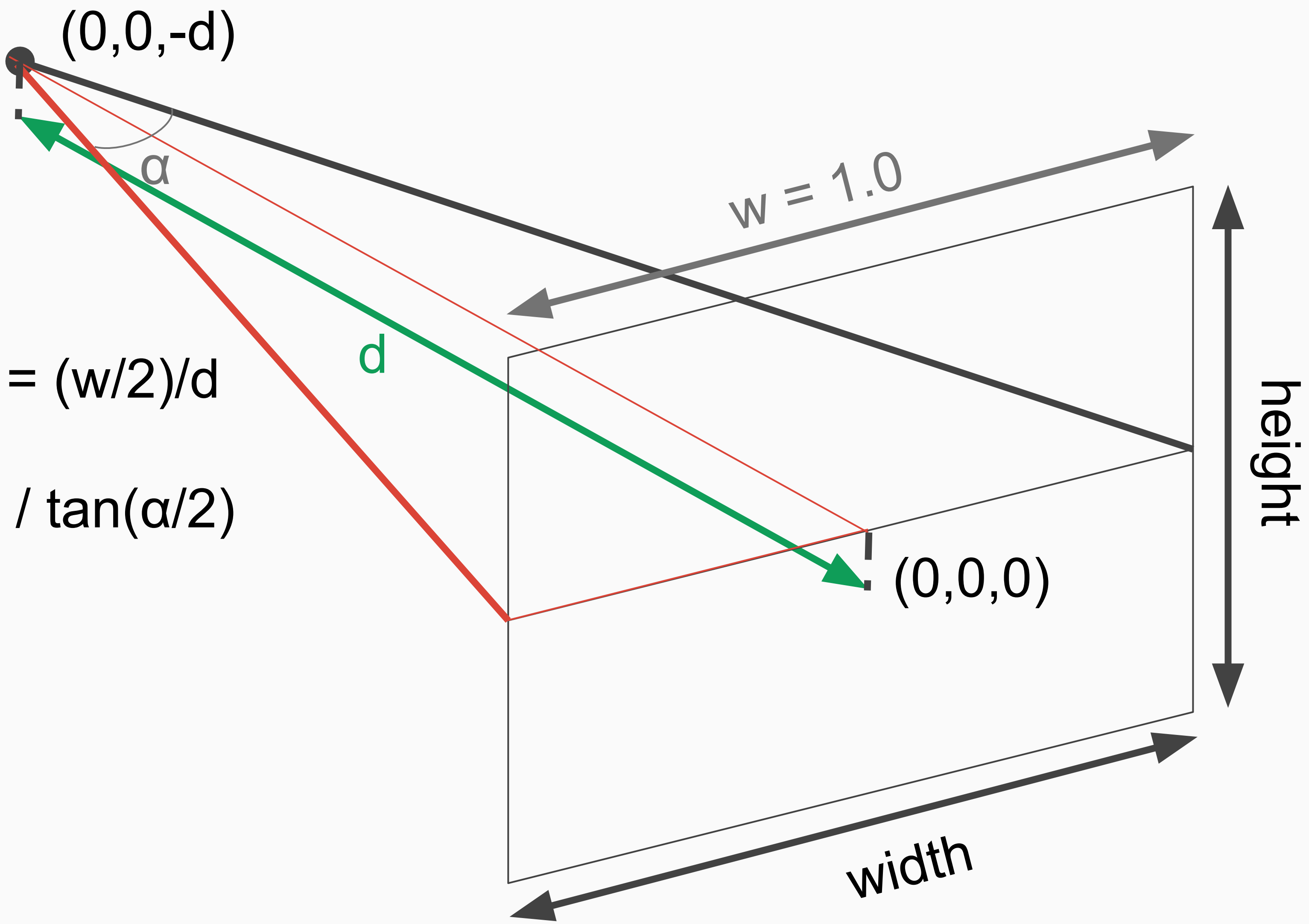
0 ≤ h' . h' ≤ h . h and 0 ≤ w' . w' ≤ w . w

Camera

Camera



Camera



$\tan(\alpha/2) = (w/2)/d$

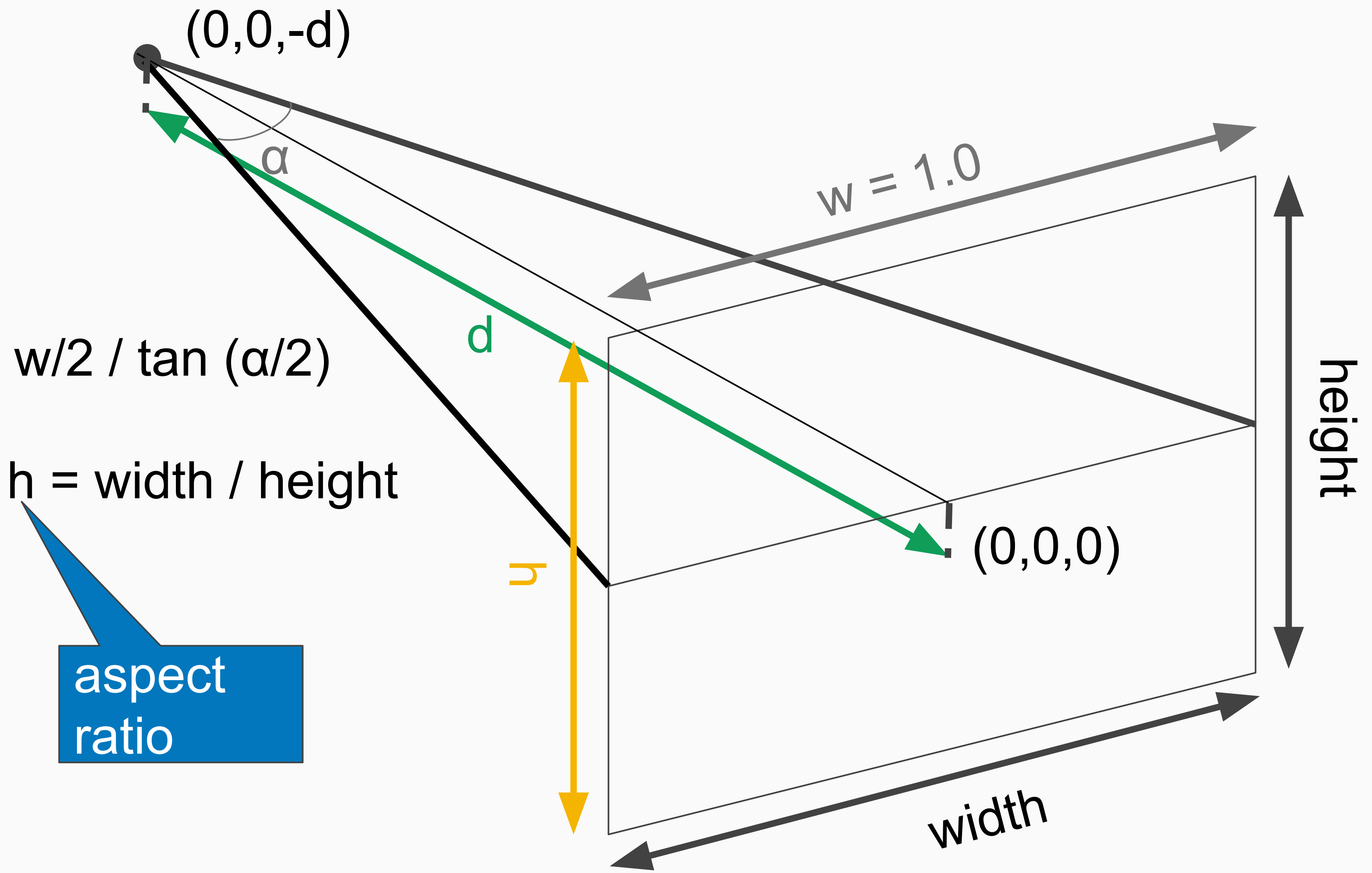
⇔

$d = (w/2) / \tan(\alpha/2)$

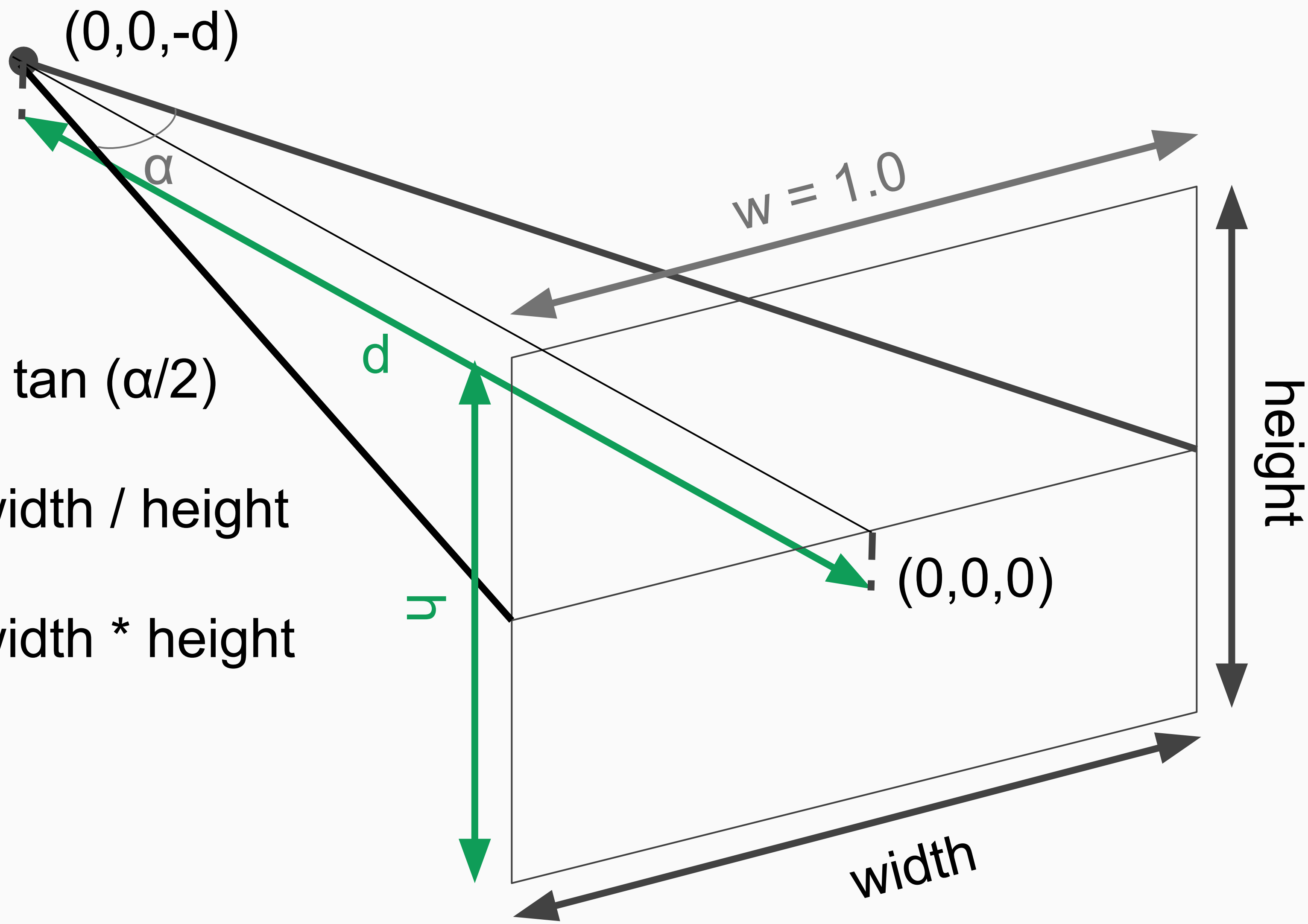
Camera

$d = w/2 / \tan(\alpha/2)$

$w / h = \text{width} / \text{height}$



Camera



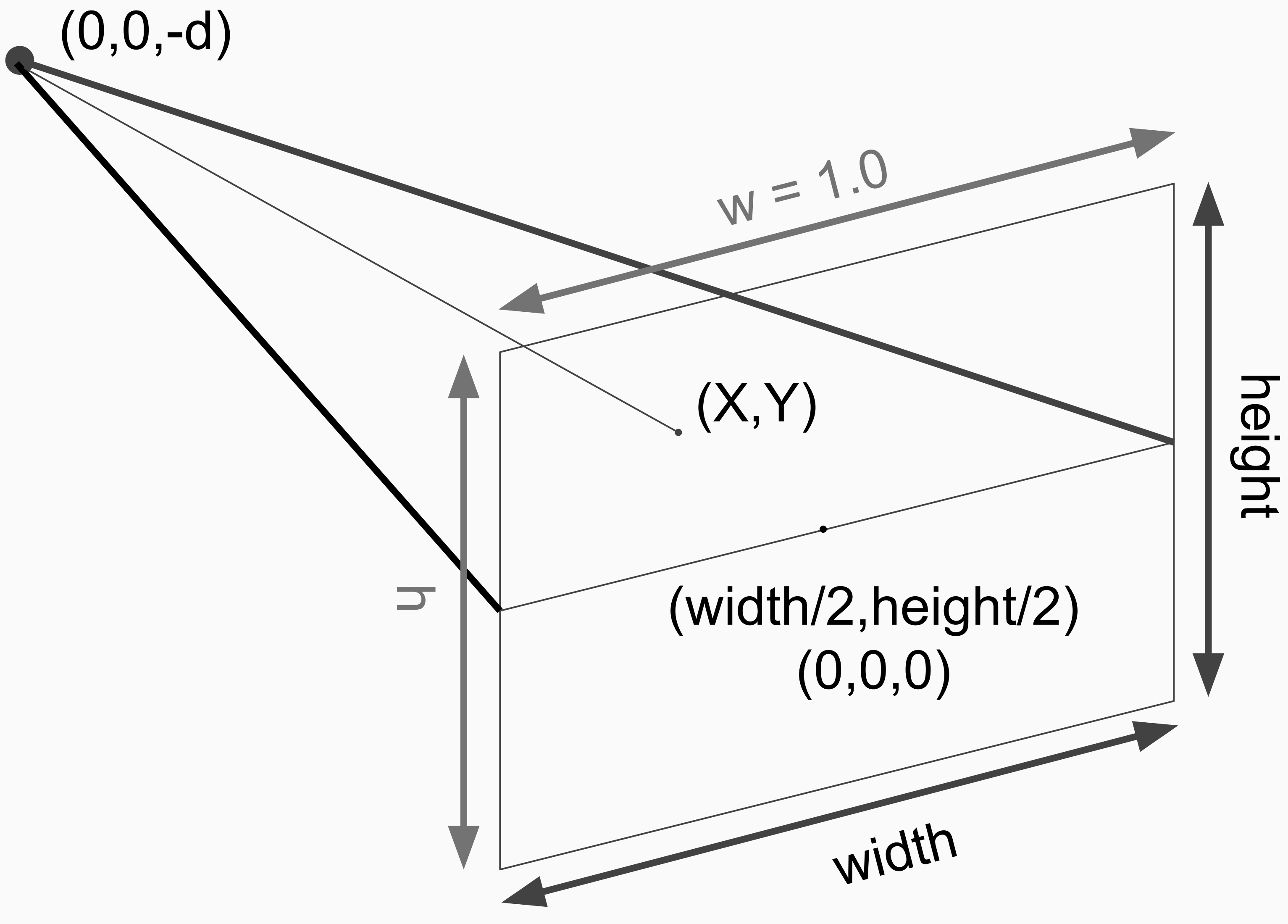
d = w/2 / tan (α/2)

