Lecture 11: Light and Shadow

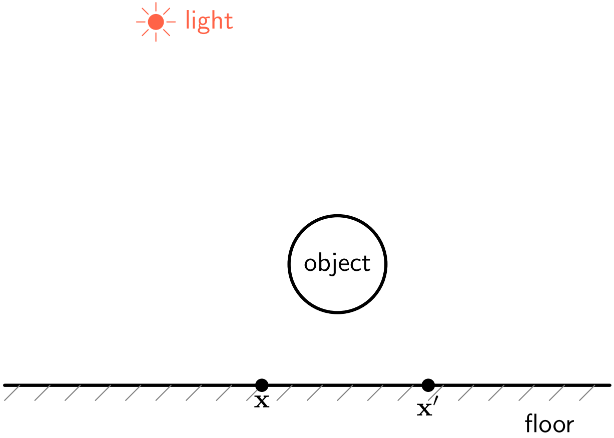
Lights, the Four Vectors, and Shadow Rays

CS 453 — Computer Graphics

Fakhir Shaheen

Forman Christian University

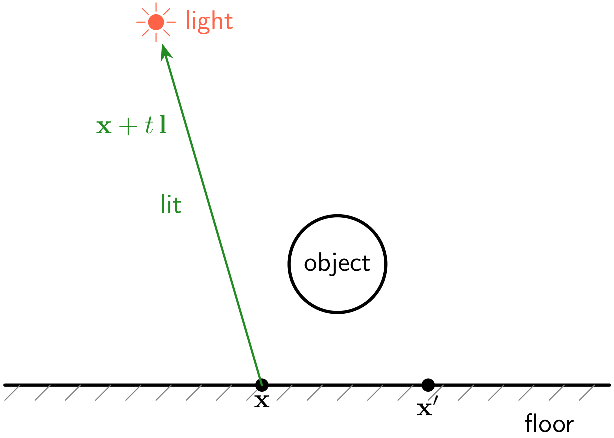
One extra ray tells who is in shadow



- From $\mathbf{x}$, send a shadow ray along $\mathbf{l}$, toward the light:

$$\mathbf{x} + t\,\mathbf{l}$$

One extra ray tells who is in shadow

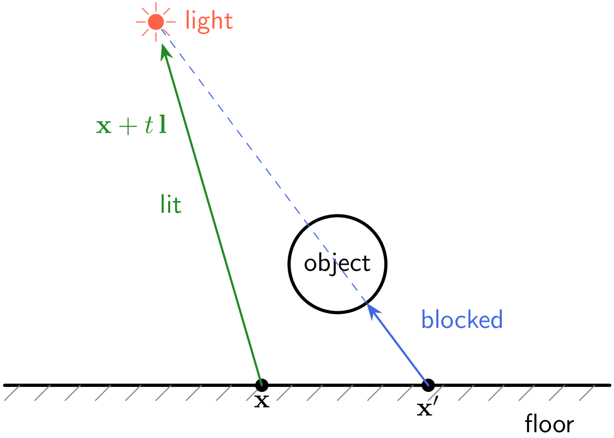


- From $\mathbf{x}$, send a shadow ray along $\mathbf{l}$, toward the light:

$$\mathbf{x} + t\,\mathbf{l}$$

- Nothing in the way: lit

One extra ray tells who is in shadow

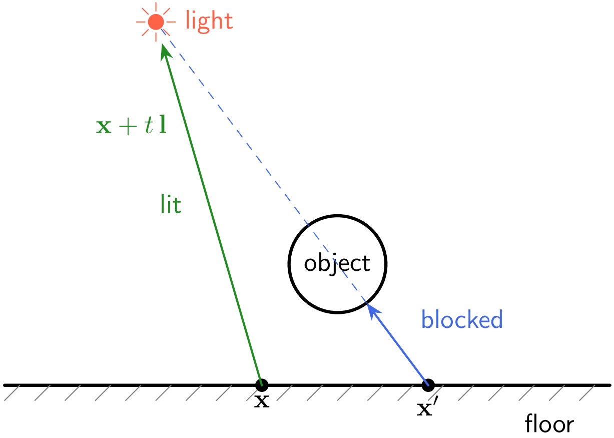


- From $\mathbf{x}$, send a shadow ray along $\mathbf{l}$, toward the light:

$$\mathbf{x} + t\,\mathbf{l}$$

- Nothing in the way: lit
- Something in the way: shadow

One extra ray tells who is in shadow



- From $\mathbf{x}$, send a shadow ray along $\mathbf{l}$, toward the light:

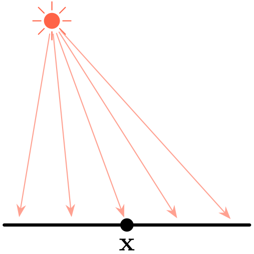
$$\mathbf{x} + t\,\mathbf{l}$$

- Nothing in the way: lit
- Something in the way: shadow
- This is called 'Shadow Ray'

Light sources

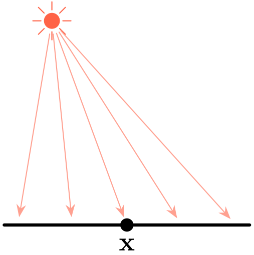
Three kinds, and what each one needs

Three kinds of light

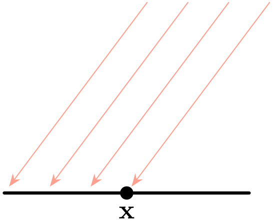


point
from one spot

Three kinds of light

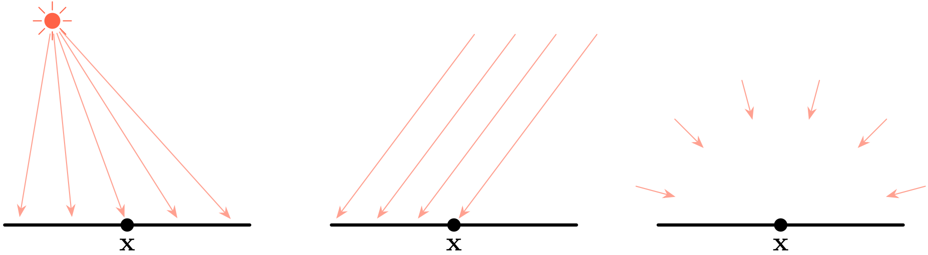


point
from one spot



directional
from one direction, far away

Three kinds of light



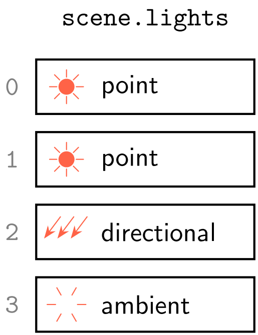
point
from one spot

directional
from one direction, far away

ambient
from everywhere at once

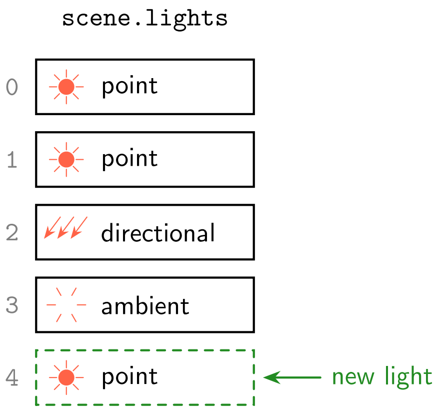
The three differ in one thing: where the light comes from

A ray tracer keeps a list of lights



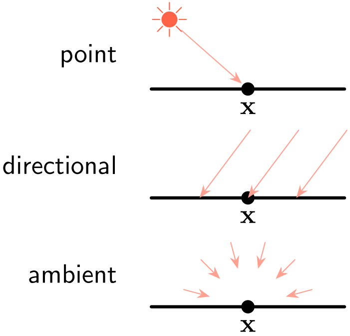
- The scene keeps all its lights in one list
- Any number of lights, any mix of kinds

A ray tracer keeps a list of lights



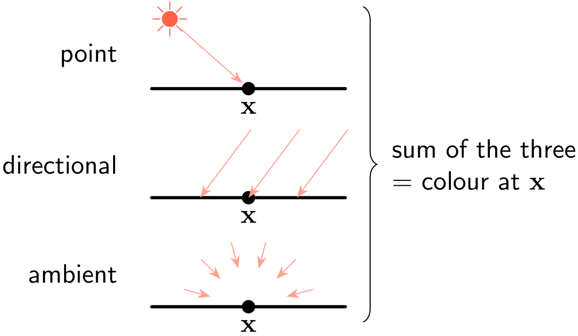
- The scene keeps all its lights in one list
- Any number of lights, any mix of kinds
- A new light is one more entry

Each light adds to the colour at $\mathbf{x}$



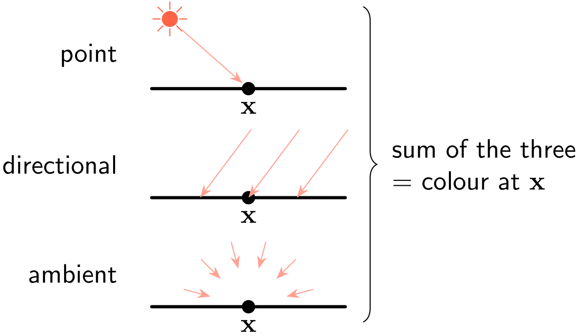
- Each light, on its own, sends some light to $\mathbf{x}$

Each light adds to the colour at $\mathbf{x}$



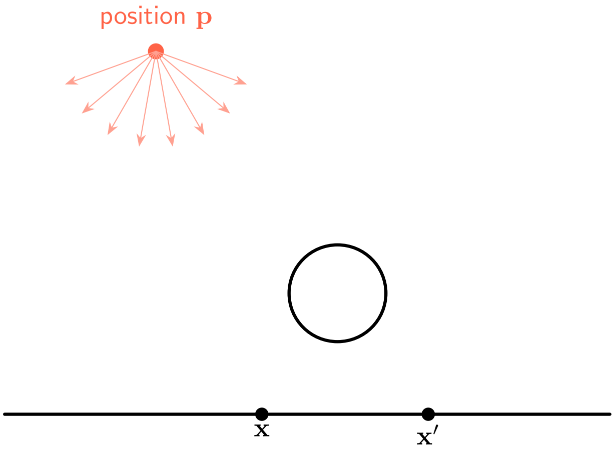
- Each light, on its own, sends some light to $\mathbf{x}$
- Add the shares: the sum is the colour at $\mathbf{x}$

Each light adds to the colour at $\mathbf{x}$



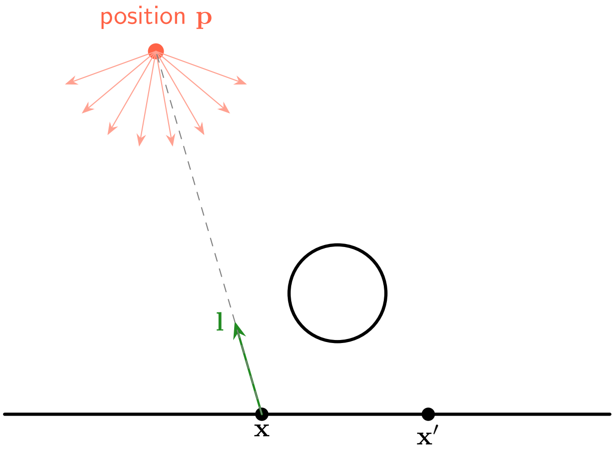
- Each light, on its own, sends some light to $\mathbf{x}$
- Add the shares: the sum is the colour at $\mathbf{x}$
- How big each share is: the shading model, later

