Extra 4: Light Paths

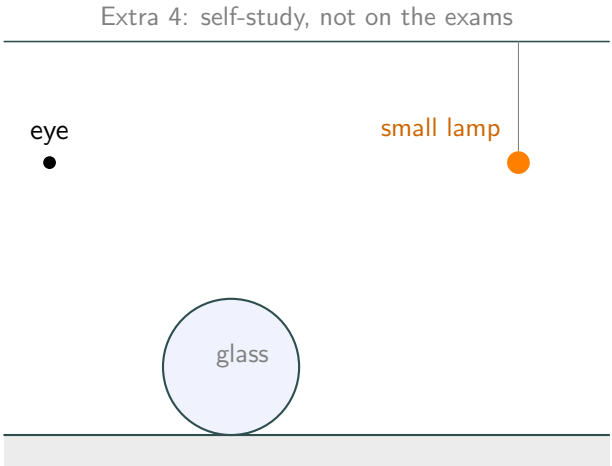
From the Lamp, From the Eye

CS 453 — Computer Graphics

Fakhir Shaheen

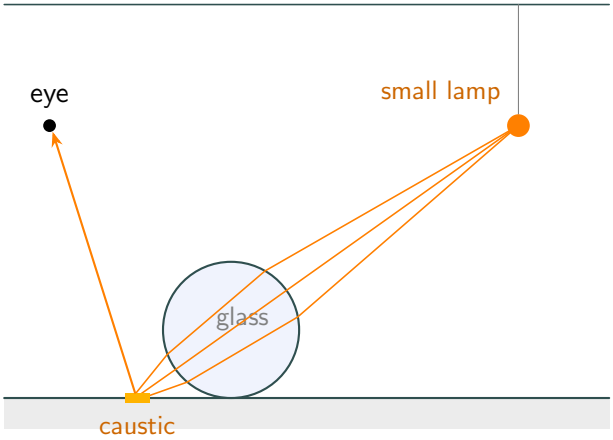
Forman Christian University

Today: which paths can a tracer find?



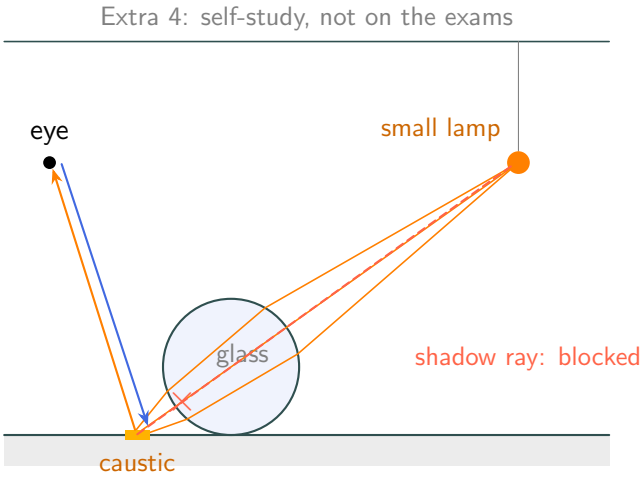
Today: which paths can a tracer find?

Extra 4: self-study, not on the exams



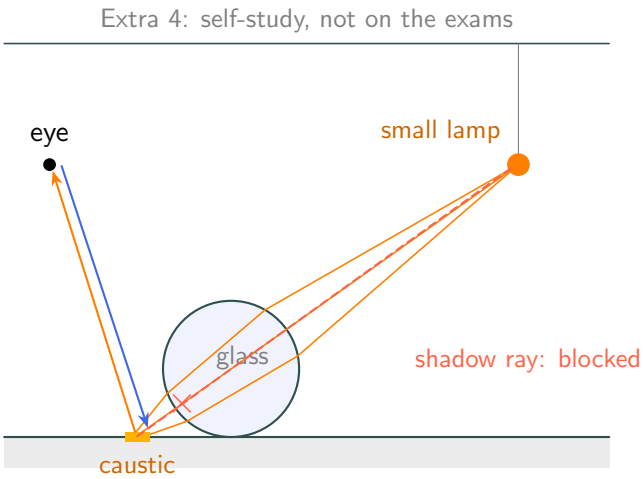
- Light reaches the floor by bending through the glass.

Today: which paths can a tracer find?