w / h = width / height

⇔

h = w / width * height

Camera

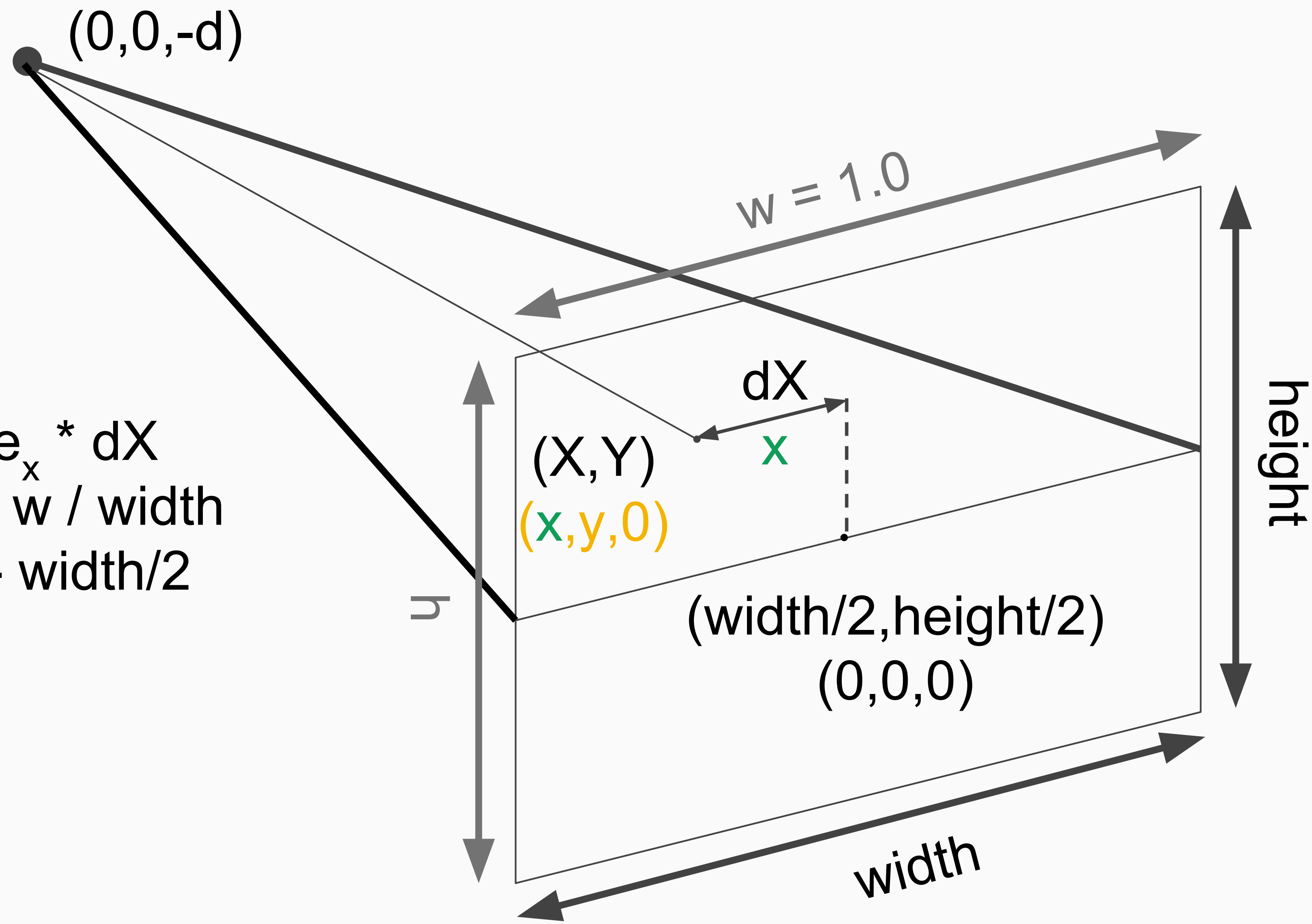


Camera

$x = scale_x * dX$

$scale_x = w / width$

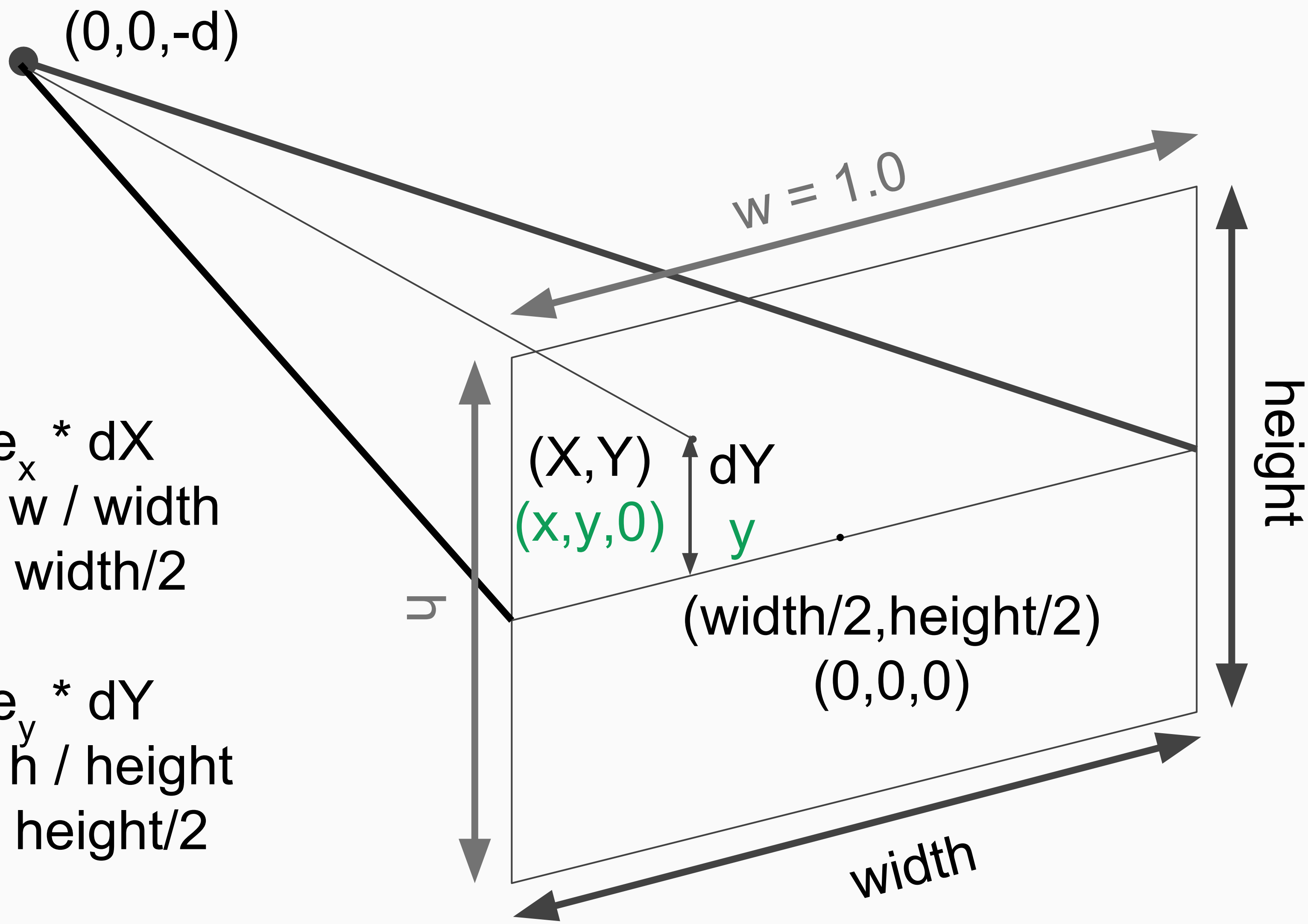
$dX = X - width/2$



Camera

$x = scale_x * dX$
$scale_x = w / width$
$dX = X - width/2$

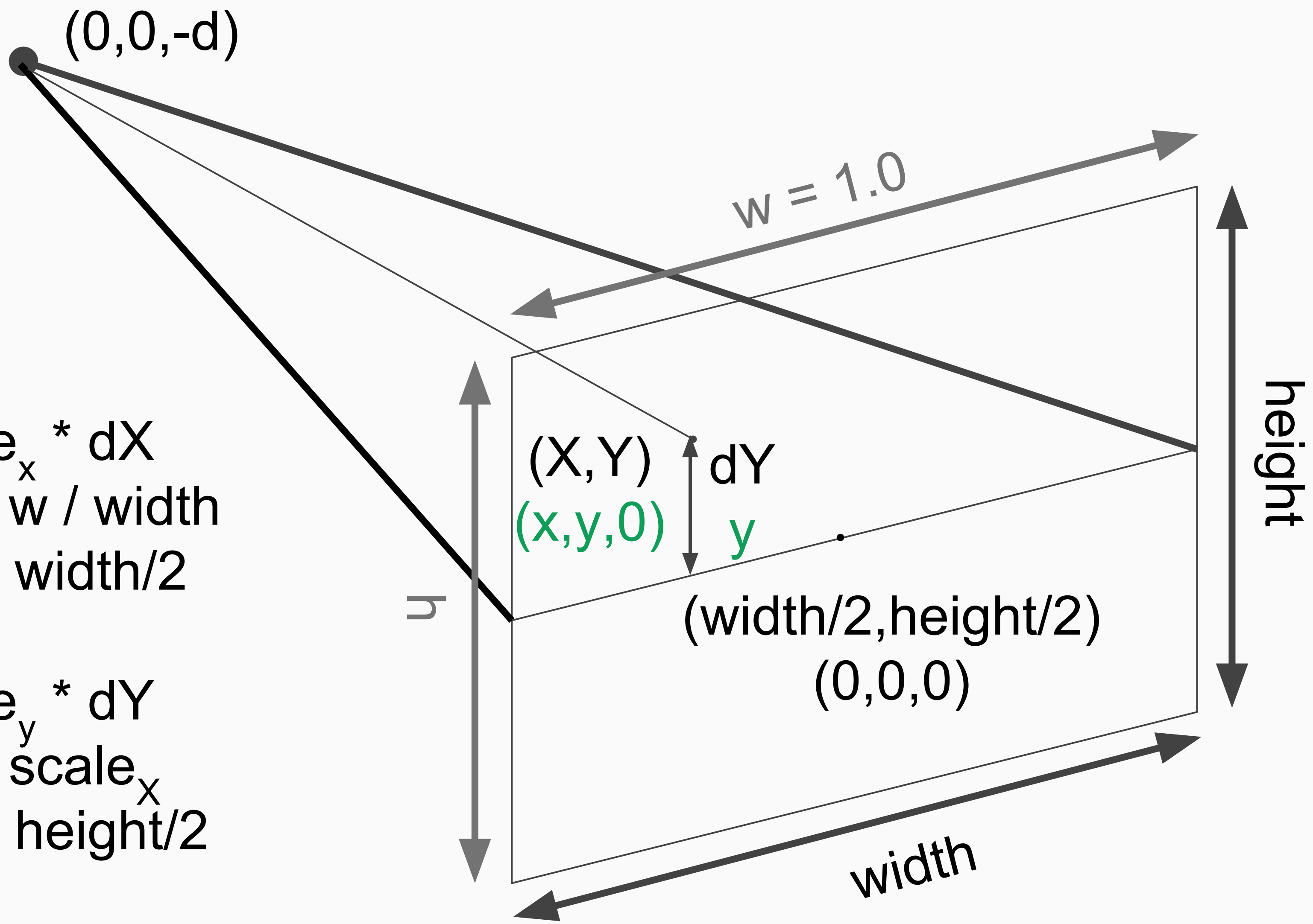
$y = scale_y * dY$
$scale_y = h / height$
$dY = Y - height/2$



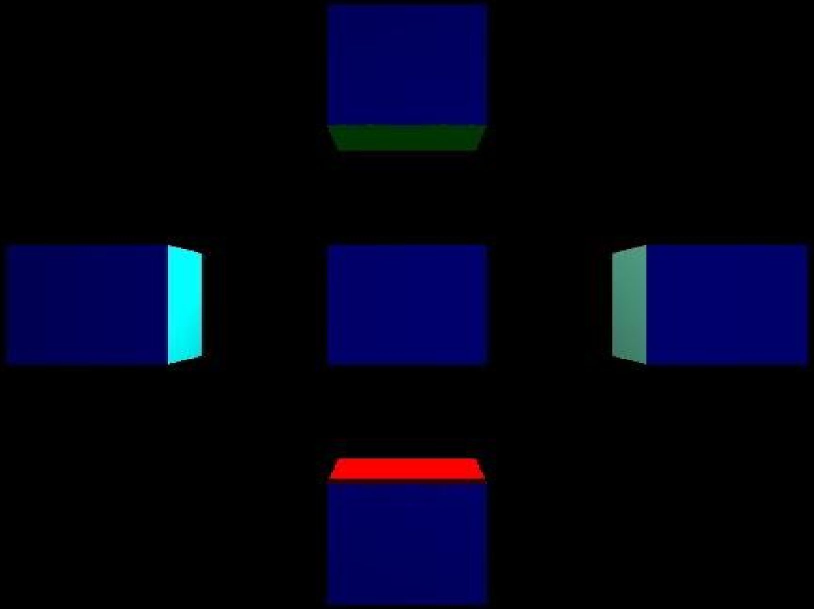
Camera

$x = scale_x * dX$
$scale_x = w / width$
$dX = X - width/2$

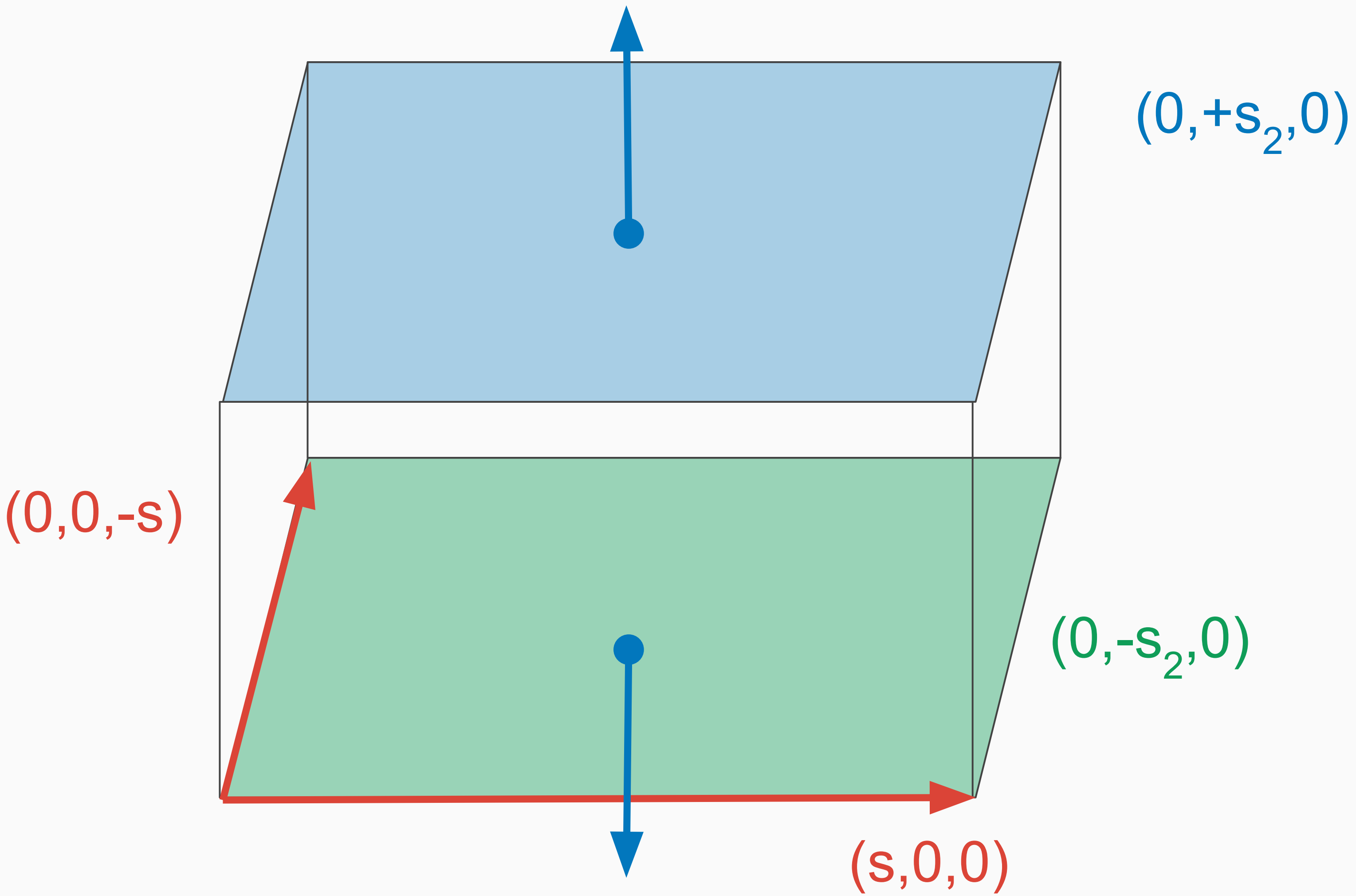
$y = scale_y * dY$
$scale_y = scale_x$
$dY = Y - height/2$