A point light shines from one spot



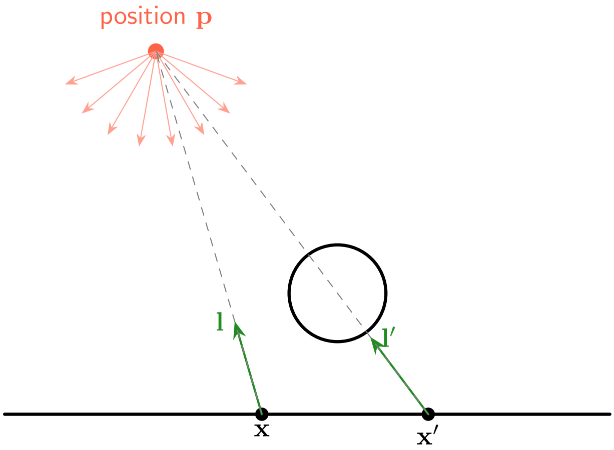
- All light leaves one position $\mathbf{p}$

A point light shines from one spot



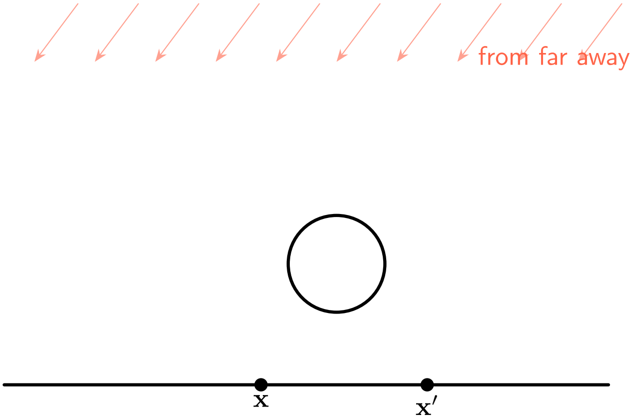
- All light leaves one position $\mathbf{p}$
- Each point looks toward it

A point light shines from one spot



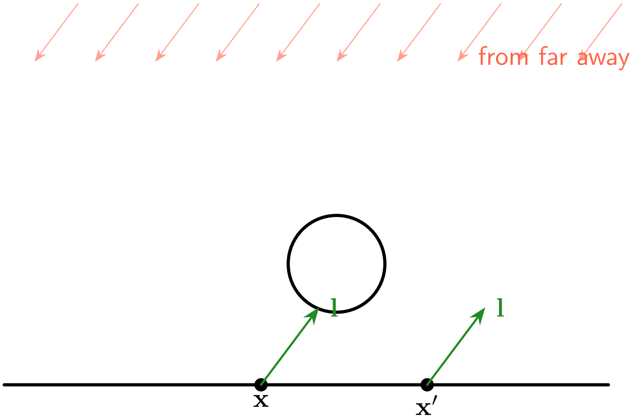
- All light leaves one position $\mathbf{p}$
- Each point looks toward it
- So $\mathbf{l}$ differs from point to point

A directional light: one direction everywhere



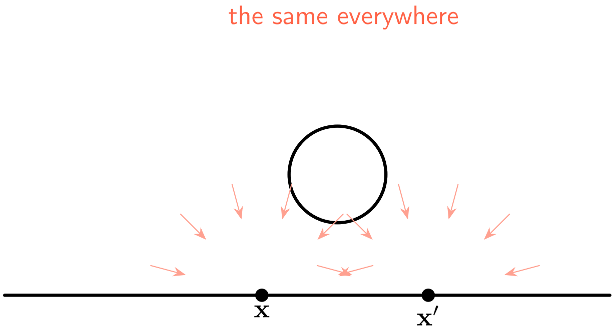
- No position, only a direction

A directional light: one direction everywhere



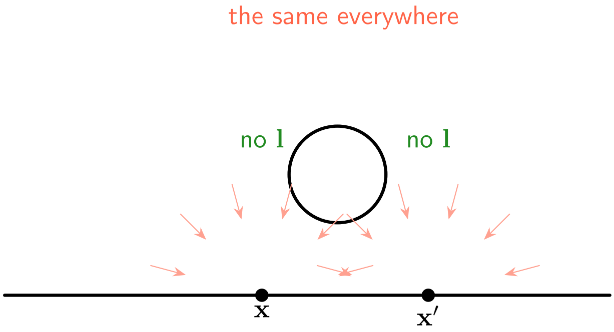
- No position, only a direction
- Same $\mathbf{l}$ at every point
- Think of the sun

Ambient light fills in the shadows



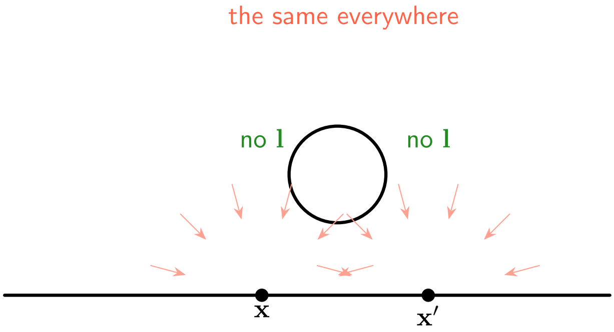
- Constant light from everywhere

Ambient light fills in the shadows



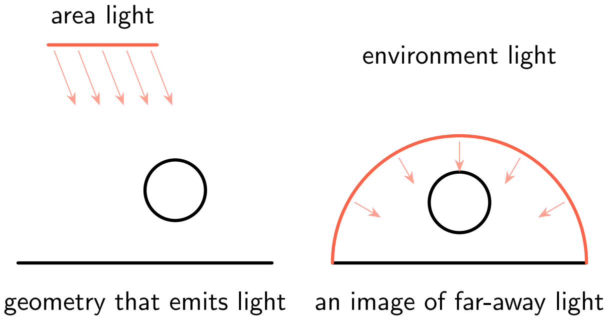
- Constant light from everywhere
- No direction to point at

Ambient light fills in the shadows



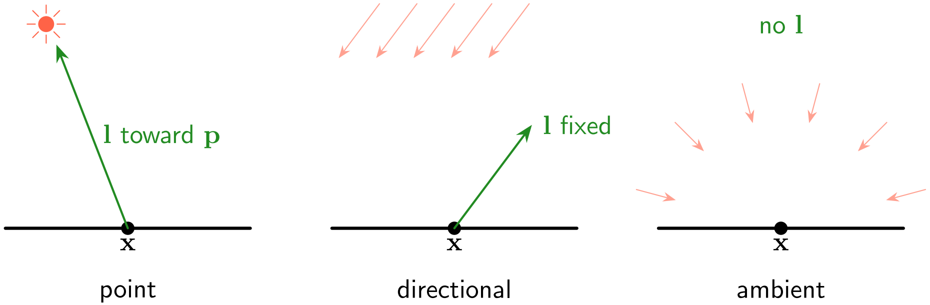
- Constant light from everywhere
- No direction to point at
- Keeps shadows from going black

Fancier lights exist; we skip them



- Area light: geometry that glows
- Environment light: a picture of the sky
- Not needed for our model

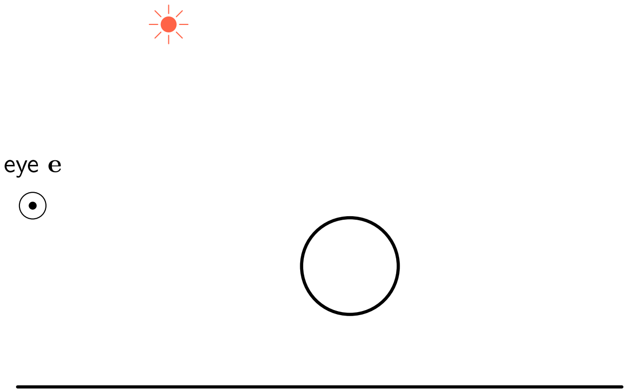
Recap: three lights



The four vectors at the hit

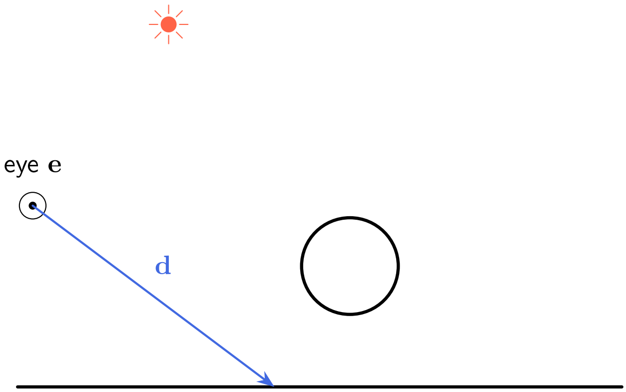
What shading needs from the ray tracer

Shading starts where the viewing ray lands



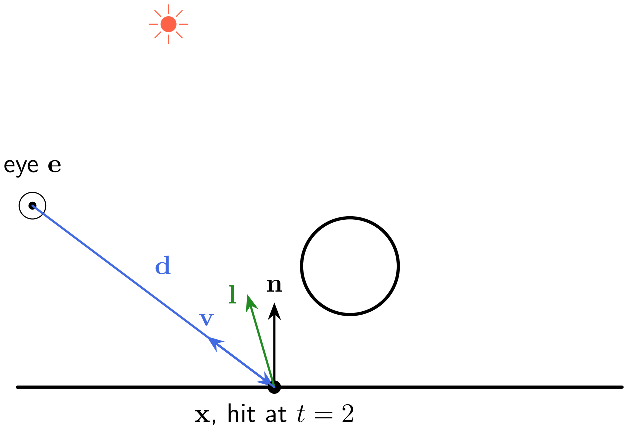
- The eye sends a viewing ray

Shading starts where the viewing ray lands



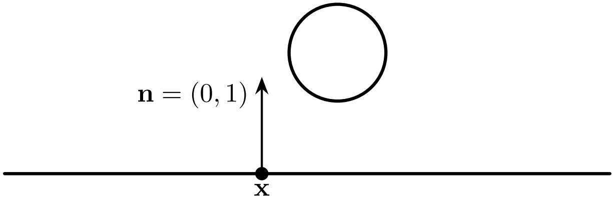
- The eye sends a viewing ray
- It hits the floor at some t

Shading starts where the viewing ray lands



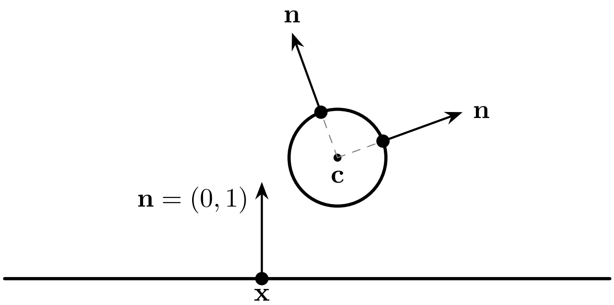
- The eye sends a viewing ray
- It hits the floor at some t
- Shading needs four vectors there:
  - $\mathbf{x}$: the hit point
  - $\mathbf{n}$: the surface normal
  - $\mathbf{l}$: toward the light
  - $\mathbf{v}$: toward the eye