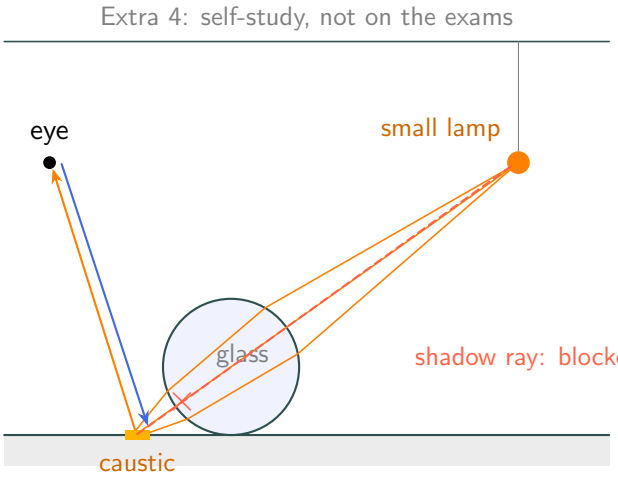
- Light reaches the floor by bending through the glass.
- Our tracer starts at the eye; its shadow ray cannot bend.

Today: which paths can a tracer find?



- Light reaches the floor by bending through the glass.
- Our tracer starts at the eye; its shadow ray cannot bend.
- Some light it finds only by luck; some never.

Today: which paths can a tracer find?

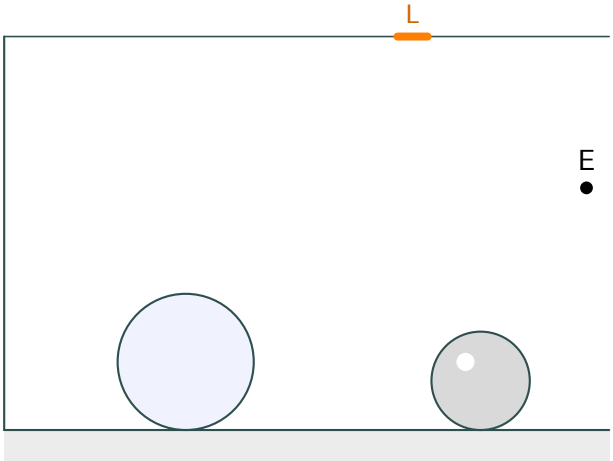


- Light reaches the floor by bending through the glass.
- Our tracer starts at the eye; its shadow ray cannot bend.
- Some light it finds only by luck; some never.
- Today: name the paths, then start from the lamp too.