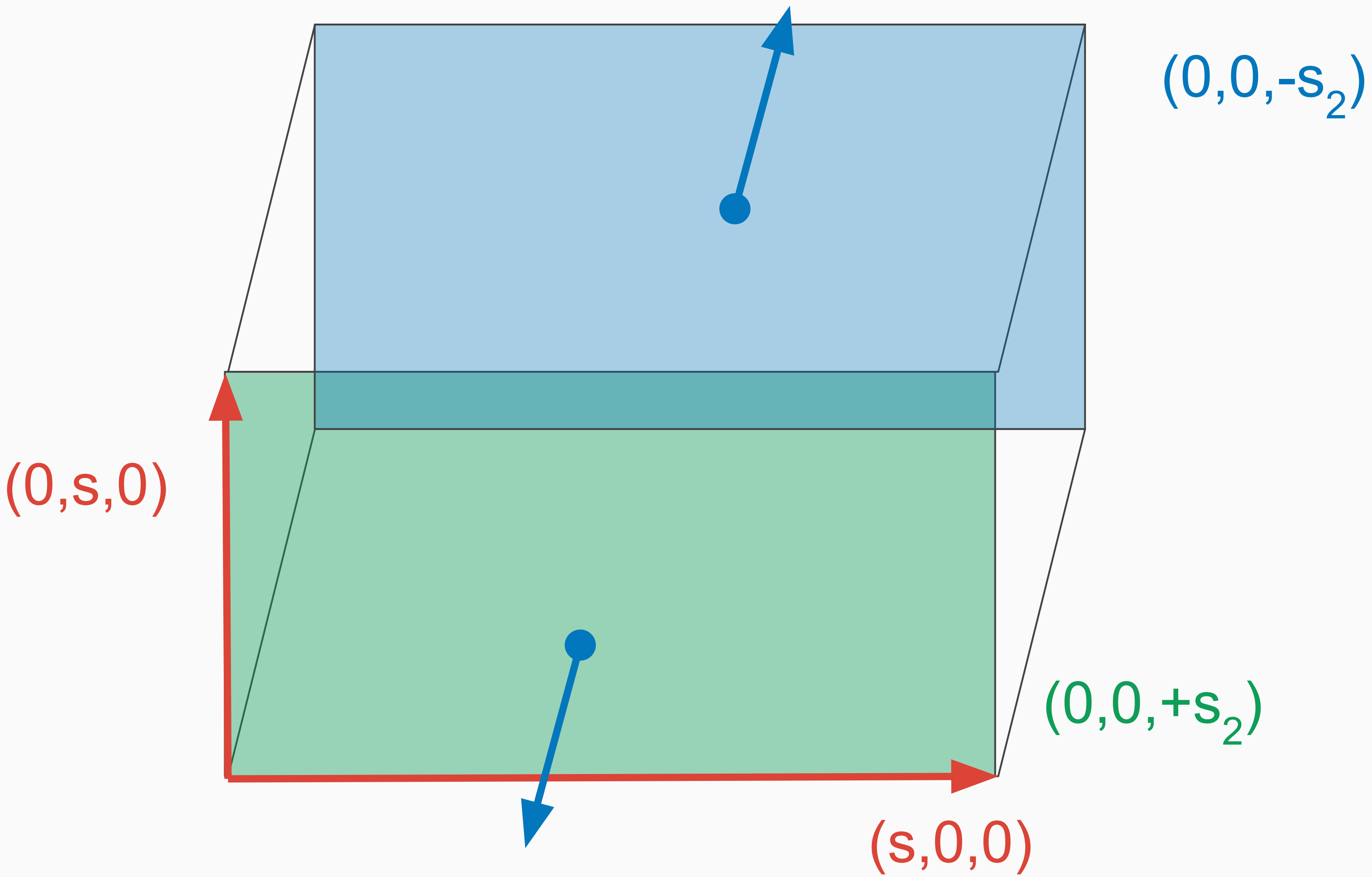
Cube



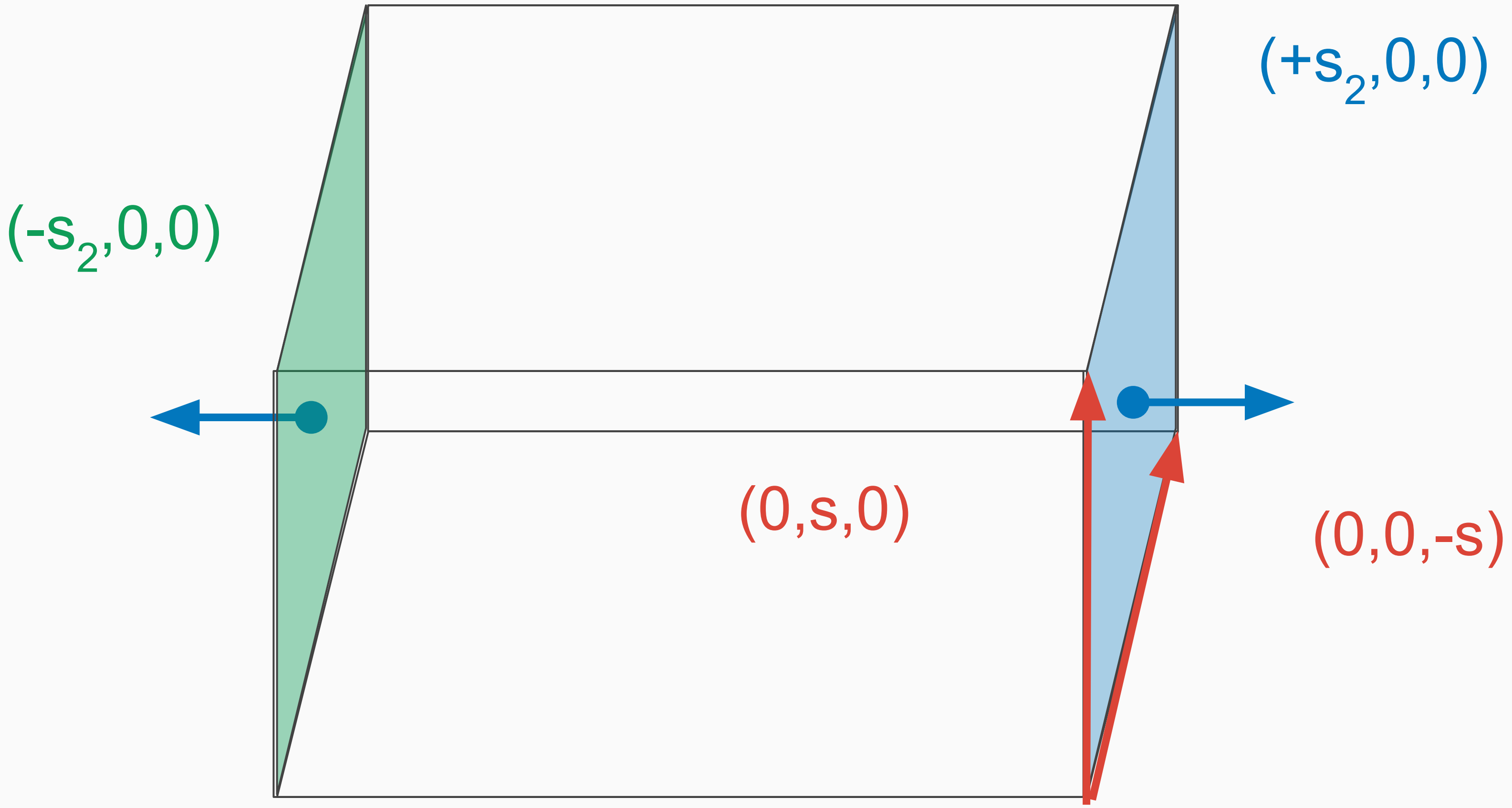
Cube

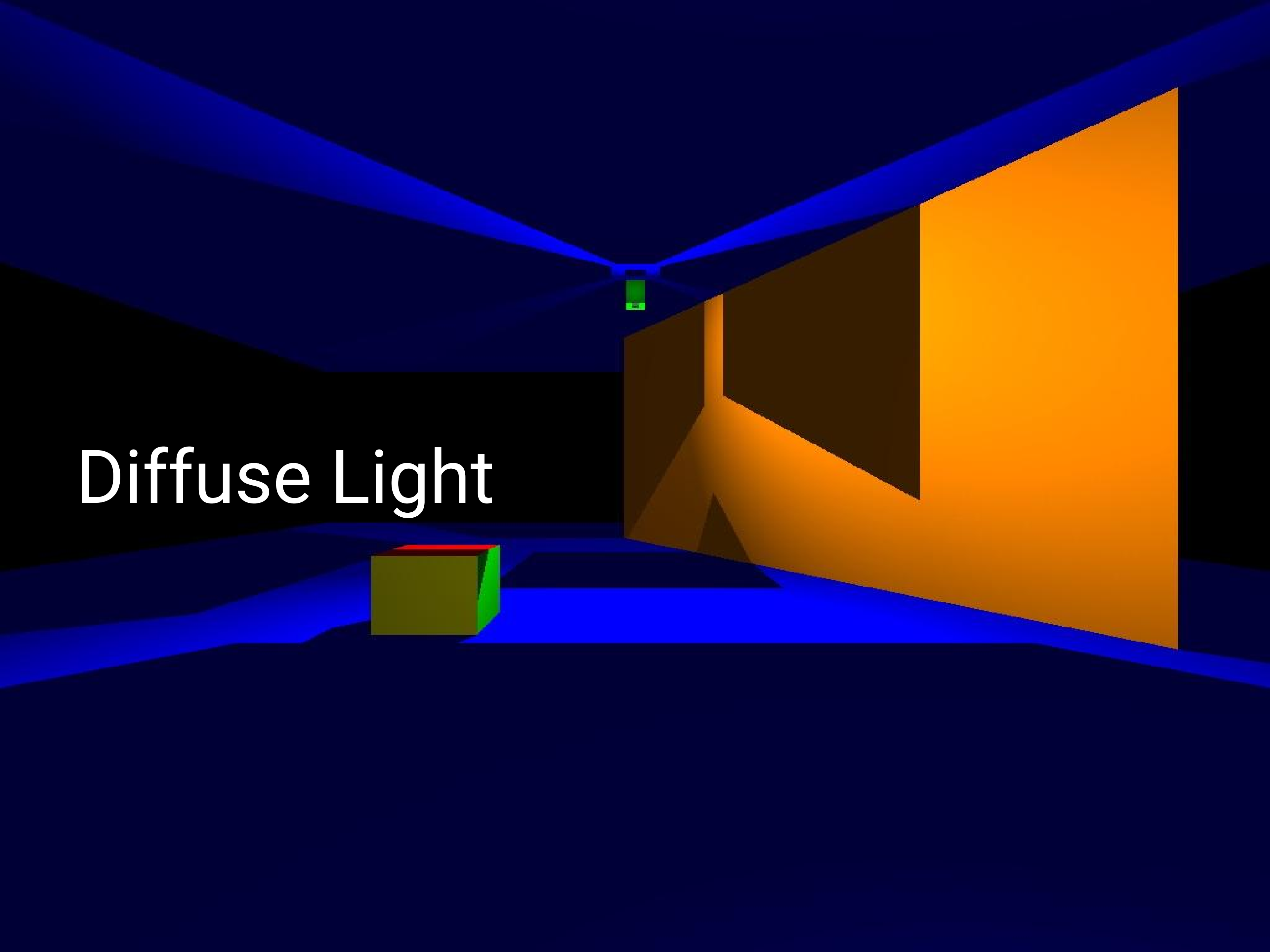


Cube

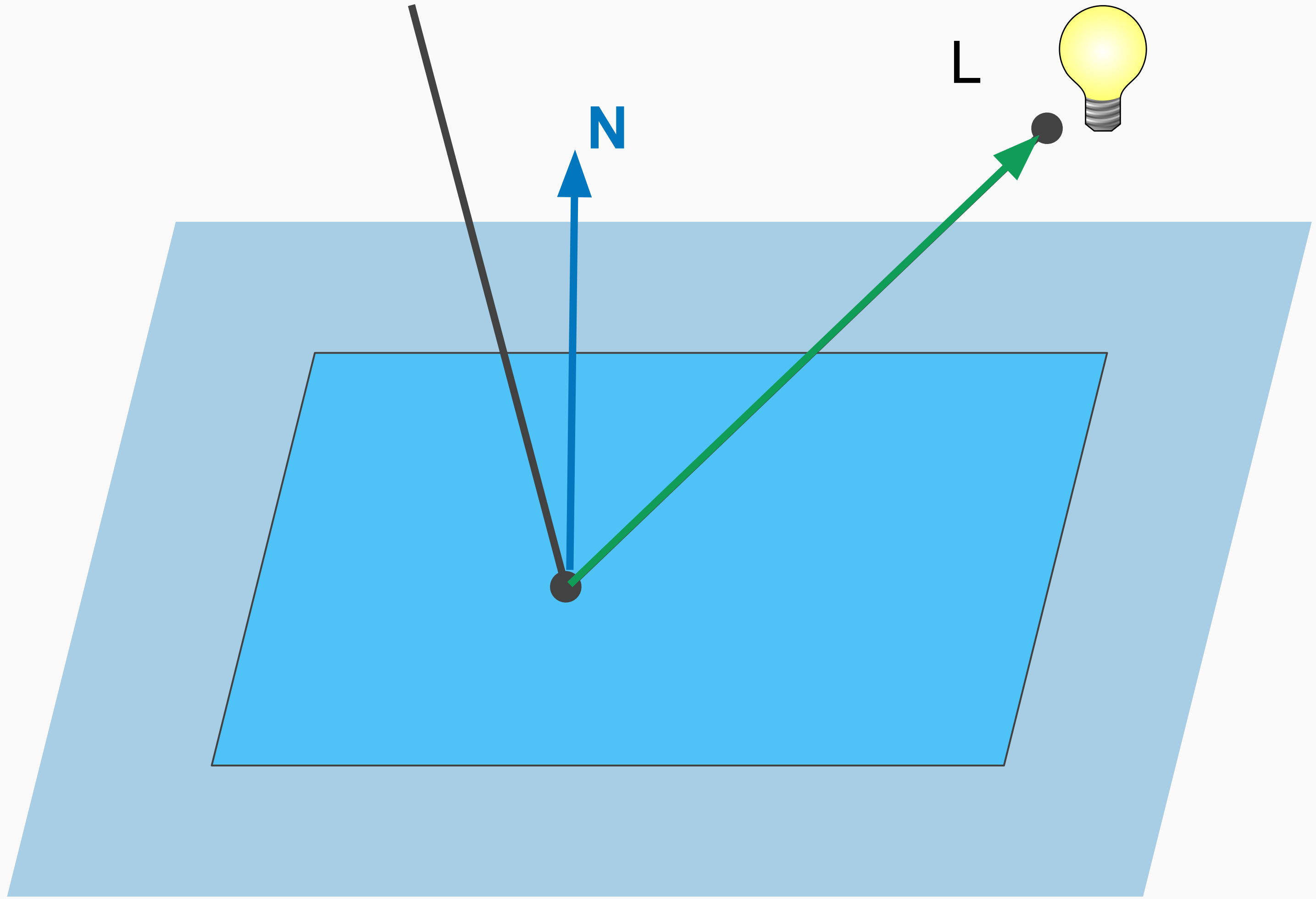


Cube

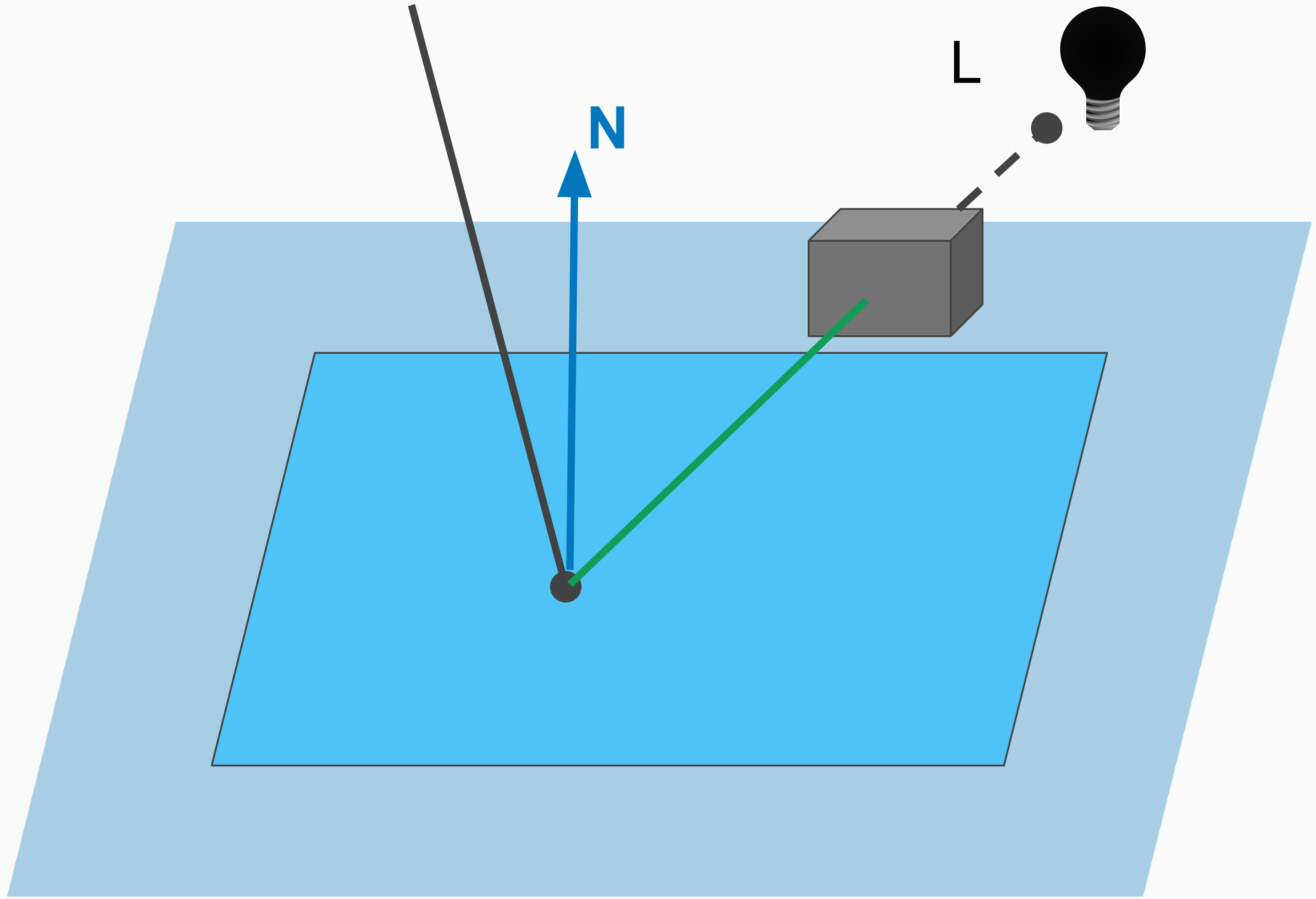



Diffuse Light

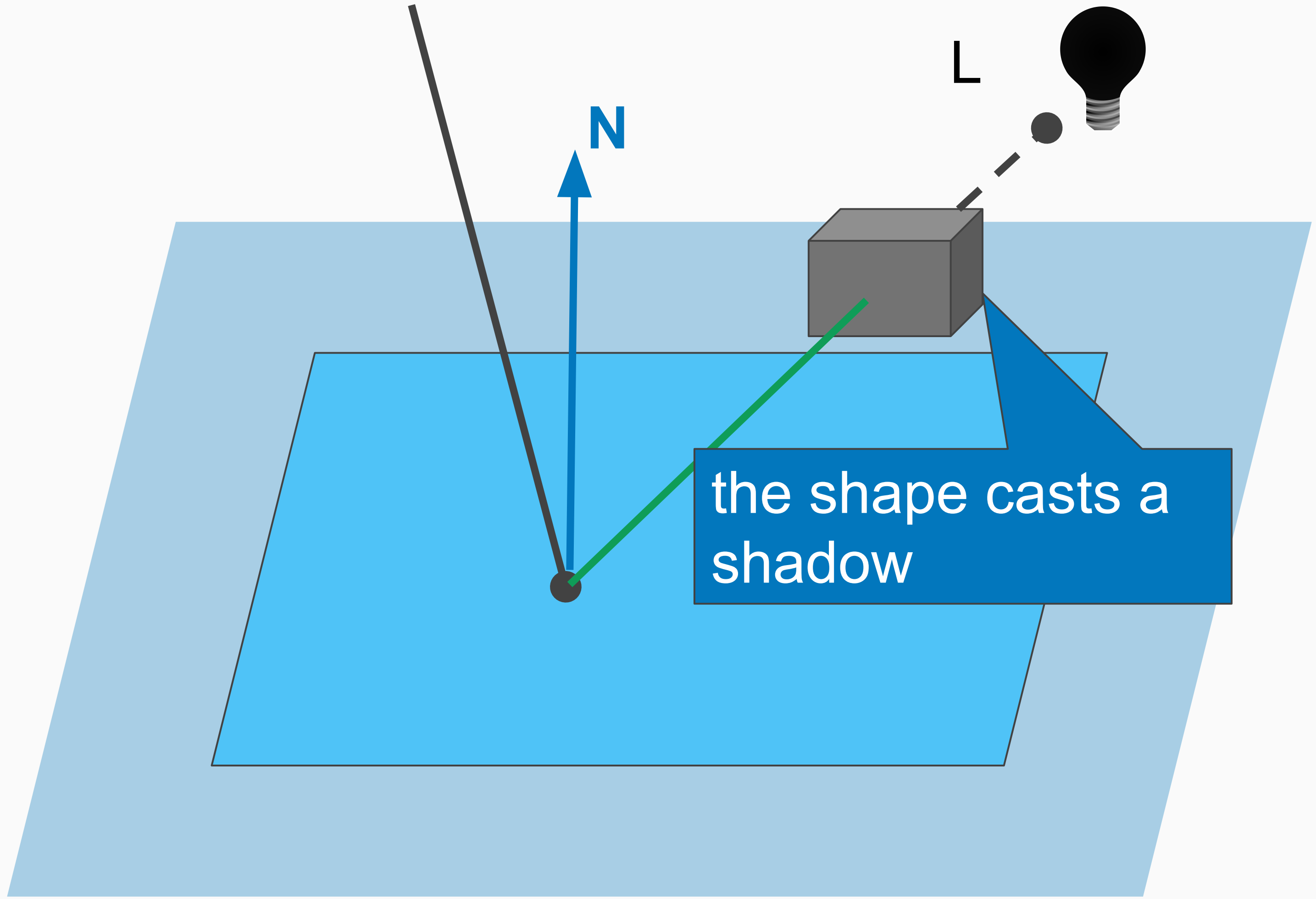