$\mathbf{n}$: each shape knows its own normal



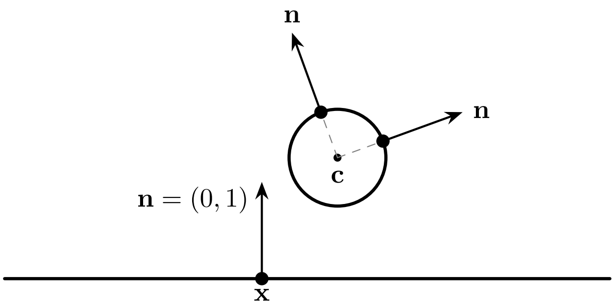
- $\mathbf{n}$: the unit direction the surface faces at the hit
- Floor: straight up, $\mathbf{n} = (0, 1)$

$\mathbf{n}$: each shape knows its own normal



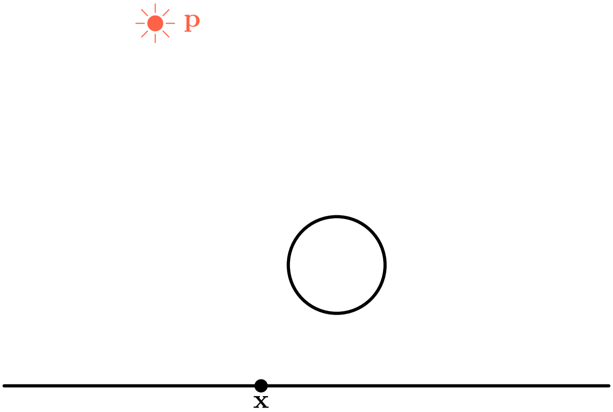
- $\mathbf{n}$: the unit direction the surface faces at the hit
- Floor: straight up, $\mathbf{n} = (0, 1)$
- Circle: from the centre $\mathbf{c}$ out through the hit

$\mathbf{n}$: each shape knows its own normal

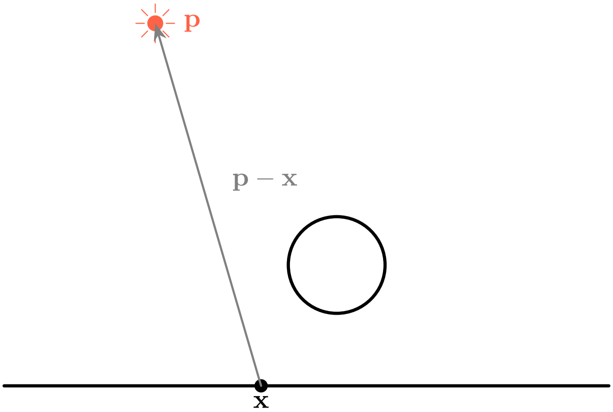


- $\mathbf{n}$: the unit direction the surface faces at the hit
- Floor: straight up, $\mathbf{n} = (0,1)$
- Circle: from the centre $\mathbf{c}$ out through the hit
- Each shape computes its own $\mathbf{n}$ and returns it with the hit

$\mathbf{l}$ points from $\mathbf{x}$ toward the light

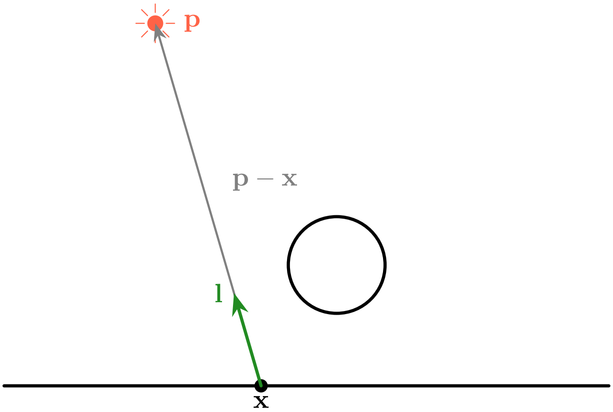


$\mathbf{l}$ points from $\mathbf{x}$ toward the light



aim at the light: $\mathbf{p} - \mathbf{x}$

$\mathbf{l}$ points from $\mathbf{x}$ toward the light

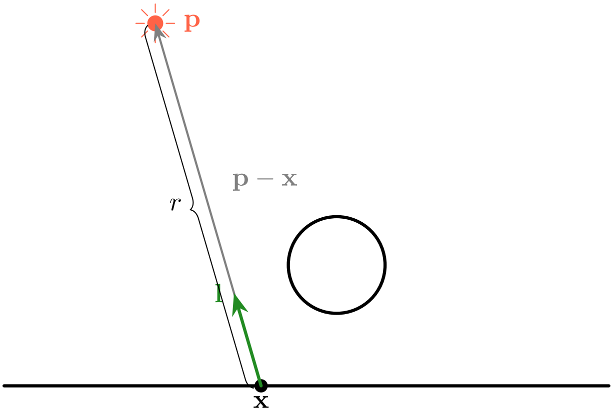


aim at the light: $\mathbf{p} - \mathbf{x}$

$$\mathbf{l} = \frac{\mathbf{p} - \mathbf{x}}{\|\mathbf{p} - \mathbf{x}\|}$$

- Unit length

$\mathbf{l}$ points from $\mathbf{x}$ toward the light



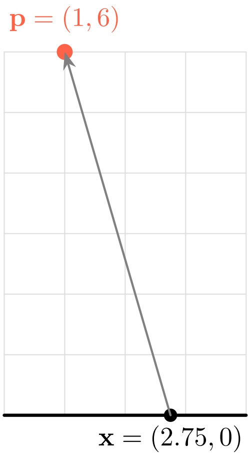
aim at the light: $\mathbf{p} - \mathbf{x}$

$$\mathbf{l} = \frac{\mathbf{p} - \mathbf{x}}{\|\mathbf{p} - \mathbf{x}\|}$$

$$r = \|\mathbf{p} - \mathbf{x}\|$$

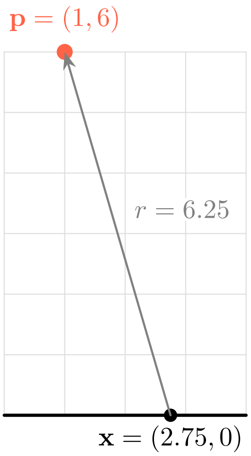
- Unit length
- r: the distance to the light

Worked: $\mathbf{l}$ at $\mathbf{x}$ is $(-0.28,\ 0.96)$



$$\mathbf{p} - \mathbf{x} = (1, 6) - (2.75, 0) = (-1.75,\ 6)$$

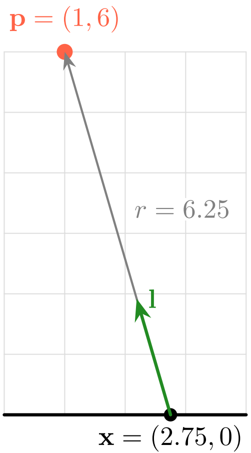
Worked: $\mathbf{l}$ at $\mathbf{x}$ is $(-0.28,\ 0.96)$



$$\mathbf{p} - \mathbf{x} = (1, 6) - (2.75, 0) = (-1.75,\ 6)$$

$$r = \sqrt{1.75^2 + 6^2} = 6.25$$

Worked: $\mathbf{l}$ at $\mathbf{x}$ is $(-0.28,\ 0.96)$

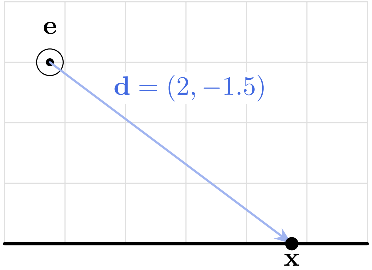


$$\mathbf{p} - \mathbf{x} = (1, 6) - (2.75, 0) = (-1.75,\ 6)$$

$$r = \sqrt{1.75^2 + 6^2} = 6.25$$

$$\mathbf{l} = \frac{(-1.75,\ 6)}{6.25} = (-0.28,\ 0.96)$$

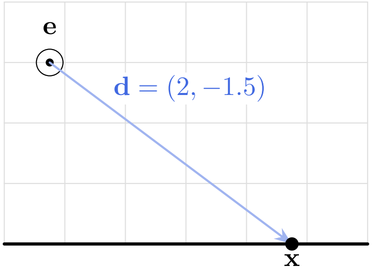
$\mathbf{v}$ points back toward the eye



$$\mathbf{v} = -\frac{\mathbf{d}}{\|\mathbf{d}\|}$$

- Opposite the viewing ray

$\mathbf{v}$ points back toward the eye

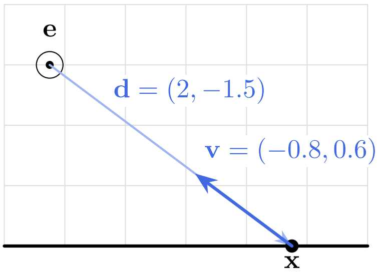


$$\mathbf{v} = -\frac{\mathbf{d}}{\|\mathbf{d}\|}$$

$$\|\mathbf{d}\| = \sqrt{2^2 + 1.5^2} = 2.5$$

- Opposite the viewing ray

$\mathbf{v}$ points back toward the eye



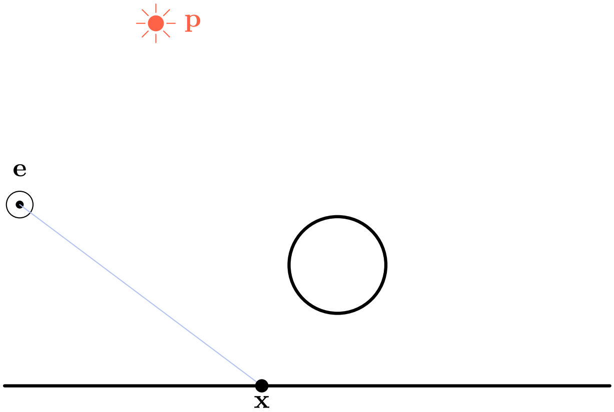
$$\mathbf{v} = -\frac{\mathbf{d}}{\|\mathbf{d}\|}$$

$$\|\mathbf{d}\| = \sqrt{2^2 + 1.5^2} = 2.5$$

$$\mathbf{v} = -\frac{(2, -1.5)}{2.5} = (-0.8,\ 0.6)$$

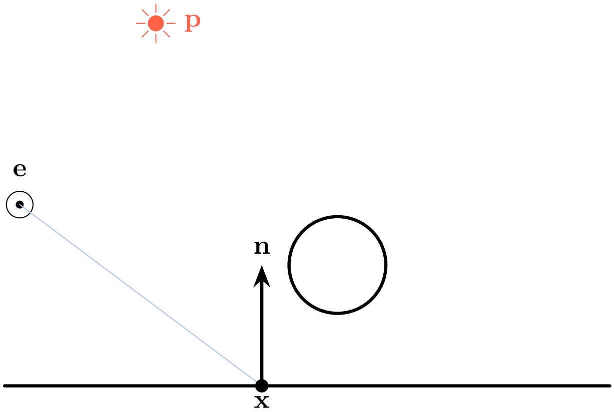
- Opposite the viewing ray
- Unit length again