Diffuse Light



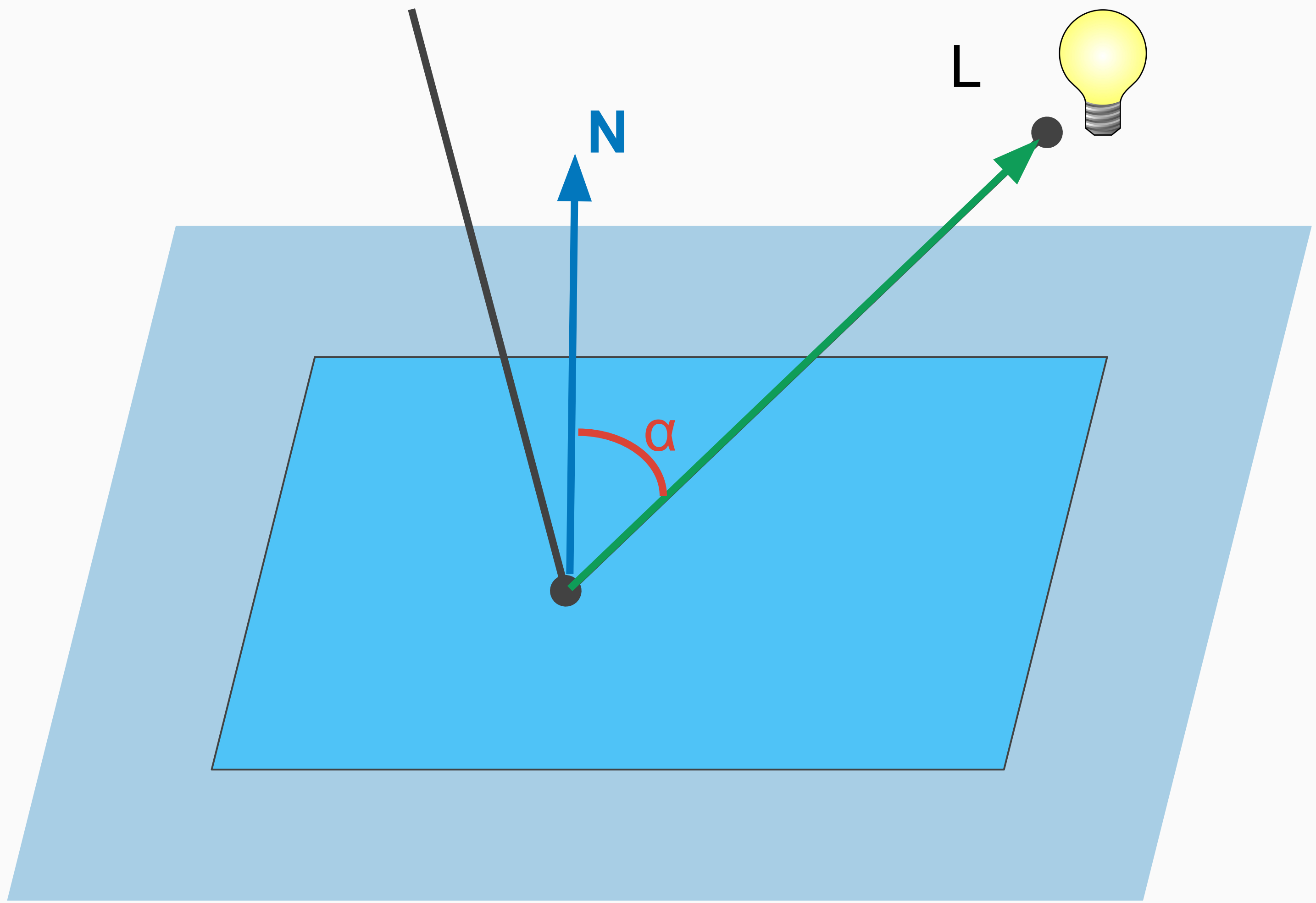
Diffuse Light



Diffuse Light

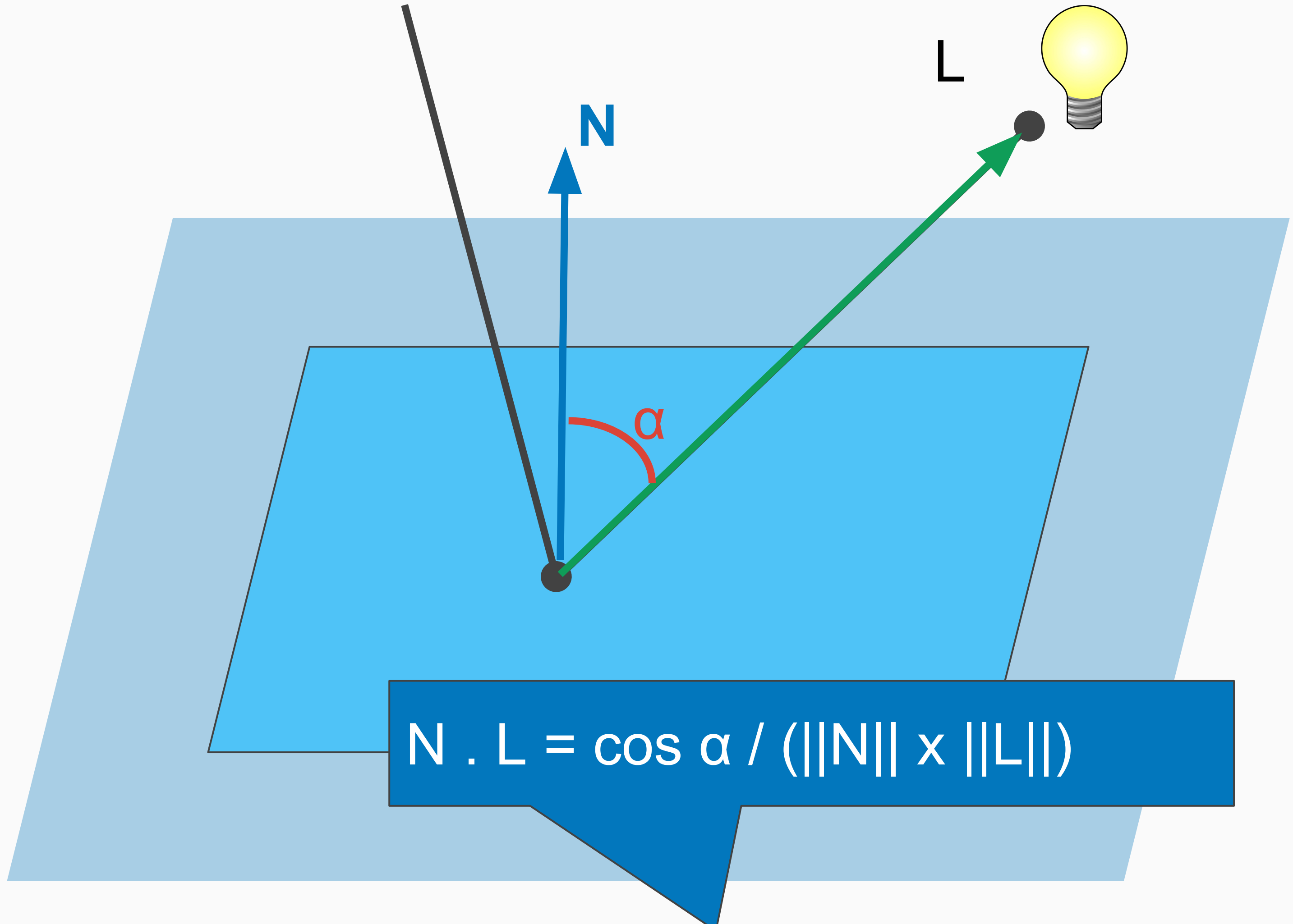


Diffuse Light



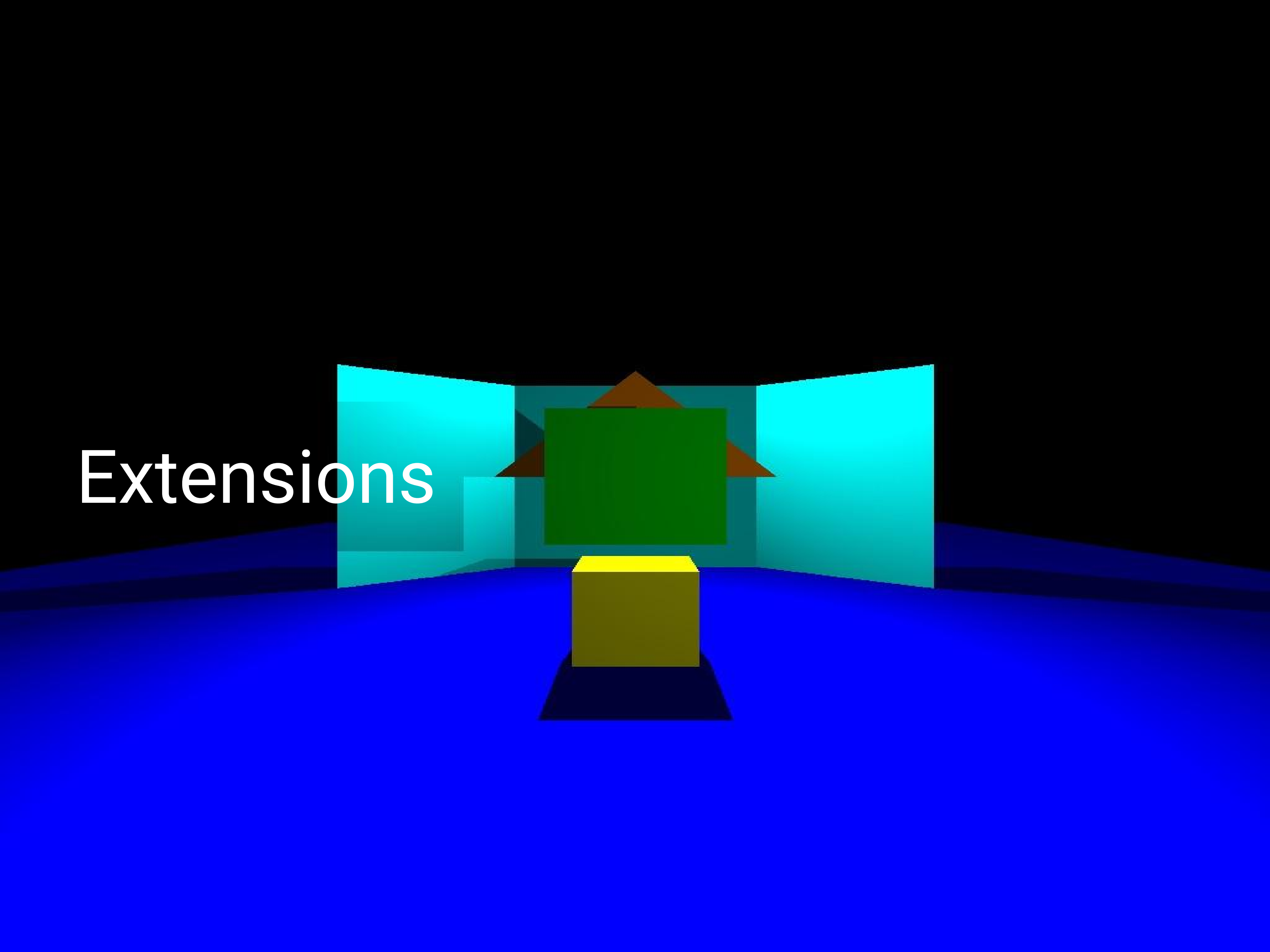
$$\text{Color} = \text{Color}_{shape} \times \cos\alpha \times I_{light}$$