All four vectors at one hit point



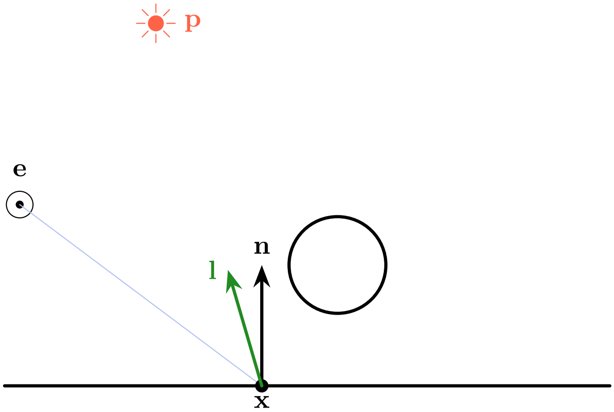
- $\mathbf{x} = (2.75,\, 0)$: the hit point

All four vectors at one hit point



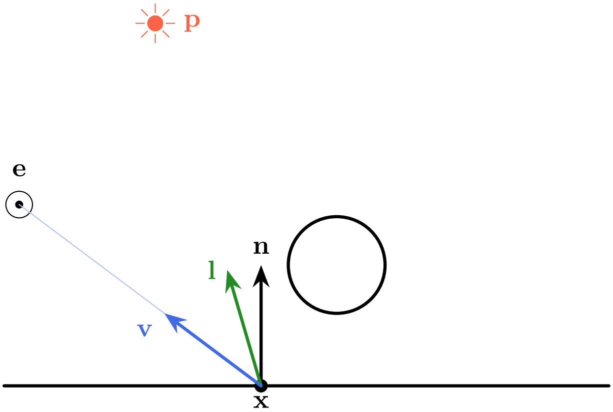
- $\mathbf{x} = (2.75,\, 0)$: the hit point
- $\mathbf{n} = (0,\, 1)$: the surface normal

All four vectors at one hit point



- $\mathbf{x} = (2.75,\, 0)$: the hit point
- $\mathbf{n} = (0,\, 1)$: the surface normal
- $\mathbf{l} = (-0.28,\, 0.96)$: toward the light

All four vectors at one hit point



- $\mathbf{x} = (2.75, 0)$: the hit point
- $\mathbf{n} = (0, 1)$: the surface normal
- $\mathbf{l} = (-0.28, 0.96)$: toward the light
- $\mathbf{v} = (-0.8, 0.6)$: toward the eye

Ambient light needs none of the four vectors

light	$\mathbf{x}$	$\mathbf{n}$	$\mathbf{l}$	$\mathbf{v}$
point	✓	✓	✓	✓
directional	✓	✓	✓	✓

Ambient light needs none of the four vectors

light	$\mathbf{x}$	$\mathbf{n}$	$\mathbf{l}$	$\mathbf{v}$
point	✓	✓	✓	✓
directional	✓	✓	✓	✓
ambient	–	–	–	–

Ambient light arrives equally at every point, from every direction

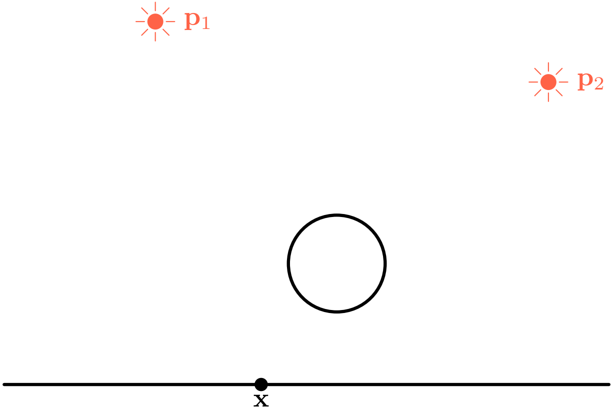
Ambient light needs none of the four vectors

light	$\mathbf{x}$	$\mathbf{n}$	$\mathbf{l}$	$\mathbf{v}$
point	✓	✓	✓	✓
directional	✓	✓	✓	✓
ambient	–	–	–	–

Ambient light arrives equally at every point, from every direction

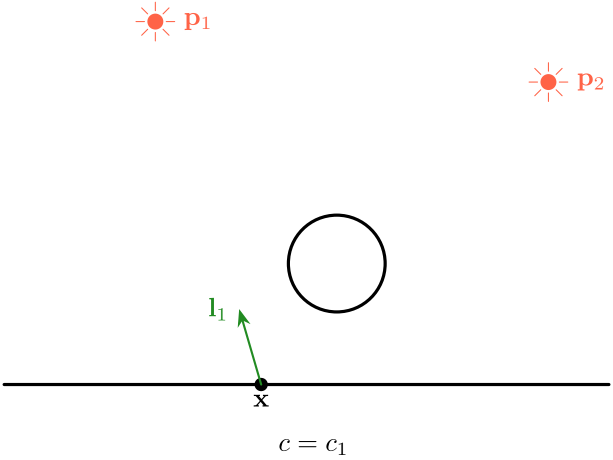
So none of $\mathbf{x}, \mathbf{n}, \mathbf{l}, \mathbf{v}$ changes its share

Several lights: add up the contributions



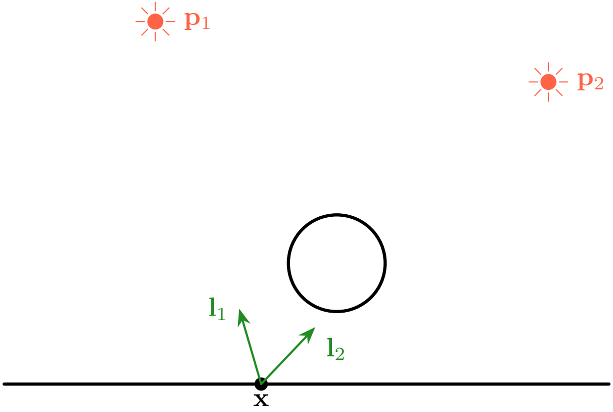
- Loop over the list

Several lights: add up the contributions



- Loop over the list
- Compute shading from each one

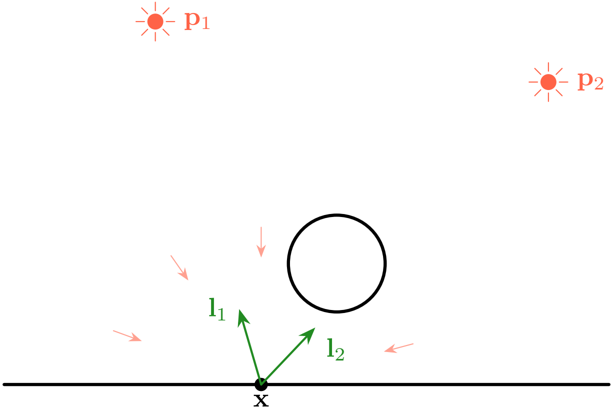
Several lights: add up the contributions



$$c = c_1 + c_2$$

- Loop over the list
- Compute shading from each one

Several lights: add up the contributions



$$c = c_1 + c_2 + c_{\text{ambient}}$$

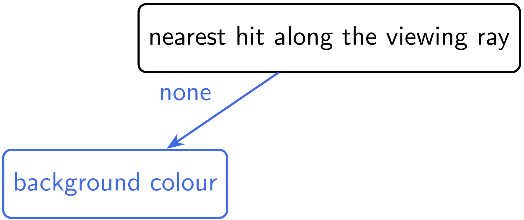
- Loop over the list
- Compute shading from each one
- Accumulate into the pixel colour

Shading one viewing ray, step by step

$$\boxed{\text{nearest hit along the viewing ray}}$$

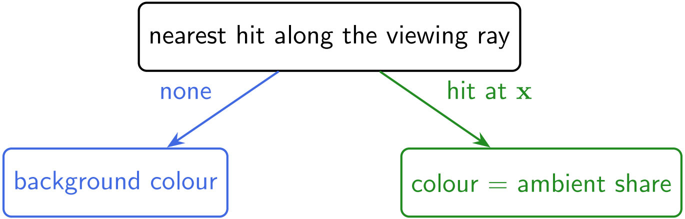
- First find the nearest hit

Shading one viewing ray, step by step



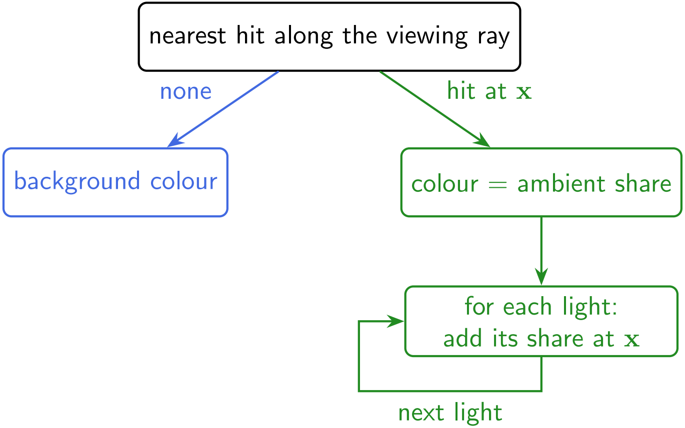
- First find the nearest hit
- Nothing hit: background colour

Shading one viewing ray, step by step



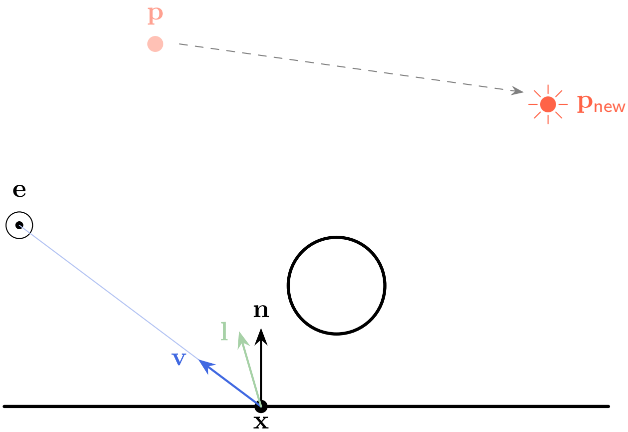
- First find the nearest hit
- Nothing hit: background colour
- Hit at $\mathbf{x}$: start from the ambient share

Shading one viewing ray, step by step



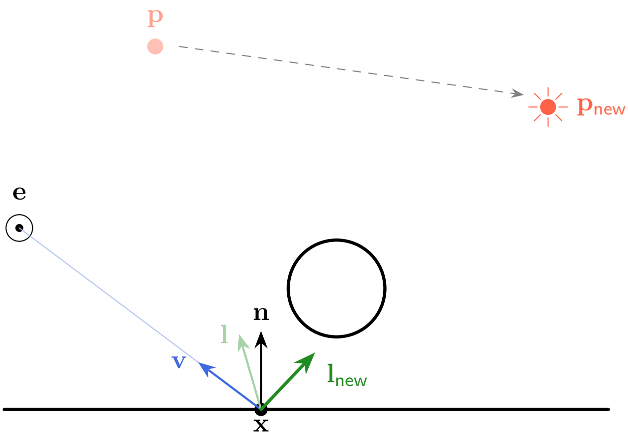
- First find the nearest hit
- Nothing hit: background colour
- Hit at $\mathbf{x}$: start from the ambient share
- Then add each light's share, one light at a time

Quick check 1: the light moves



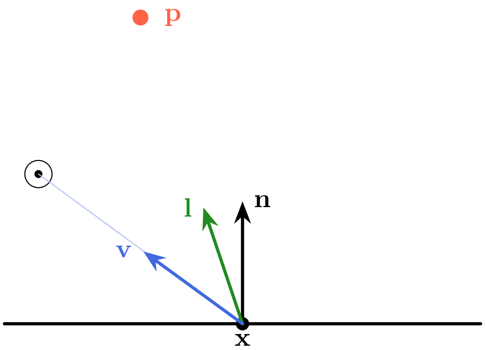
- Same eye, same hit point
- Which of $\mathbf{x}, \mathbf{n}, \mathbf{l}, \mathbf{v}$ change?

Quick check 1: the light moves



- Same eye, same hit point
- Which of $\mathbf{x}, \mathbf{n}, \mathbf{l}, \mathbf{v}$ change?
- Only $\mathbf{l}$

Recap: the four vectors



$$\mathbf{x} = \mathbf{e} + t\,\mathbf{d}$$

$\mathbf{n}$: from the surface that was hit

$$\mathbf{l} = \frac{\mathbf{p} - \mathbf{x}}{\|\mathbf{p} - \mathbf{x}\|}, \quad r = \|\mathbf{p} - \mathbf{x}\|$$

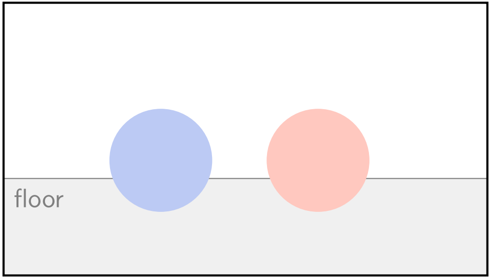
$$\mathbf{v} = -\frac{\mathbf{d}}{\|\mathbf{d}\|}$$

Shadows

One more ray, toward the light

Without shadows, objects seem to float

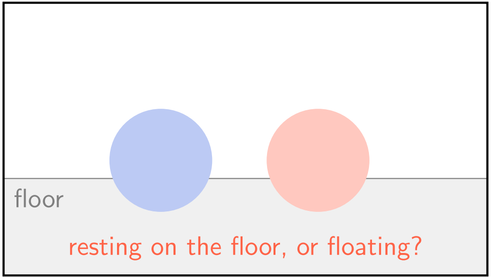
rendered image, no shadows



- Two balls, each one flat colour

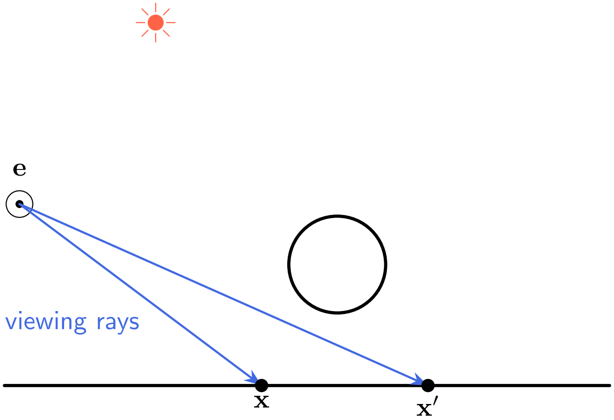
Without shadows, objects seem to float

rendered image, no shadows



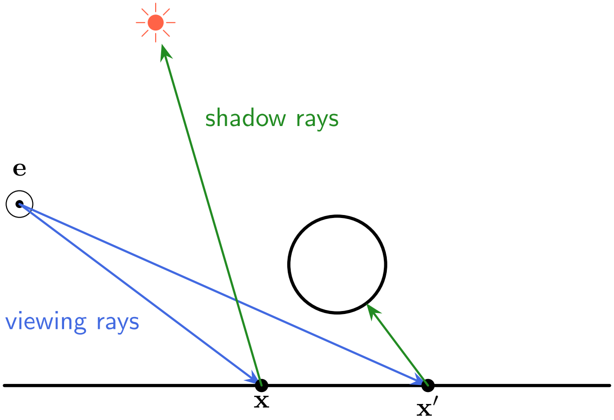
- Two balls, each one flat colour
- On the floor, or above it?
- Without shadows, we cannot tell

Shadow rays are not viewing rays



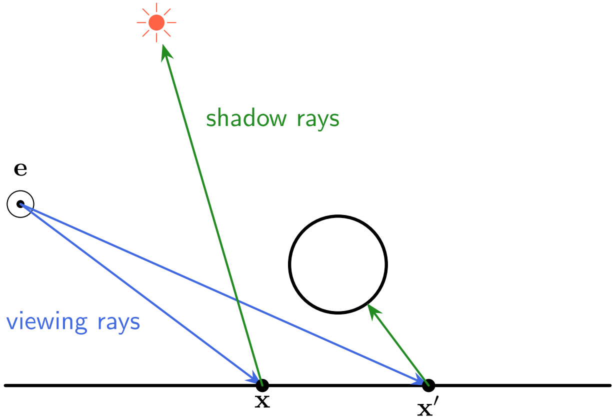
- Viewing ray: eye to surface

Shadow rays are not viewing rays



- Viewing ray: eye to surface
- Shadow ray: surface to light

Shadow rays are not viewing rays



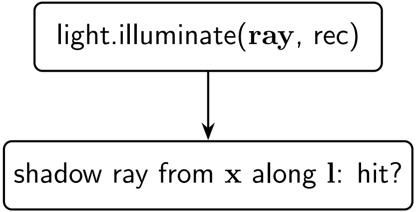
- Viewing ray: eye to surface
- Shadow ray: surface to light
- Same formula, new job

Add one test before shading each light

$\boxed{\text{light.illuminate}(\mathbf{ray}, \text{ rec})}$

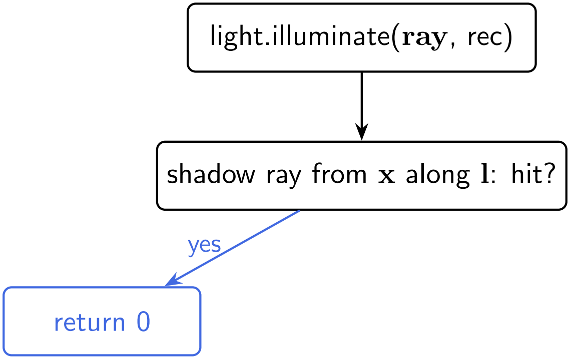
- Inside each light's illuminate

Add one test before shading each light



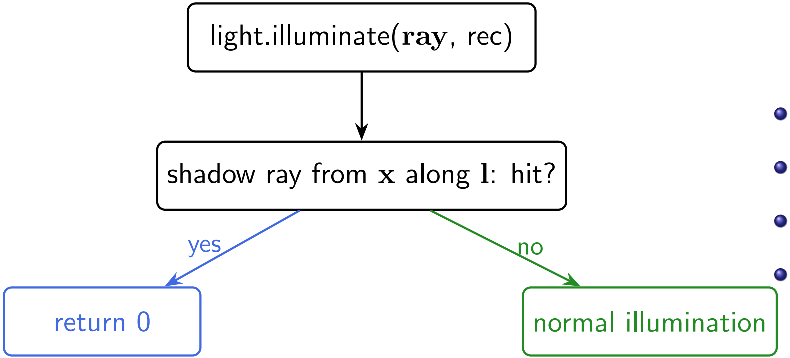
- Inside each light's illuminate
- Test first, shade second

Add one test before shading each light



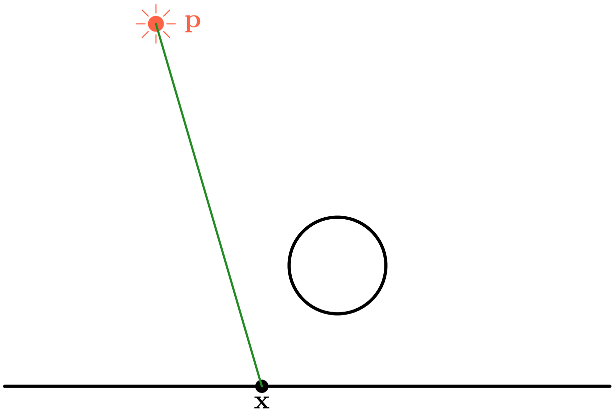
- Inside each light's illuminate
- Test first, shade second
- Blocked: this light gives 0

Add one test before shading each light



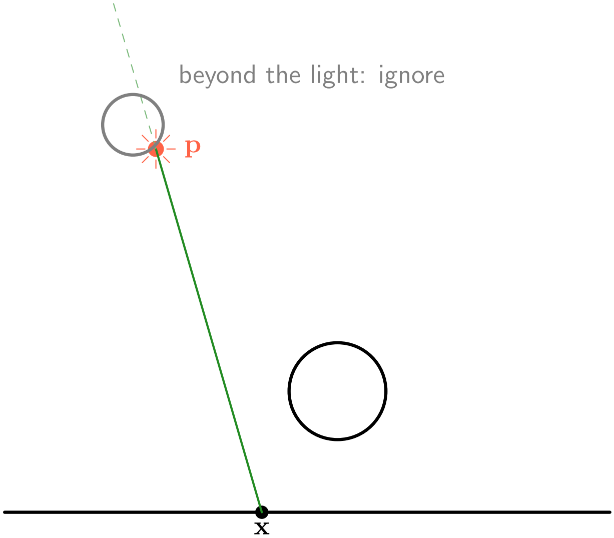
- Inside each light's illuminate
- Test first, shade second
- Blocked: this light gives 0
- Clear: shade as before

The shadow ray must stop at the light



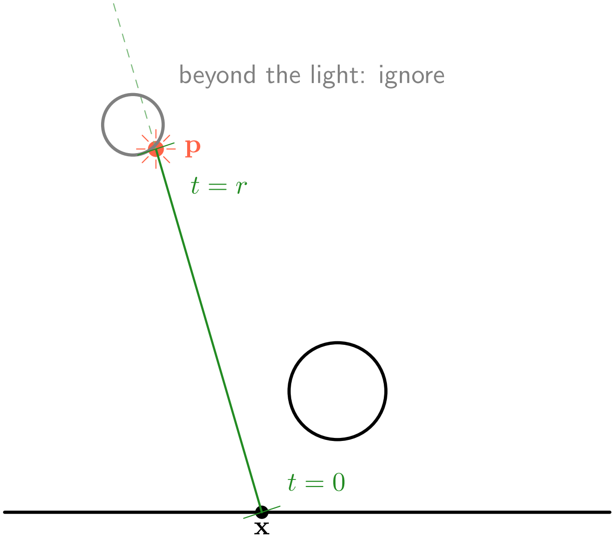
- Only between $\mathbf{x}$ and the light