Diffuse Light

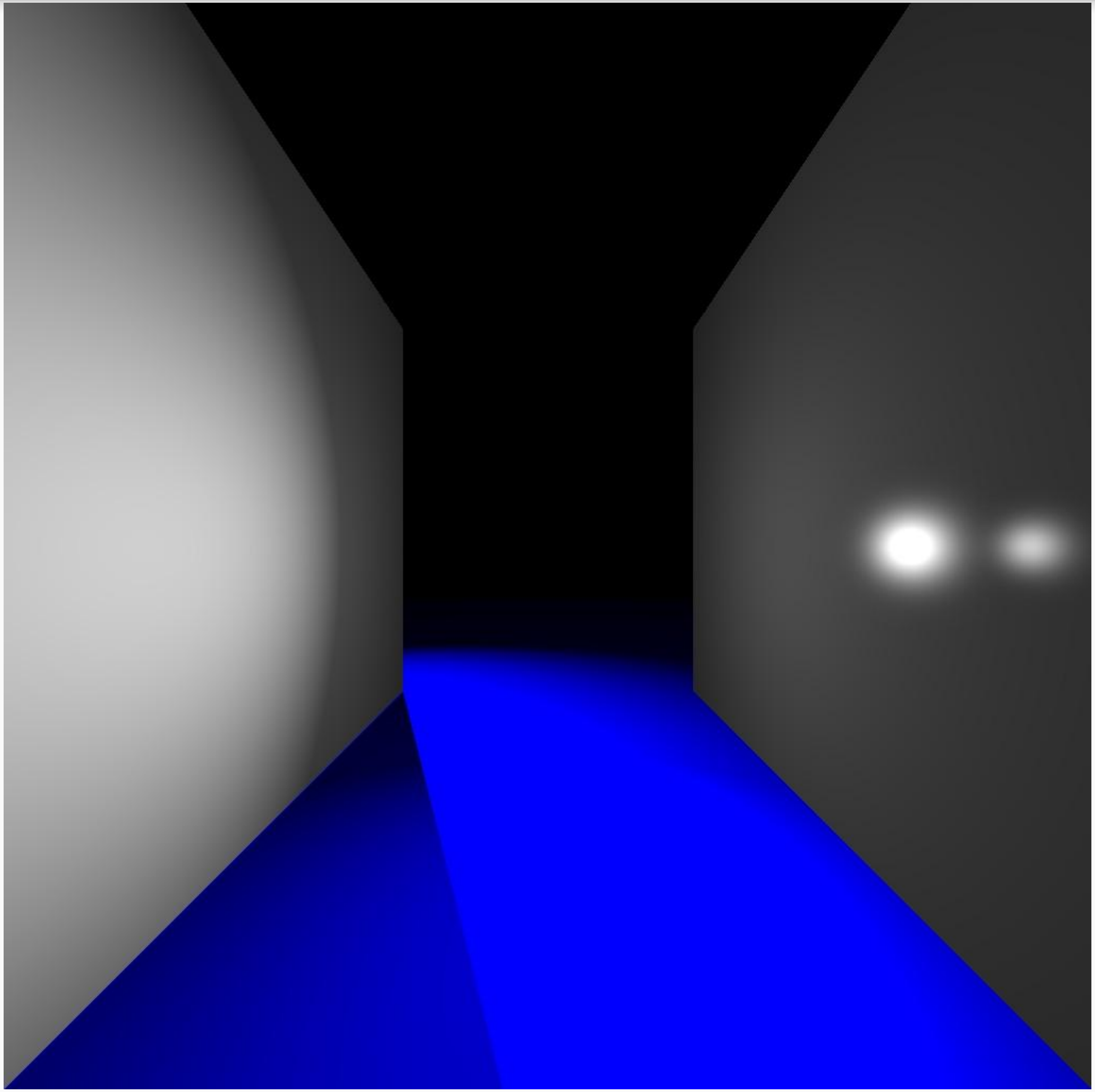


$$Color = Color_{shape} \times (N \cdot L) \times I_{light}$$

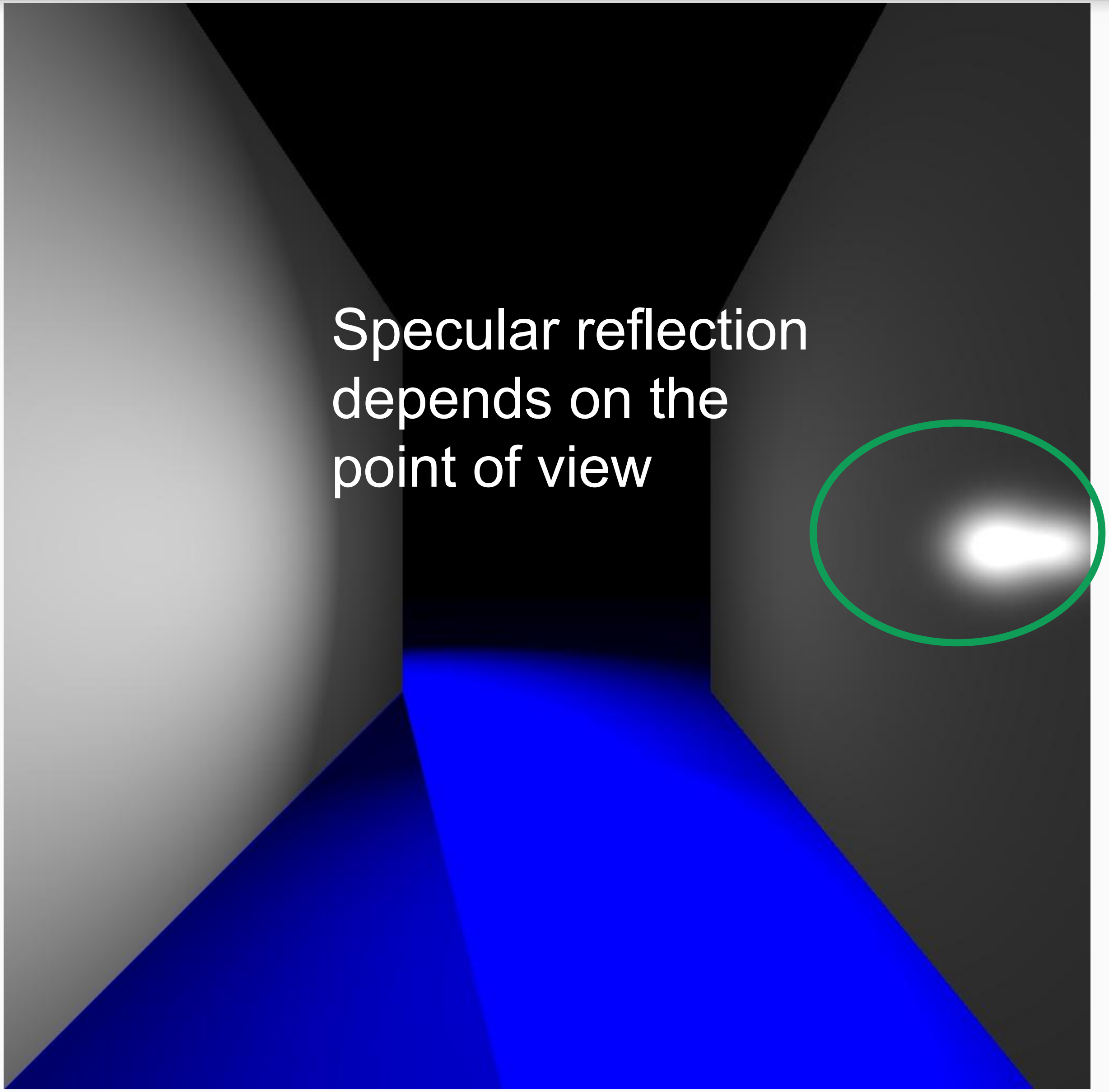
Extensions



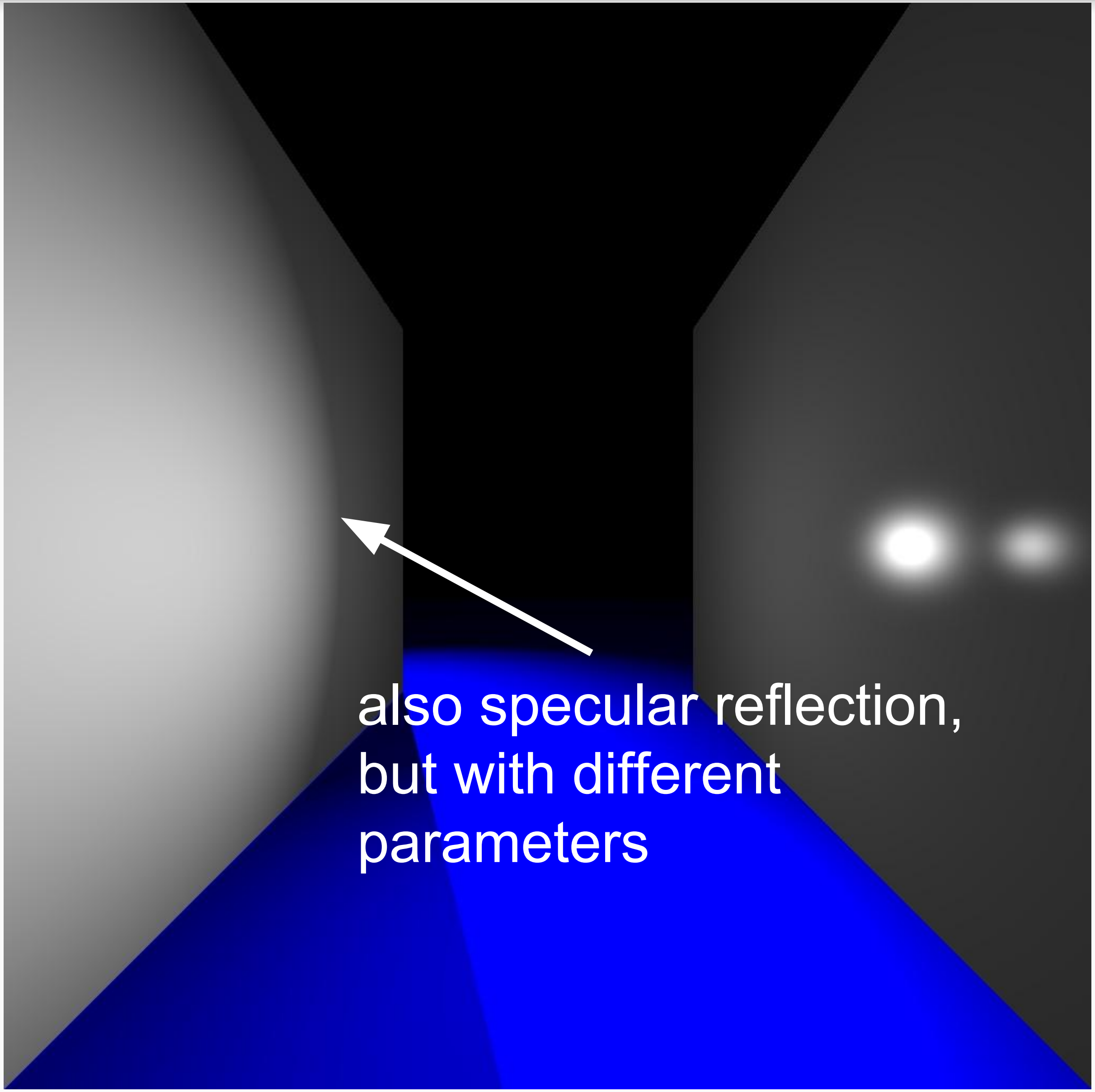
Specular lights



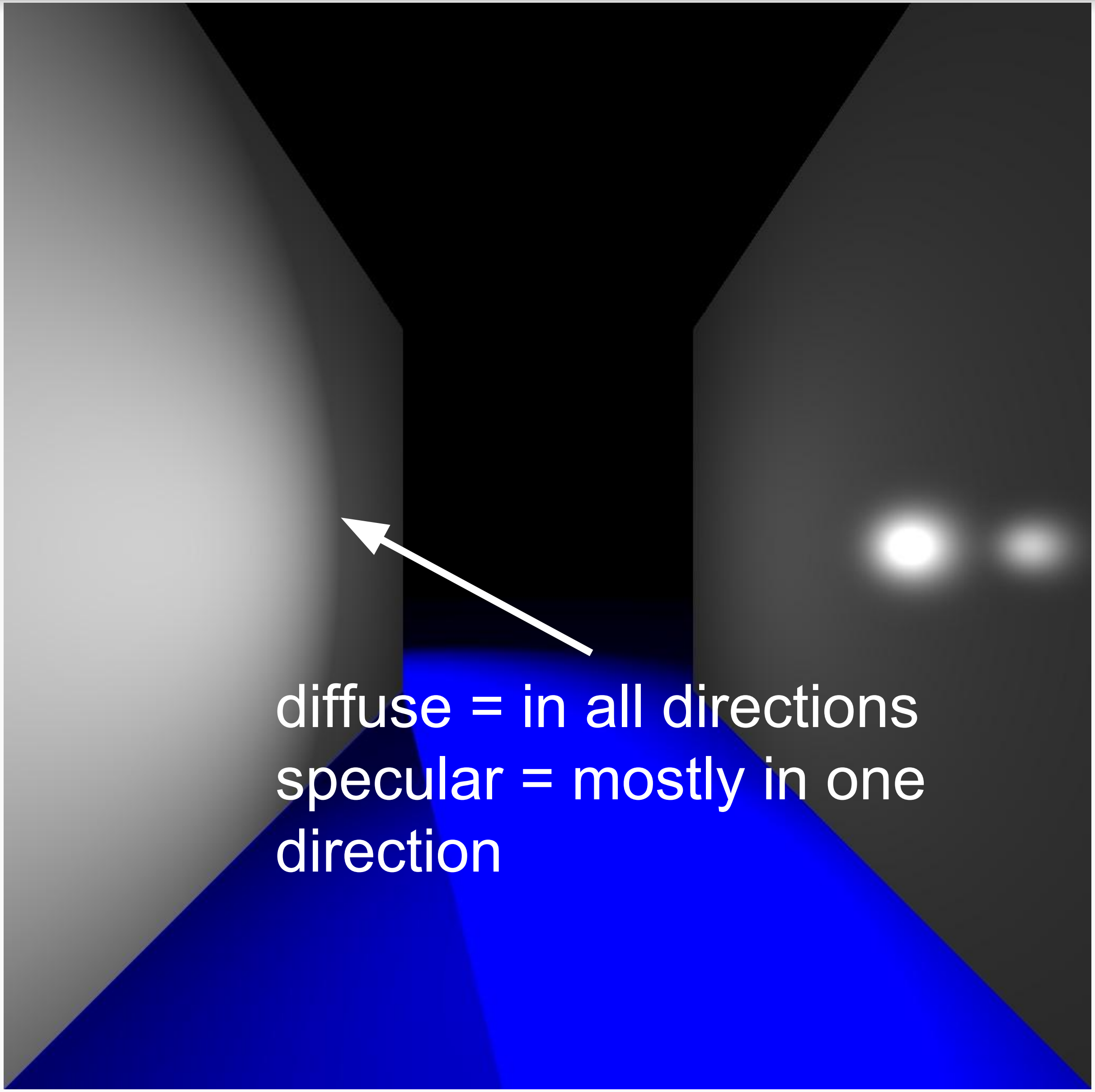
Specular lights



Specular lights

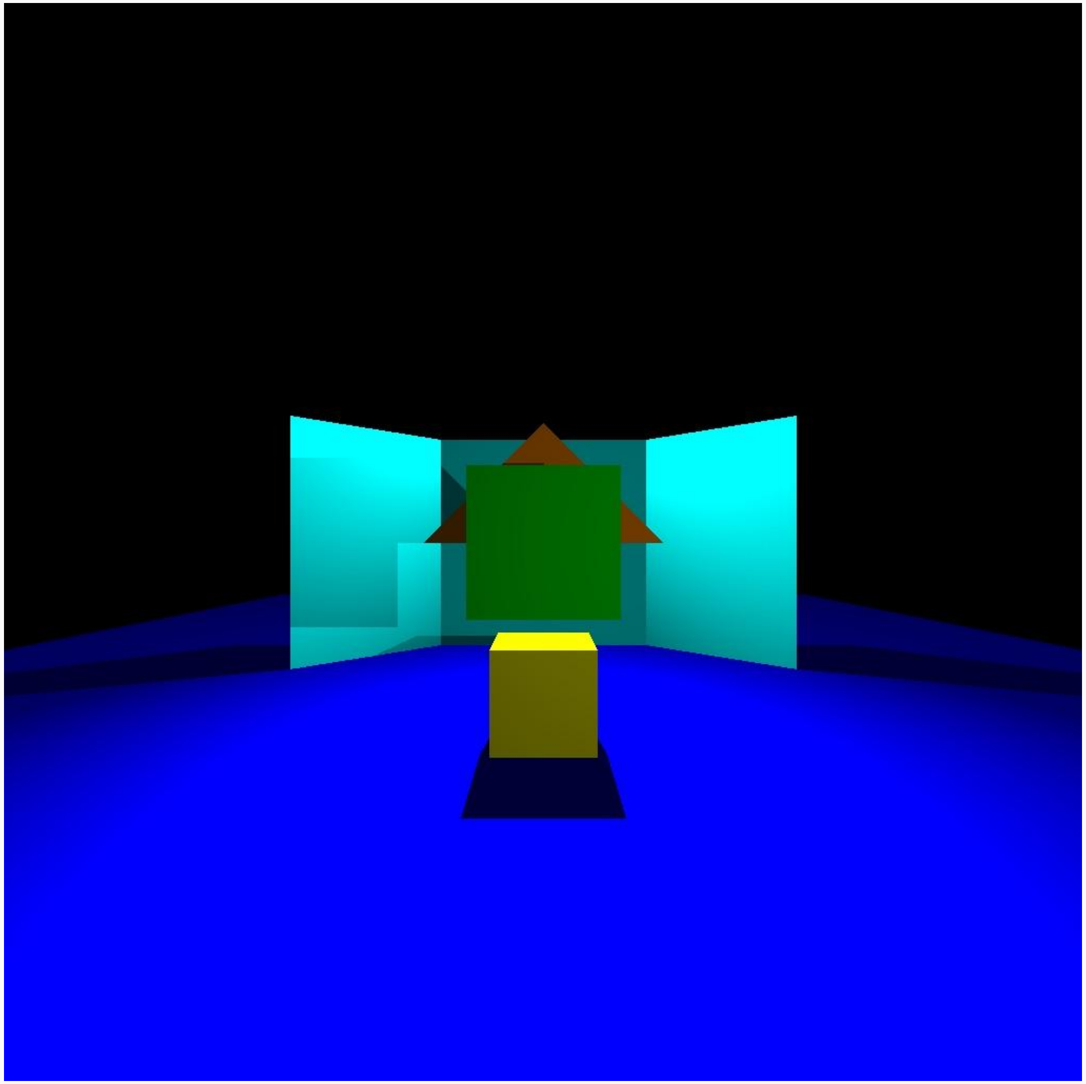


Specular lights



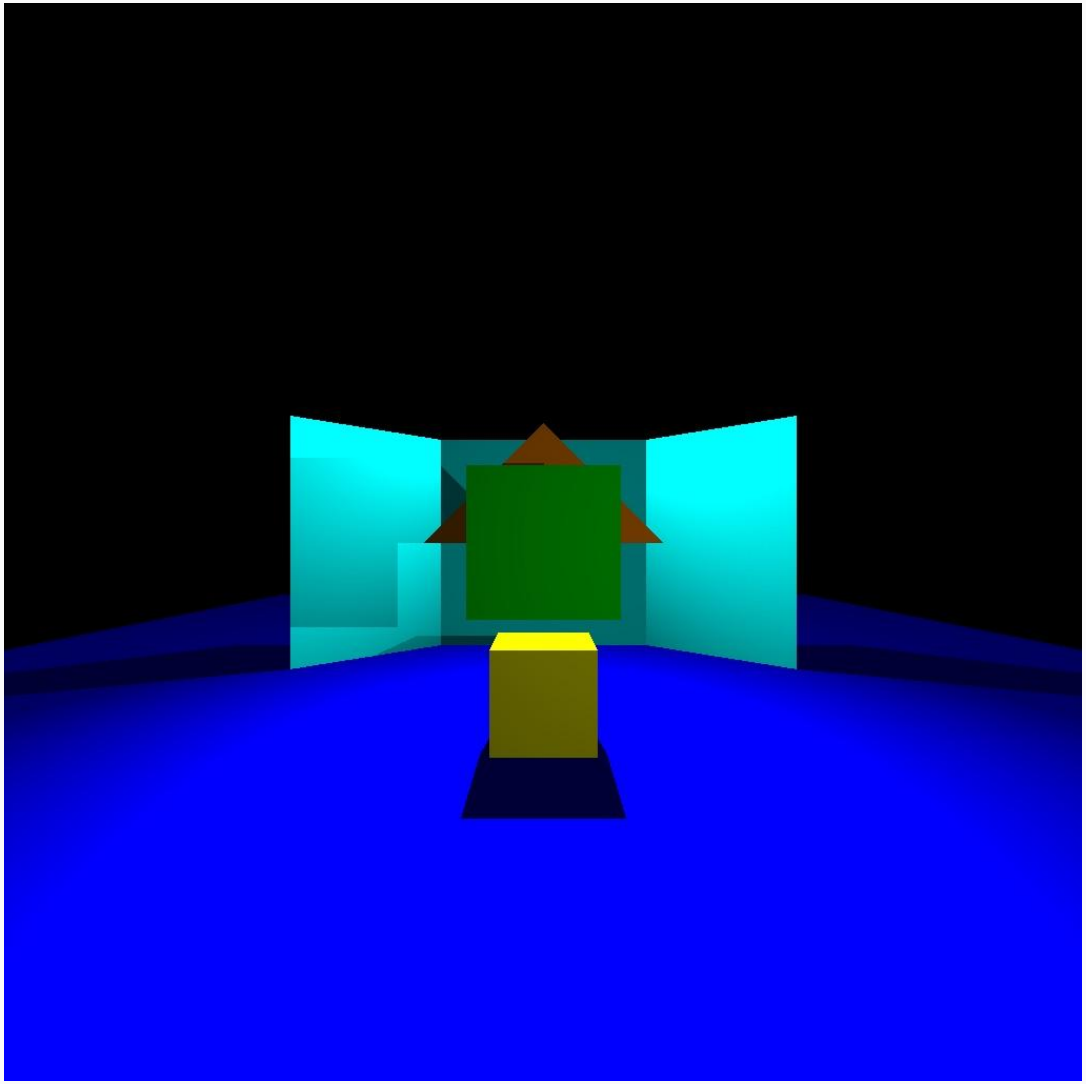
Triangles

Just like rectangles!



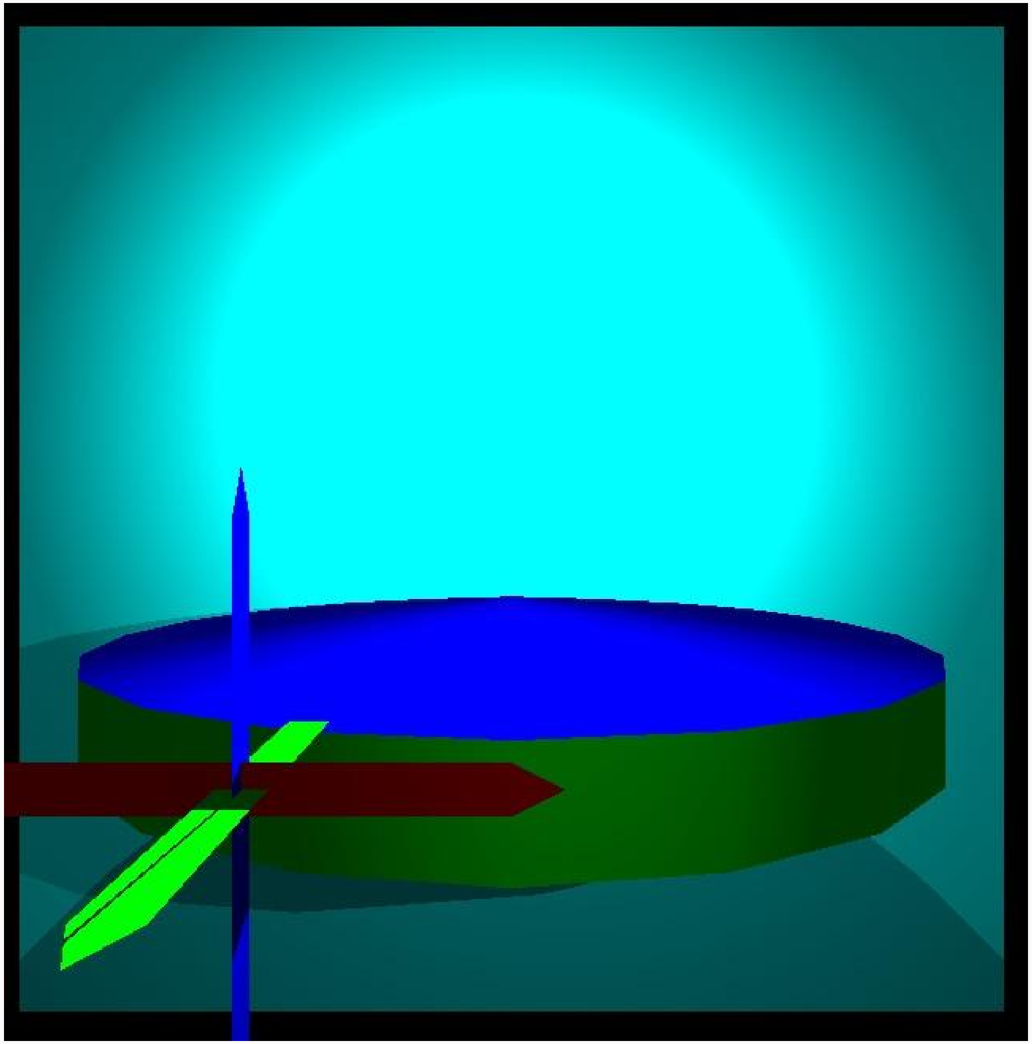
Triangles

Why should we care about triangles?