The shadow ray must stop at the light



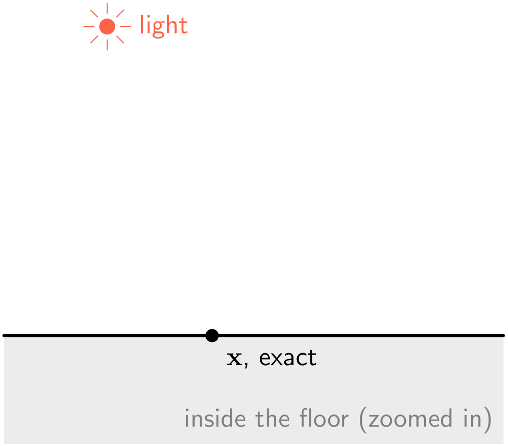
- Only between $\mathbf{x}$ and the light
- Past the light does not matter

The shadow ray must stop at the light



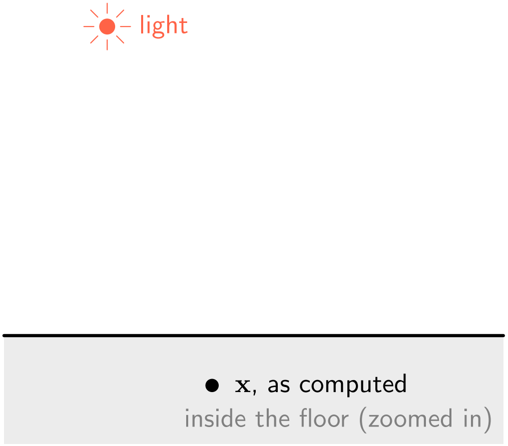
- Only between $\mathbf{x}$ and the light
- Past the light does not matter
- Test $t \in [0, r]$

The trap: the ray hits its own floor



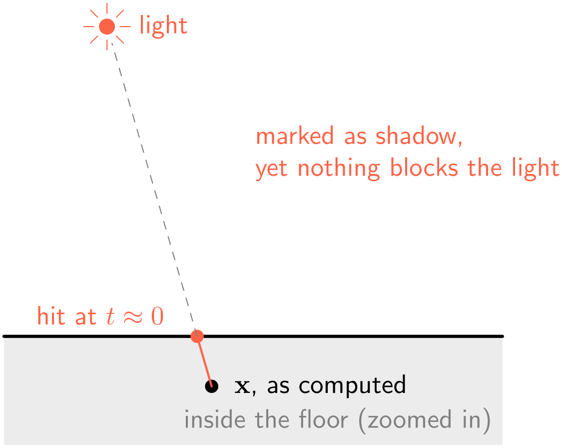
- $\mathbf{x}$ should lie exactly on the floor

The trap: the ray hits its own floor



- $\mathbf{x}$ should lie exactly on the floor
- Round-off can leave it a hair below

The trap: the ray hits its own floor

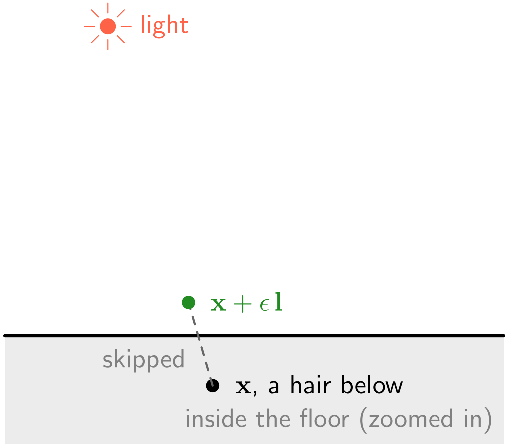


- $\mathbf{x}$ should lie exactly on the floor
- Round-off can leave it a hair below
- The shadow ray then hits the floor at $t \approx 0$
- So $\mathbf{x}$ counts as shadowed: *shadow acne*

Start the shadow ray at ϵ

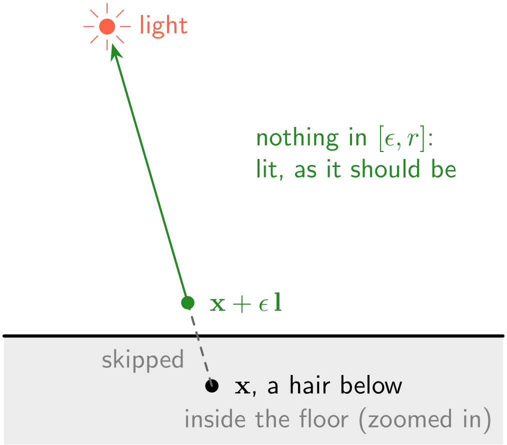
- Start the search a tiny step along the ray

Start the shadow ray at ϵ



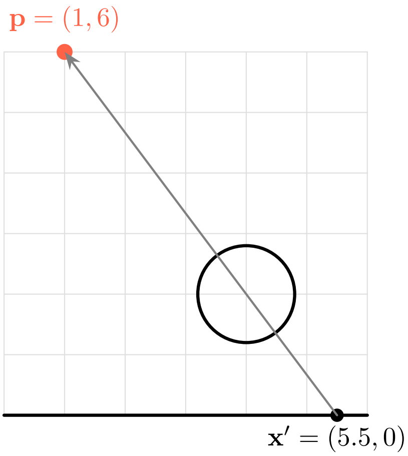
- Start the search a tiny step along the ray
- ϵ: small, but larger than round-off

Start the shadow ray at ϵ



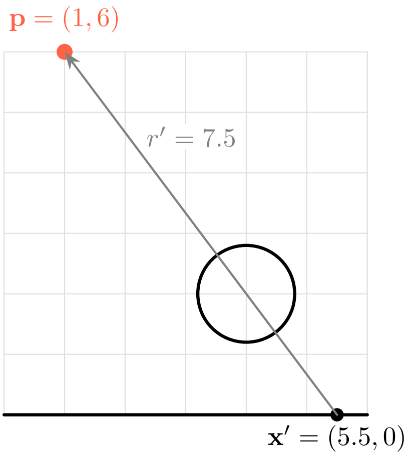
- Start the search a tiny step along the ray
- ϵ: small, but larger than round-off
- Test $t \in [\epsilon, r]$ instead of $[0, r]$

Worked: $\mathbf{l}'$ and r' at $\mathbf{x}'$



$$\mathbf{p} - \mathbf{x}' = (1, 6) - (5.5, 0) = (-4.5,\ 6)$$

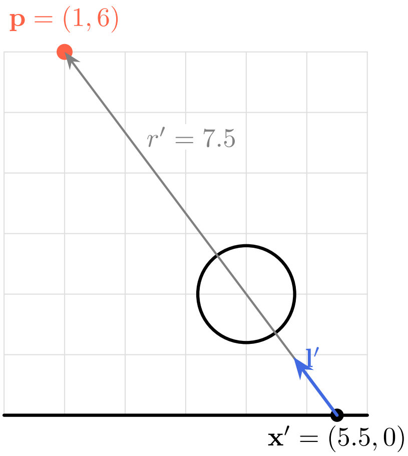
Worked: $\mathbf{l}'$ and r' at $\mathbf{x}'$



$$\mathbf{p} - \mathbf{x}' = (1, 6) - (5.5, 0) = (-4.5,\ 6)$$

$$r' = \sqrt{4.5^2 + 6^2} = 7.5$$

Worked: $\mathbf{l}'$ and r' at $\mathbf{x}'$

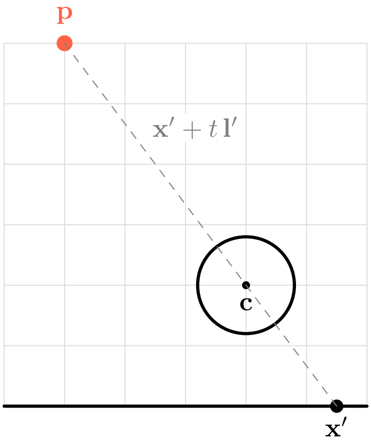


$$\mathbf{p} - \mathbf{x}' = (1, 6) - (5.5, 0) = (-4.5,\ 6)$$

$$r' = \sqrt{4.5^2 + 6^2} = 7.5$$

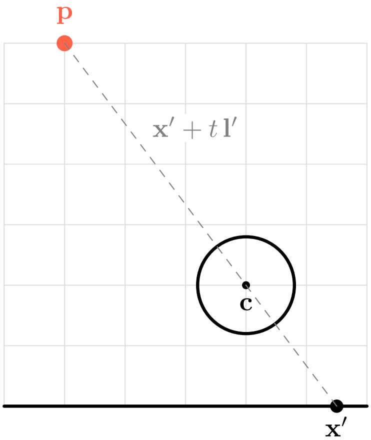
$$\mathbf{l}' = \frac{(-4.5,\ 6)}{7.5} = (-0.6,\ 0.8)$$

Worked: $\mathbf{x}'$ is blocked at $t = 1.7$



Any hit with $t \in [\epsilon, r'] = [\epsilon, 7.5]$?

Worked: $\mathbf{x}'$ is blocked at $t = 1.7$

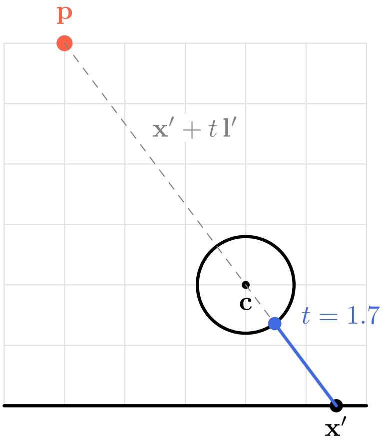


Any hit with $t \in [\epsilon,\, r'] = [\epsilon,\, 7.5]$?

$$\mathbf{c} - \mathbf{x}' = (-1.5,\; 2) = 2.5\,\mathbf{l}'$$

- The ray passes through the centre $\mathbf{c}$, at $t = 2.5$

Worked: $\mathbf{x}'$ is blocked at $t = 1.7$



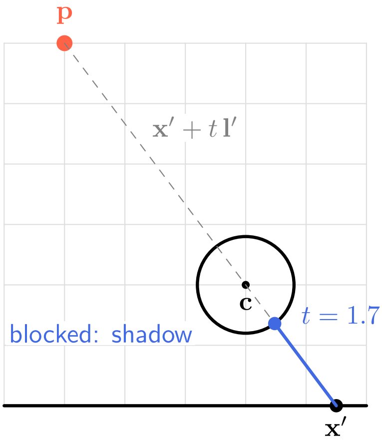
Any hit with $t \in [\epsilon,\, r'] = [\epsilon,\, 7.5]$?

$$\mathbf{c} - \mathbf{x}' = (-1.5,\ 2) = 2.5\,\mathbf{l}'$$

first hit: $t = 2.5 - 0.8 = 1.7$

- The ray passes through the centre $\mathbf{c}$, at $t = 2.5$
- It meets the circle one radius (0.8) earlier

Worked: $\mathbf{x}'$ is blocked at $t = 1.7$



Any hit with $t \in [\epsilon, r'] = [\epsilon, 7.5]$?