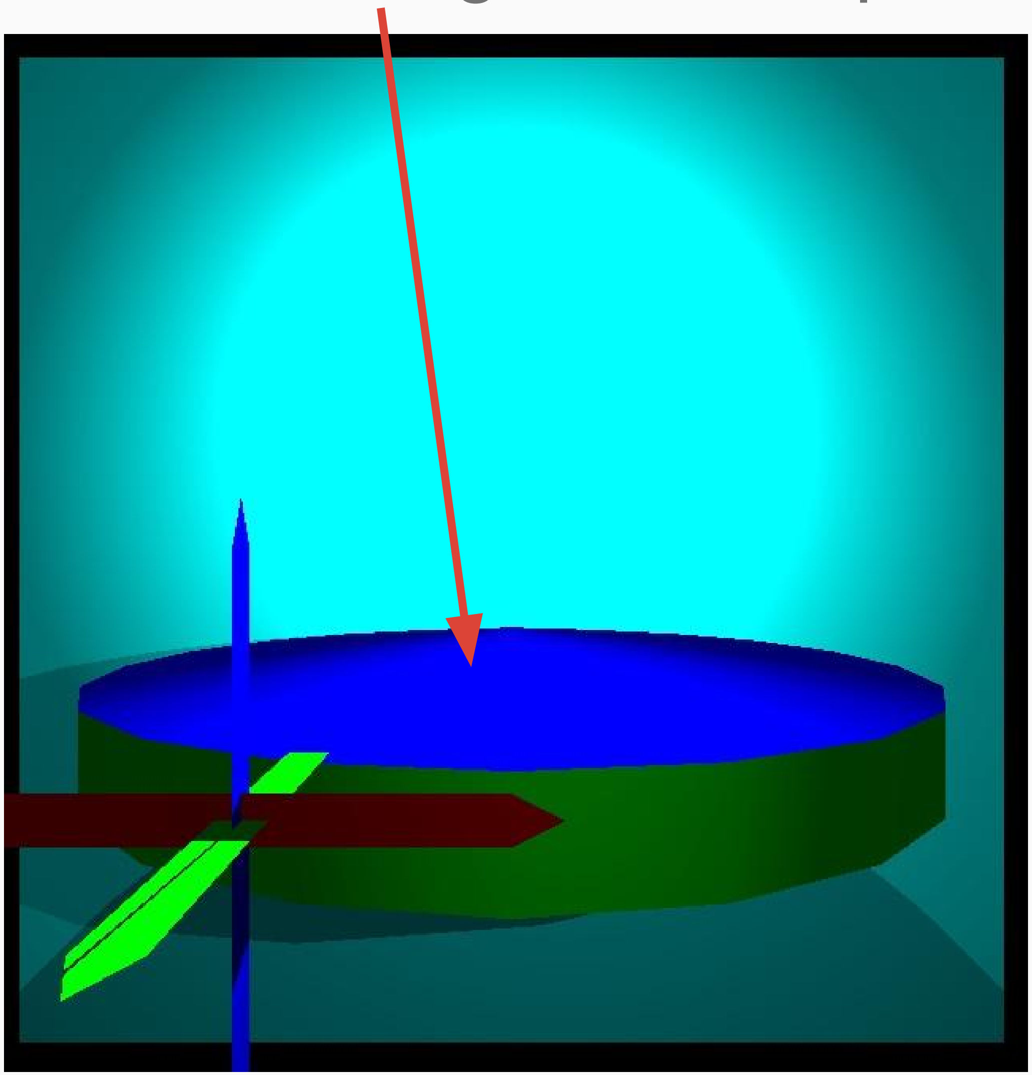
Triangles

There are 40+ triangles in this picture!



Triangles

There are 40+ triangles in this picture!



Parallelism

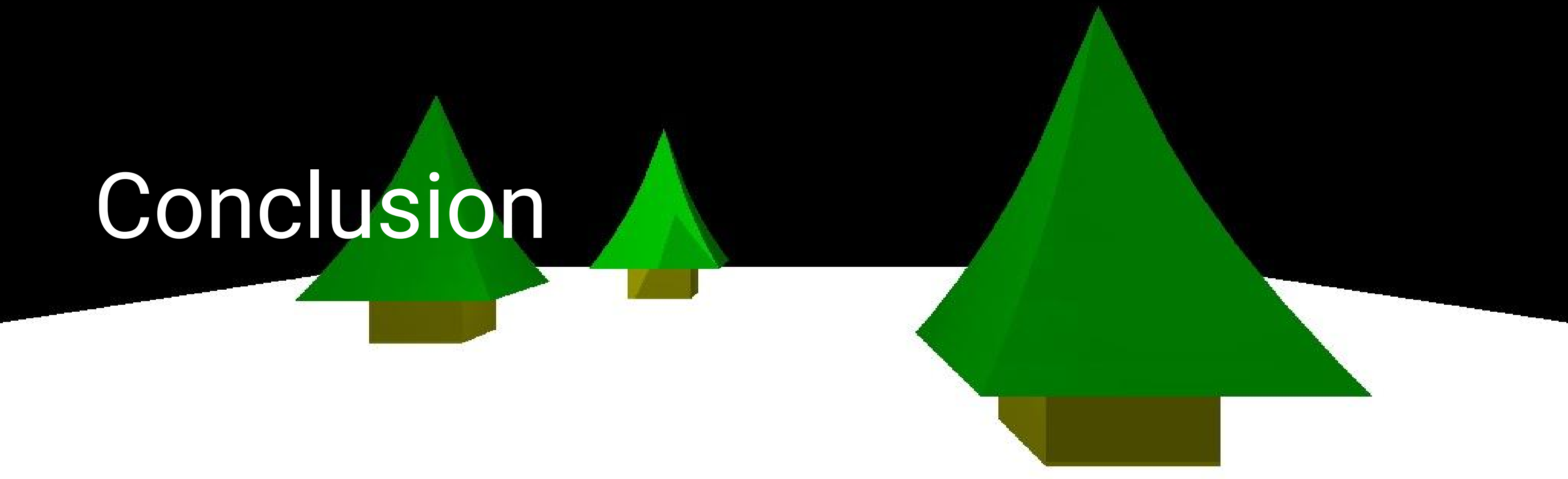
Compute lots of pixels in parallel:
raytracing is almost perfectly parallelisable!

Parallelism

```
$ time dist/build/raytrace/raytrace +RTS -N1
8.47 real      8.36 user      0.05 sys
$ time dist/build/raytrace/raytrace +RTS -N4
3.78 real     11.29 user      0.09 sys
```

~ **2.24** speedup

Conclusion



Conclusion

- ★ We've build a **raytracer**
- ★ in an **hour**
- ★ in less than **300 lines**
- ★ it's easily **extendable**

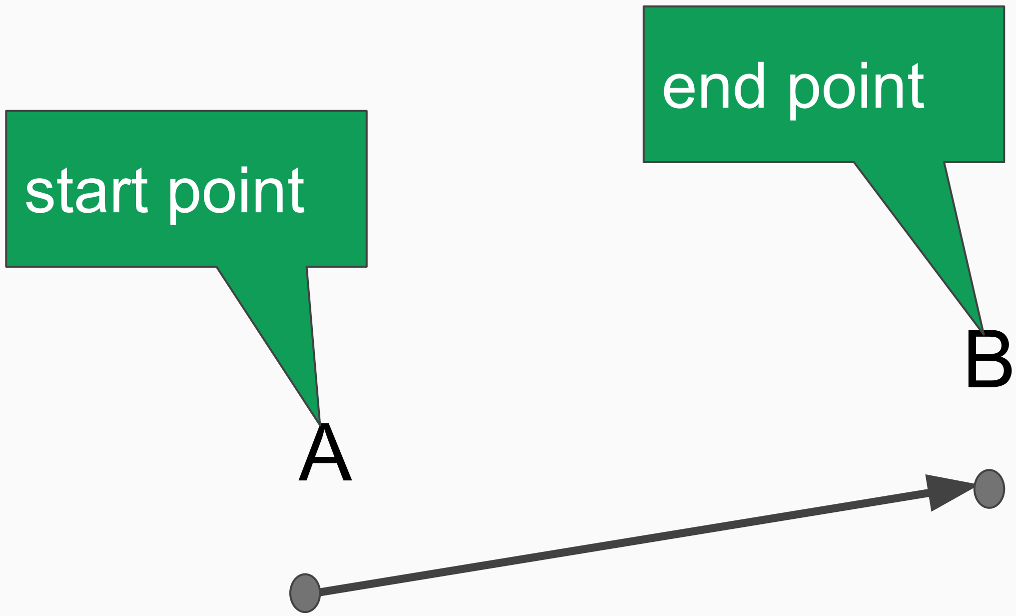
Conclusion

https://bitbucket.org/AlexanderV/raytrace

- what is raytracing
- Vectors (+,scalar,-,*,/,dot- and cross-product, L2 norm)
- Rays
- Shape
    - rectangle definition
    - intersection
    - color
- Camera
- Tracing
- combining shapes
- lights