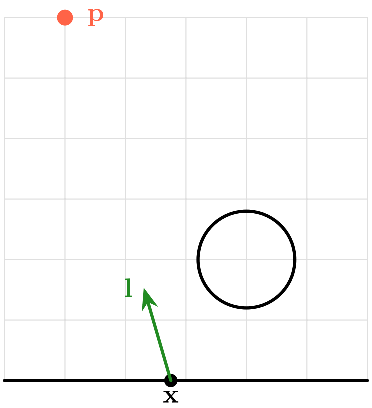
$$\mathbf{c} - \mathbf{x}' = (-1.5,\ 2) = 2.5\,\mathbf{l}'$$

$$\text{first hit: } t = 2.5 - 0.8 = 1.7$$

$$\epsilon \leq 1.7 \leq 7.5 \ \Rightarrow\ \text{shadow}$$

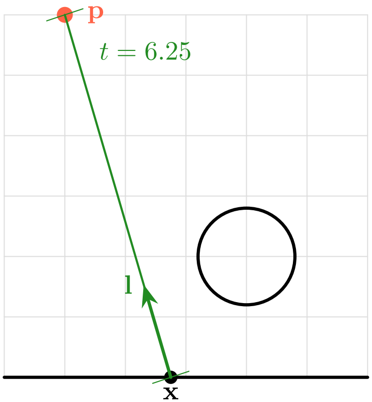
- The ray passes through the centre $\mathbf{c}$, at $t = 2.5$
- It meets the circle one radius (0.8) earlier

Worked: $\mathbf{x}$ finds nothing in $[\epsilon, 6.25]$



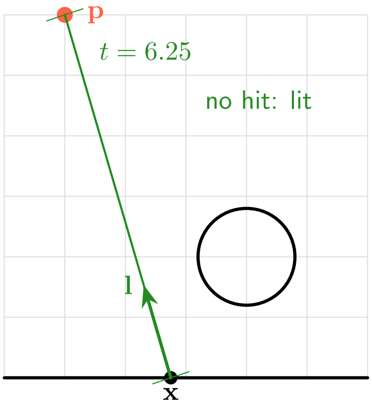
- $\mathbf{l} = (-0.28, 0.96)$, $r = 6.25$ from before

Worked: $\mathbf{x}$ finds nothing in $[\epsilon, 6.25]$



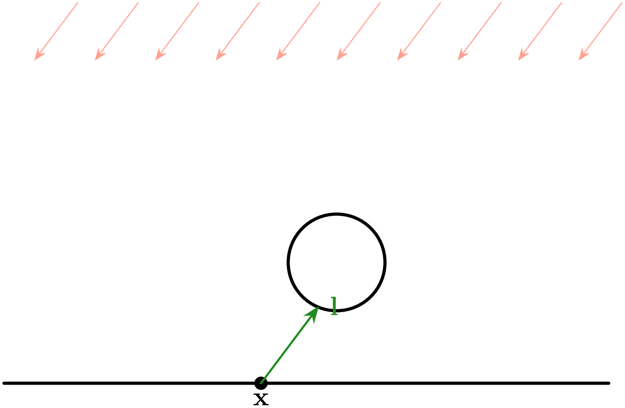
- $\mathbf{l} = (-0.28, 0.96)$, $r = 6.25$ from before
- Walk from ϵ to 6.25: no object

Worked: $\mathbf{x}$ finds nothing in $[\epsilon, 6.25]$



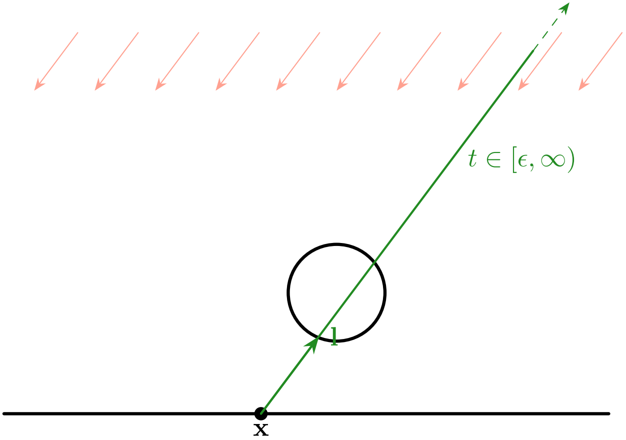
- $\mathbf{l} = (-0.28, 0.96)$, $r = 6.25$ from before
- Walk from ϵ to 6.25: no object
- No hit: add this light's share

Directional light: no far end



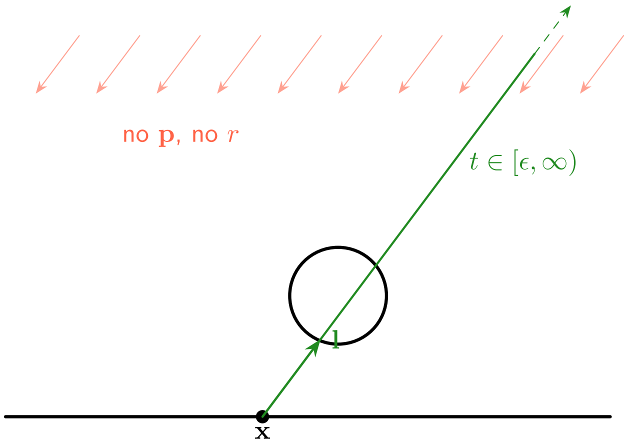
- Light has no position

Directional light: no far end



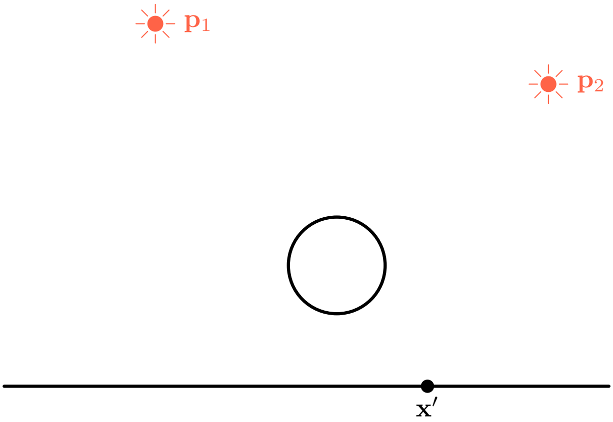
- Light has no position
- Search $[\epsilon, \infty)$ instead of $[\epsilon, r]$

Directional light: no far end



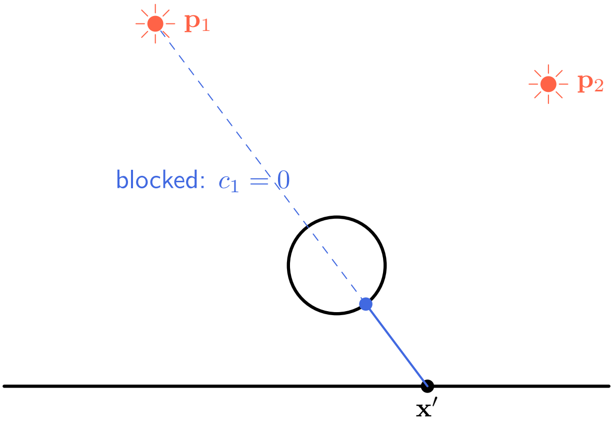
- Light has no position
- Search $[\epsilon, \infty)$ instead of $[\epsilon, r]$
- Anything along $\mathbf{l}$ blocks it

One shadow ray per light



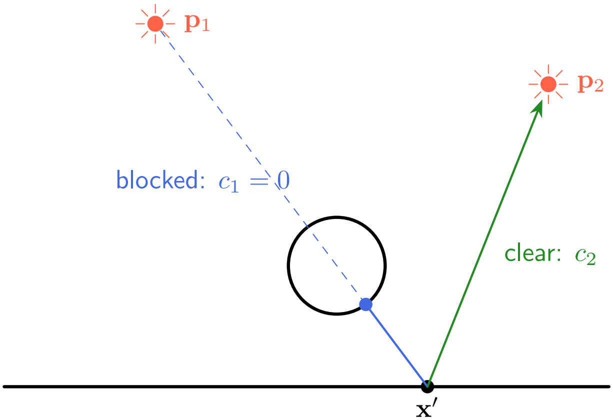
- Each light gets its own test

One shadow ray per light



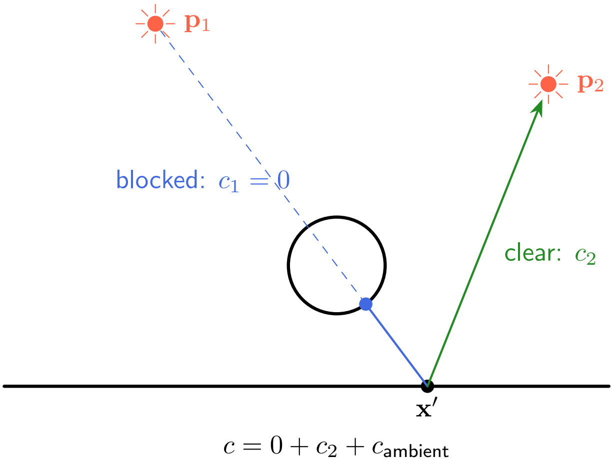
- Each light gets its own test
- Blocked by one

One shadow ray per light



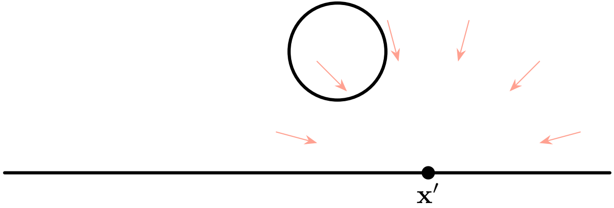
- Each light gets its own test
- Blocked by one
- Lit by the other

One shadow ray per light



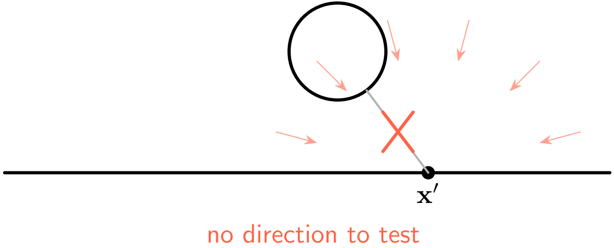
- Each light gets its own test
- Blocked by one
- Lit by the other
- Still add them up

Ambient light casts no shadow ray



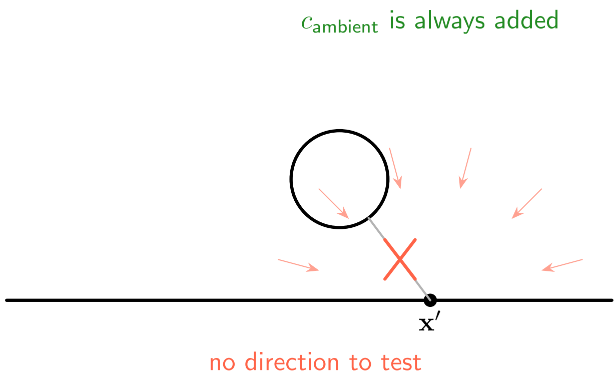
- Light from everywhere

Ambient light casts no shadow ray



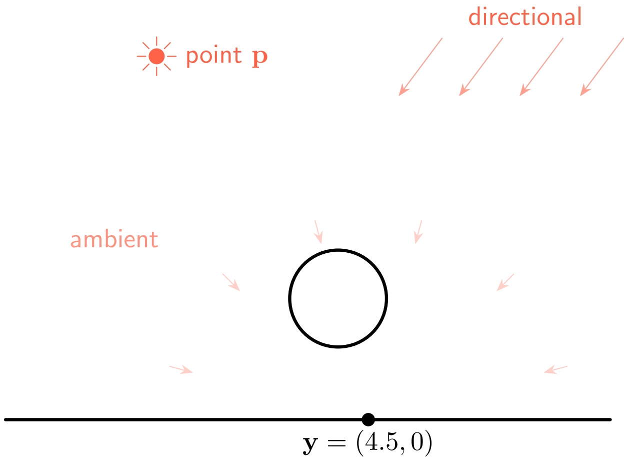
- Light from everywhere
- No $\mathbf{l}$, so no shadow ray

Ambient light casts no shadow ray



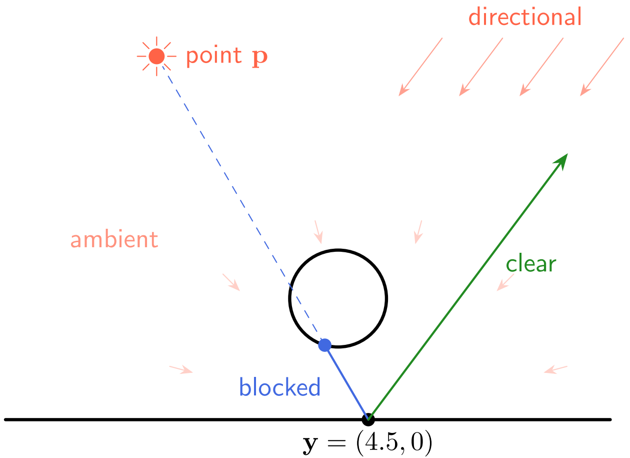
- Light from everywhere
- No $\mathbf{l}$, so no shadow ray
- Even $\mathbf{x}'$ gets its share

Quick check 2: which lights reach $\mathbf{y}$?



- Point, directional, ambient
- Which shadow rays are blocked?

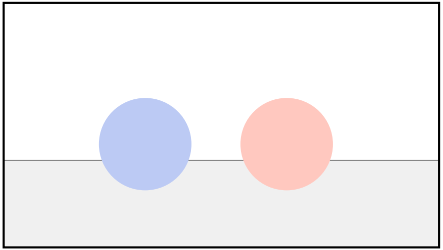
Quick check 2: which lights reach $\mathbf{y}$?



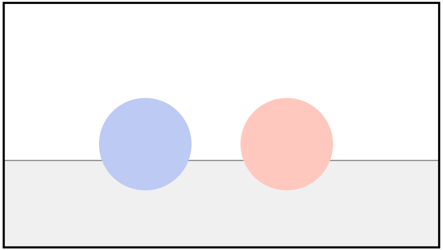
- Point, directional, ambient
- Which shadow rays are blocked?
- Point: blocked; the other two reach $\mathbf{y}$

Shadows show what rests on what

no shadows

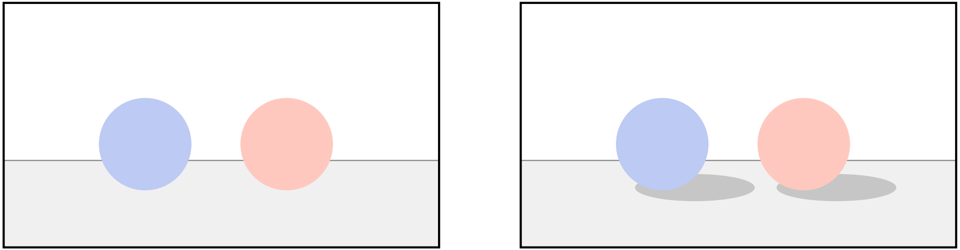


with shadows



Shadows show what rests on what

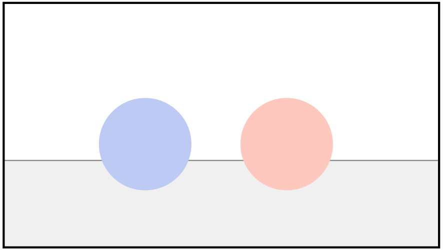
no shadows with shadows



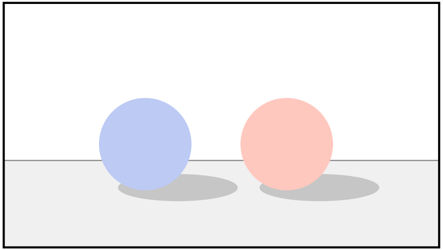
- Same scene, shadows added

Shadows show what rests on what

no shadows

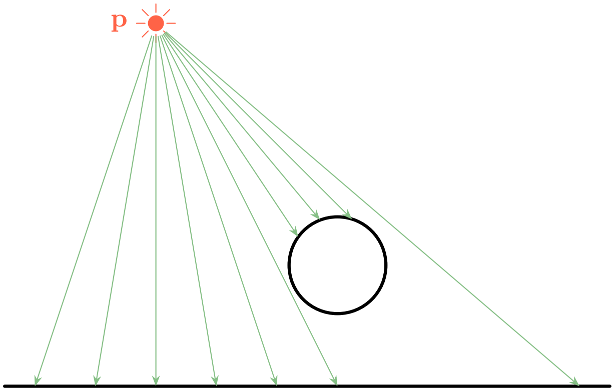


with shadows



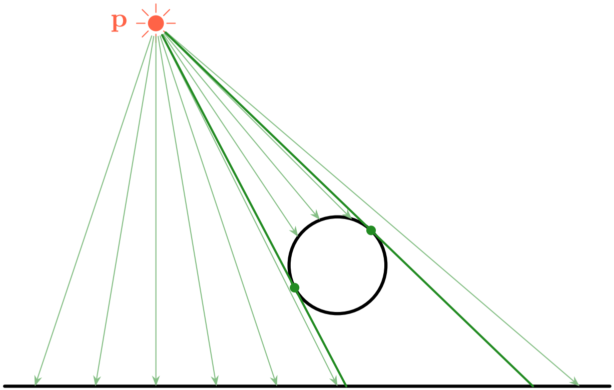
- Same scene, shadows added
- Now the balls sit on the floor

Seen from the light: a fan of rays



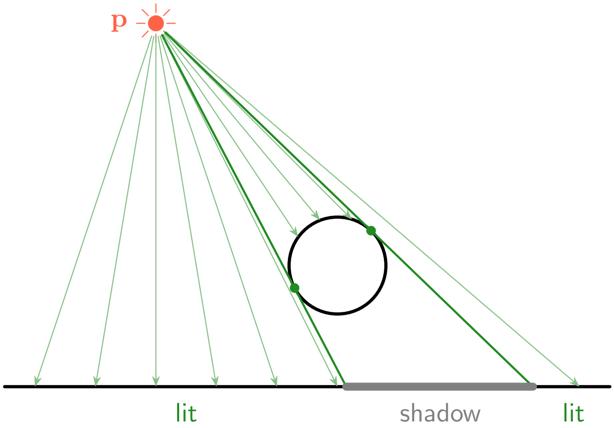
- Reverse each shadow ray: rays leave the light
- Where a ray lands, the floor is lit

Seen from the light: a fan of rays



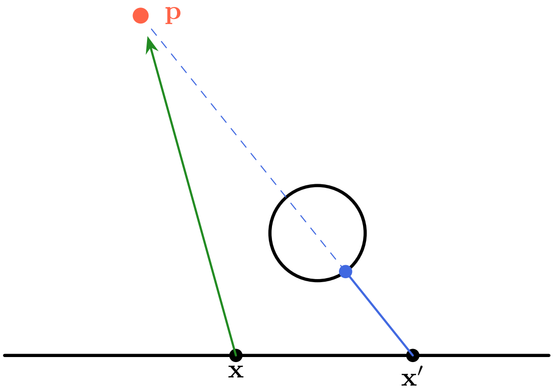
- Reverse each shadow ray: rays leave the light
- Where a ray lands, the floor is lit
- Two rays just graze the object: the shadow's edges

Seen from the light: a fan of rays



- Reverse each shadow ray: rays leave the light
- Where a ray lands, the floor is lit
- Two rays just graze the object: the shadow's edges
- No ray lands between them: shadow

Recap: the shadow test



shadow ray $\mathbf{x} + t\,\mathbf{l}$, searched over $t \in [\epsilon, r]$

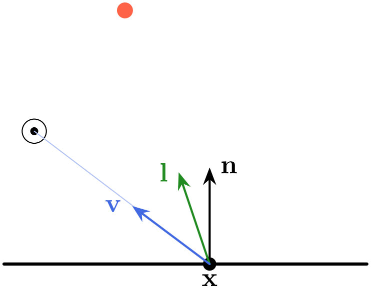
a hit: in shadow, this light adds nothing

no hit: add this light's share

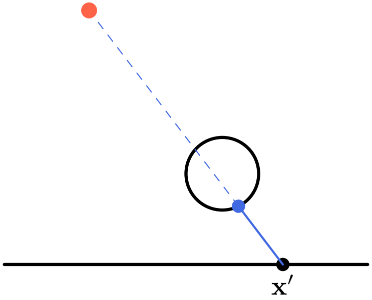
directional light: $r \to \infty$

ambient light: no test

Summary



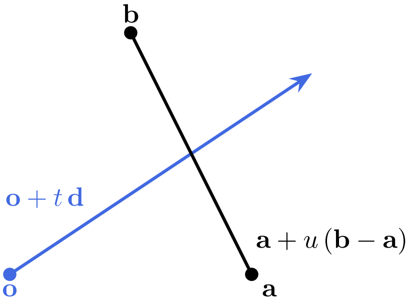
four vectors at the hit



shadow ray: $[\epsilon, r]$, hit means shadow

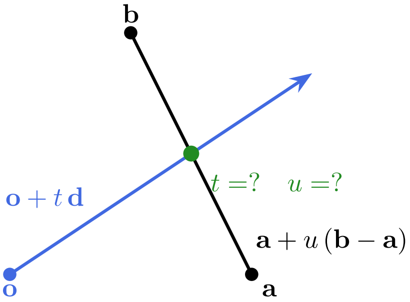
- Three lights: point, directional, ambient
- $\mathbf{x} = \mathbf{e} + t\mathbf{d}$, $\mathbf{n}$ from the surface, $\mathbf{l}$ toward the light, $\mathbf{v} = -\mathbf{d}/\|\mathbf{d}\|$
- One shadow ray per light; start at ϵ; ambient needs none

Next lecture: finding the hit exactly



- Today the hit t came off the picture
- Next time: any ray, any wall

Next lecture: finding the hit exactly



- Today the hit t came off the picture
- Next time: any ray, any wall
- At the crossing both formulas give the same point
- Two equations, two unknowns

Exit Ticket

Today's question *(2 minutes)*

A point light sits at $\mathbf{p} = (6, 5)$. We shade the floor point $\mathbf{x} = (2, 2)$.

(a) Write $\mathbf{l}$ and r.

(b) Over which t does the shadow ray search?

CS 453 — Exit Ticket #11

Name: ______________________

Roll #: ____________ Date: ____________

Answer:

____

____

Hand it to me on your way out.

No name, no attendance.