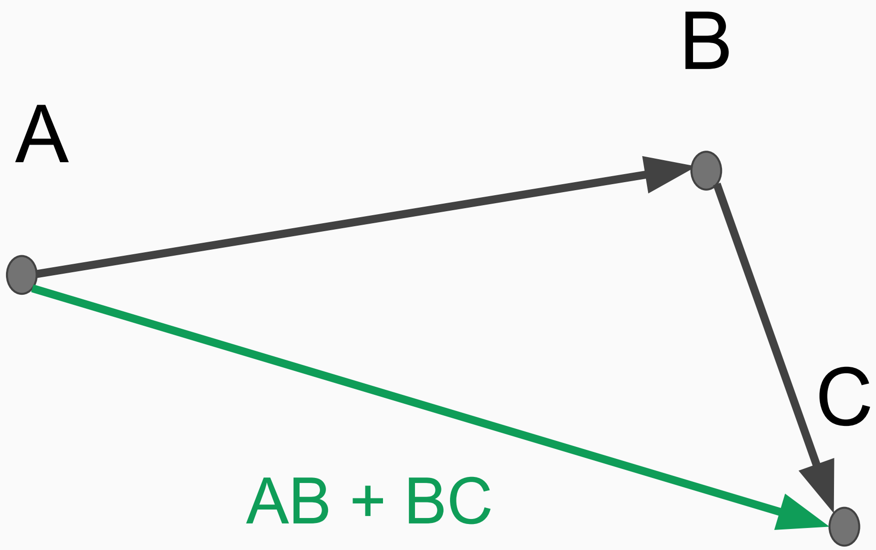
Vectors

2D vectors

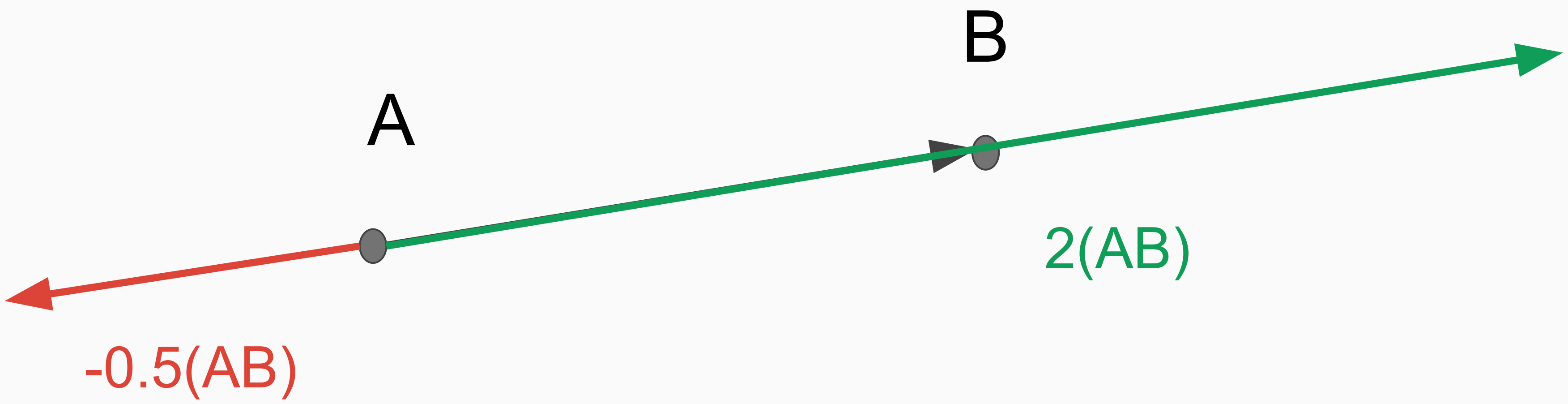


determine:
length + direction

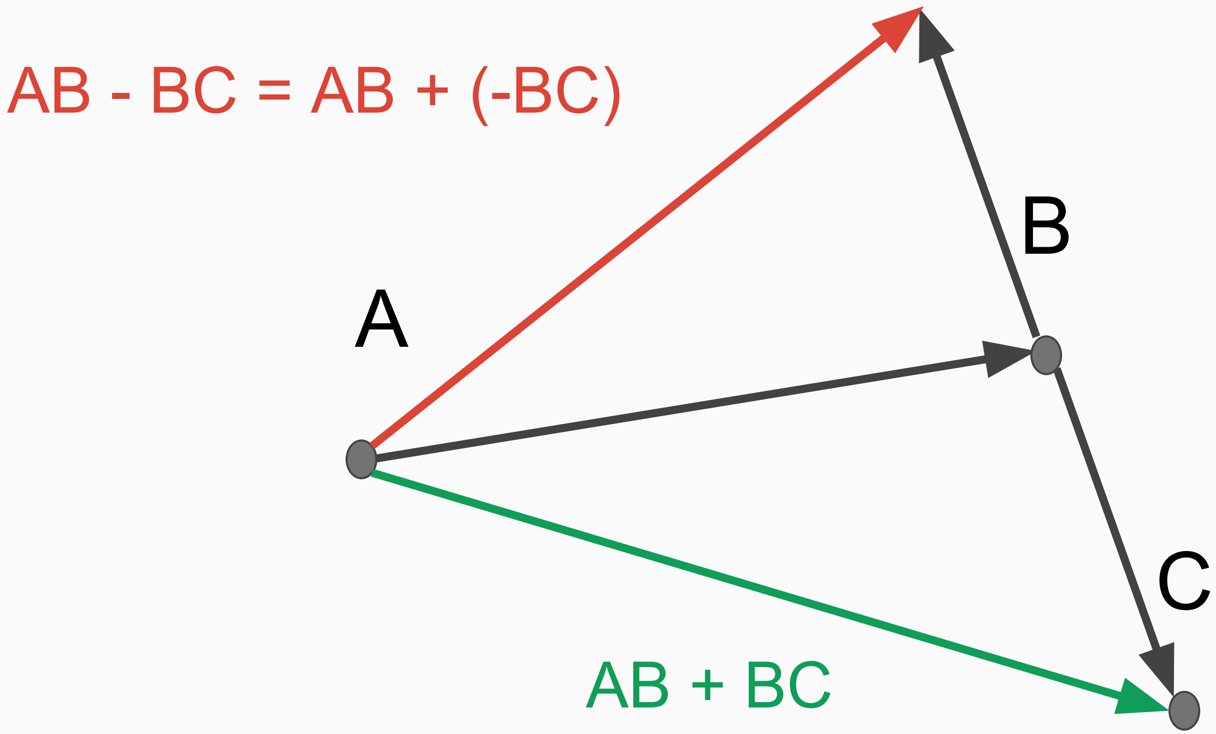
2D vectors



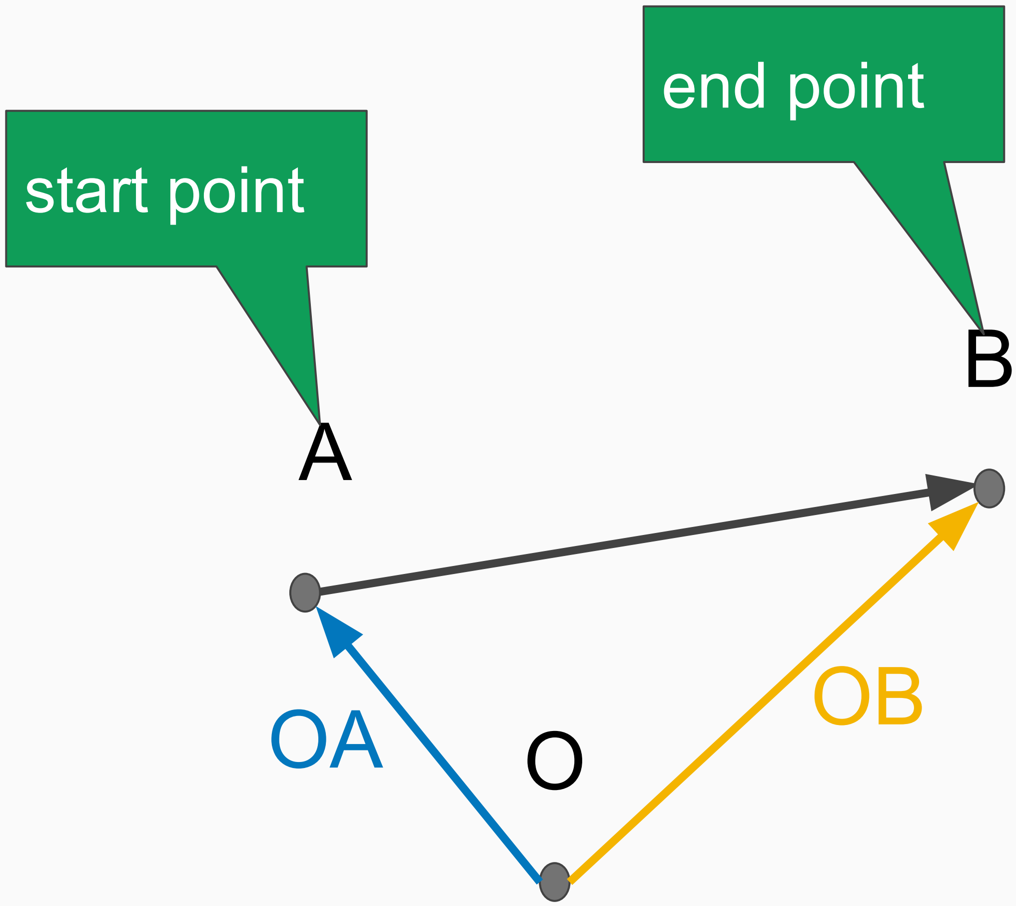
2D vectors - Addition



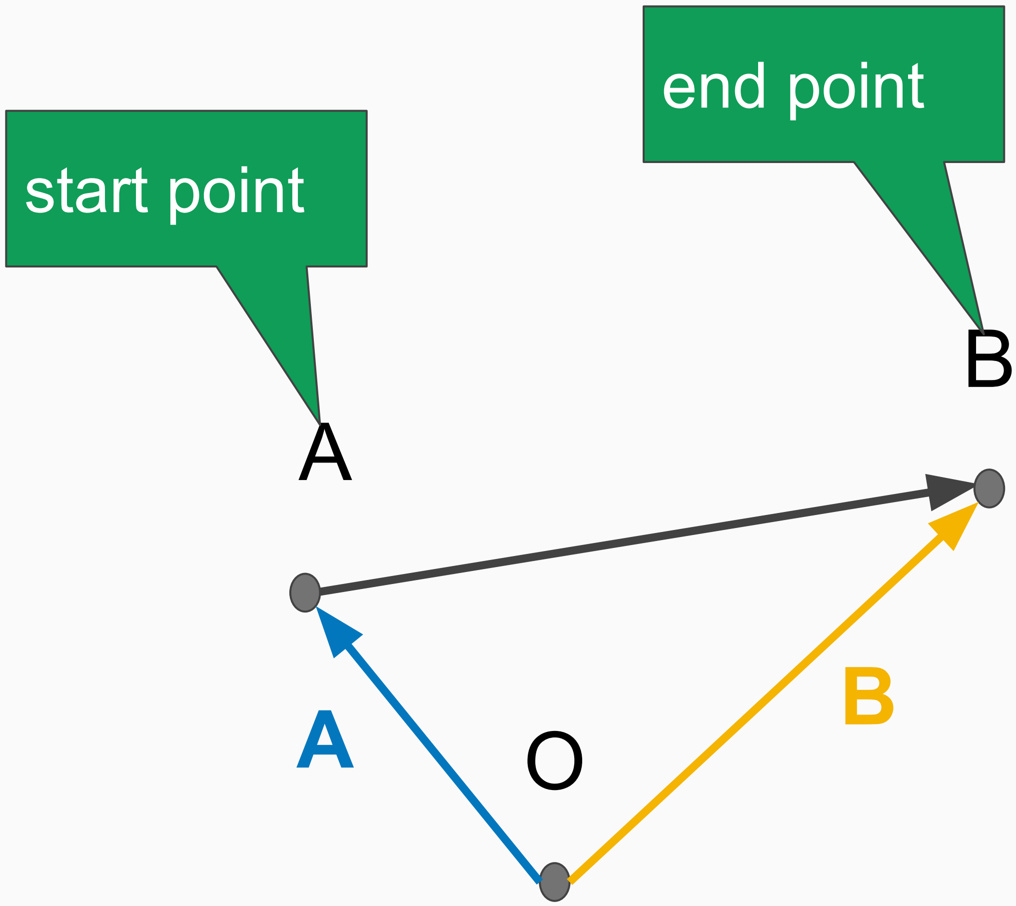
2D vectors - subtraction



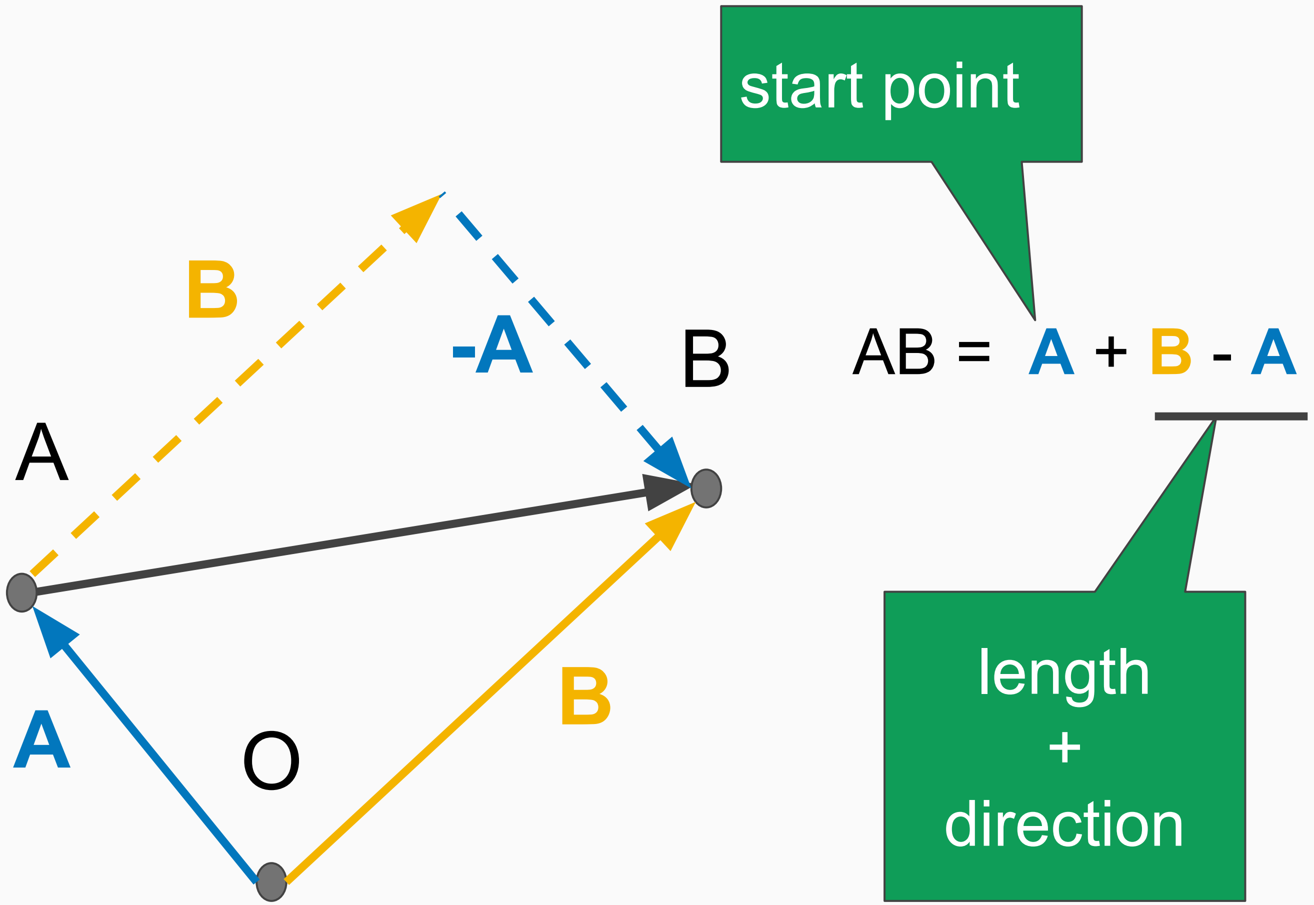
2D vectors - point vector



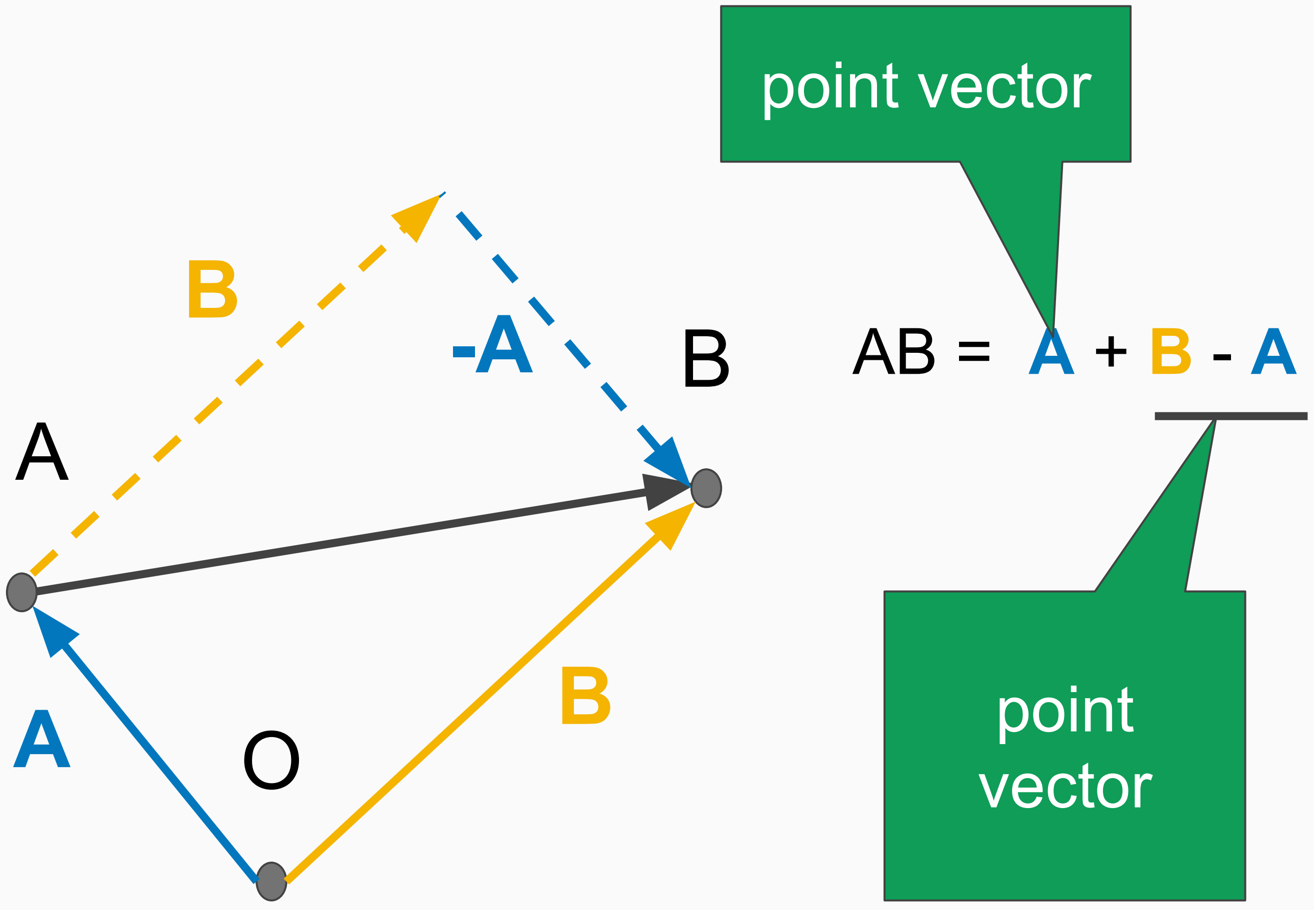
2D vectors - point vector



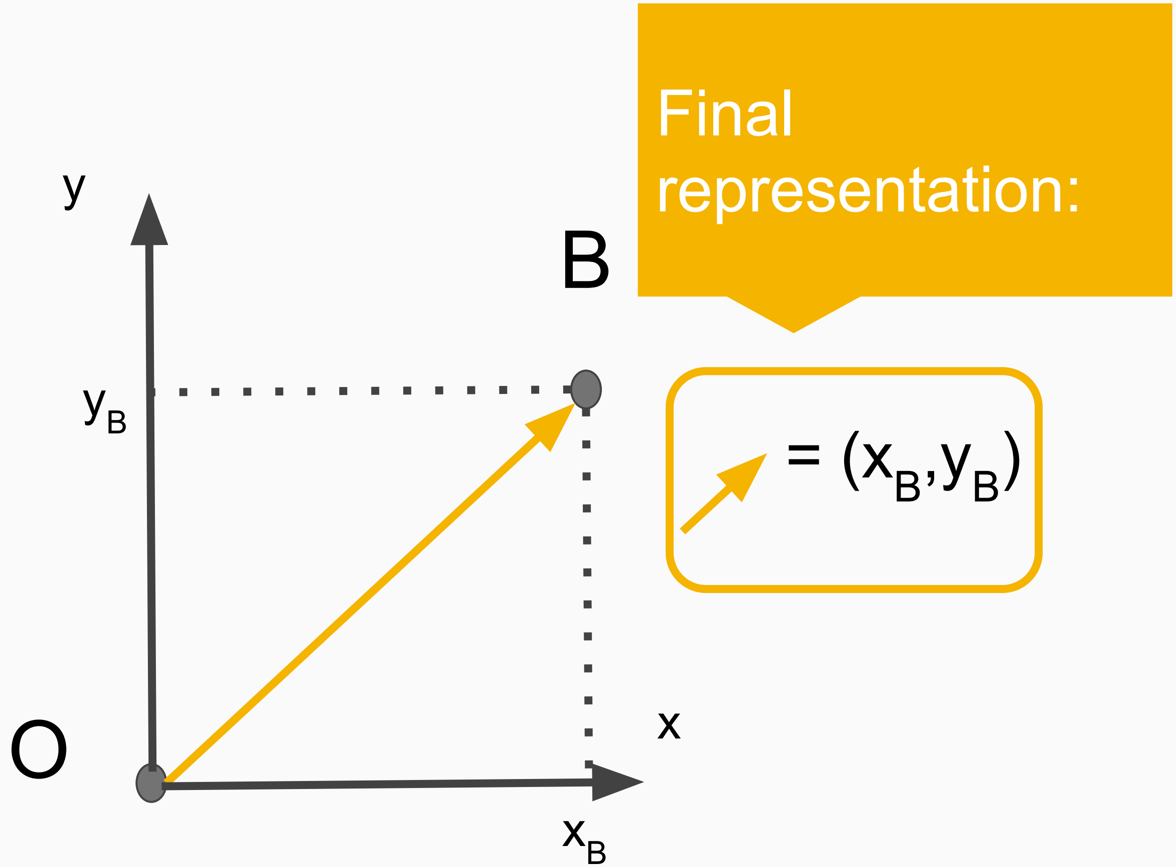
2D vectors - point vector



2D vectors - point vector



2D vectors - point vector



2D vectors

euclidian norm

$$||(x,y)|| = \sqrt{(x^2 + y^2)}$$

length of the vector

inner product

$$(x,y) . (u,v) = x*u+y*v$$

cosine of the angle of the two vectors

3D vectors

euclidian norm

$||(x,y,z)|| = \sqrt{}(x^2 + y^2 + z^2)$

length of the vector

inner product

$(x,y,z) \,.\, (u,v,w) = x*u+y*v + z*w$

cosine of the angle of the two vectors

outer product

(x,y,z) ✖ $(u,v,w) = (p,q,r)$

normal of two vectors

Rays