Naming paths

Four letters, and the strings our tracer misses

Four letters for a path

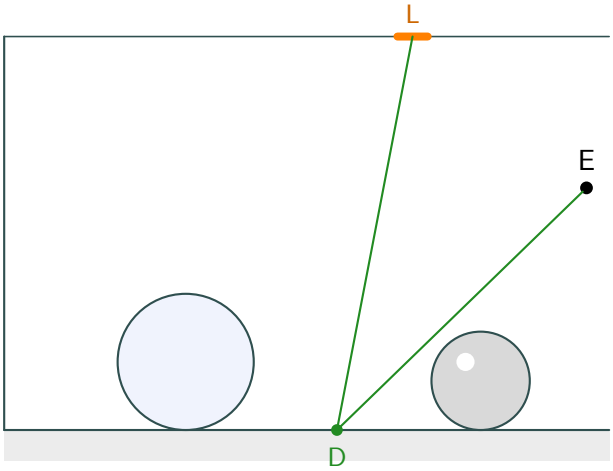


L lamp **E** eye

D chalk: light goes every way

S mirror, glass: one way only

Four letters for a path



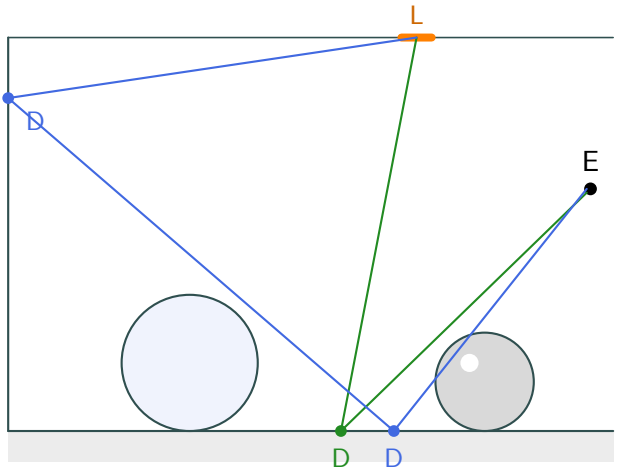
L lamp **E** eye

D chalk: light goes every way

S mirror, glass: one way only

L D E direct light

Four letters for a path



L lamp **E** eye

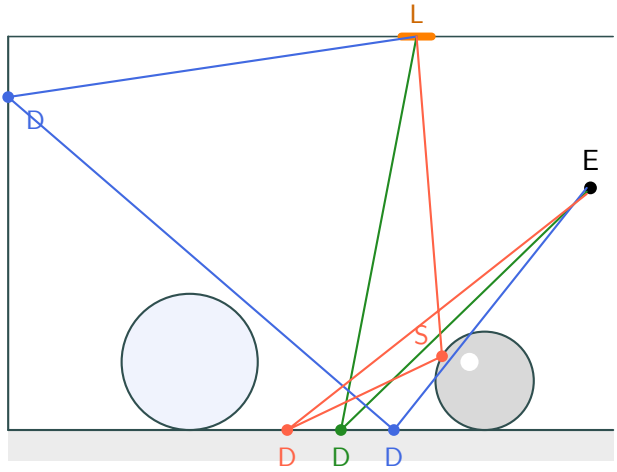
D chalk: light goes every way

S mirror, glass: one way only

L D E direct light

L D D E bounced twice

Four letters for a path



L lamp **E** eye

D chalk: light goes every way

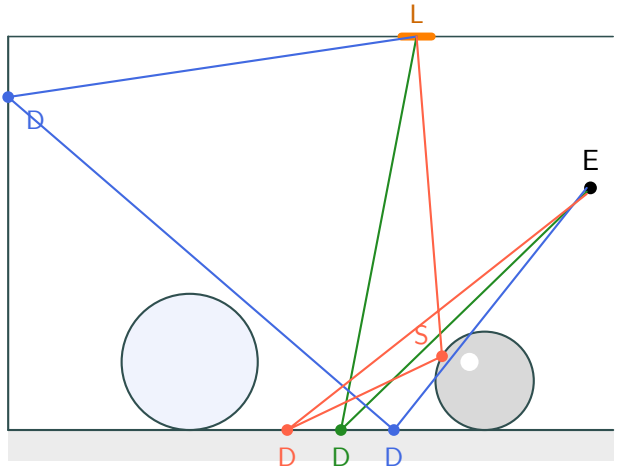
S mirror, glass: one way only

L D E direct light

L D D E bounced twice

L S D E off the mirror first

Four letters for a path



L lamp **E** eye

D chalk: light goes every way

S mirror, glass: one way only

L D E direct light

L D D E bounced twice

L S D E off the mirror first

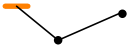
- One letter per vertex, read from the lamp.

Patterns: one string for many paths

D^+ one D or more	D^* none or more	$(S\|D)$ either one

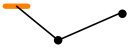
Patterns: one string for many paths

D^+ one D or more	D^* none or more	$(S\|D)$ either one

L D E	direct light: one chalk bounce	

Patterns: one string for many paths

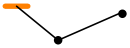
D^+ one D or more	D^* none or more	(S|D) either one

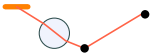
L D E	direct light: one chalk bounce 
L D D^+ E	indirect light: two chalk bounces or more

Patterns: one string for many paths

D^+ one D or more	D^* none or more	$(S\|D)$ either one

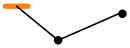
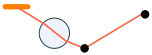
L D E	direct light: one chalk bounce
L D D^+ E	indirect light: two chalk bounces or more
L S^+ D E	a caustic: mirrors or glass, then chalk





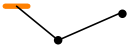
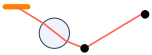
Patterns: one string for many paths

D^+ one D or more	D^* none or more	$(S\|D)$ either one

L D E	direct light: one chalk bounce 
L D D^+ E	indirect light: two chalk bounces or more
L S^+ D E	a caustic: mirrors or glass, then chalk 
L $(S\|D)^*$ E	every path there is

Patterns: one string for many paths

D^+ one D or more	D^* none or more	$(S\|D)$ either one

$L\,D\,E$	direct light: one chalk bounce 
$L\,D\,D^+\,E$	indirect light: two chalk bounces or more
$L\,S^+\,D\,E$	a caustic: mirrors or glass, then chalk 
$L\,(S\|D)^*\,E$	every path there is

- A renderer must find $L\,(S|D)^*\,E$; each method finds a part of it.

Older methods, as patterns

method	finds	misses, for example
Whitted (lectures 17, 18)	$L\,D\,S^*\,E$	bounced light, caustics

Older methods, as patterns

method	**finds**	**misses, for example**
Whitted (lectures 17, 18)	$L\,D\,S^*\,E$	bounced light, caustics
radiosity	$L\,D^*\,E$	every mirror and glass

Older methods, as patterns

method	**finds**	**misses, for example**
Whitted (lectures 17, 18)	$L\,D\,S^*\,E$	bounced light, caustics
radiosity	$L\,D^*\,E$	every mirror and glass
lecture 20, Extra 1	$L\,D\,(S\|D)^*\,E$	nothing, in principle

Older methods, as patterns

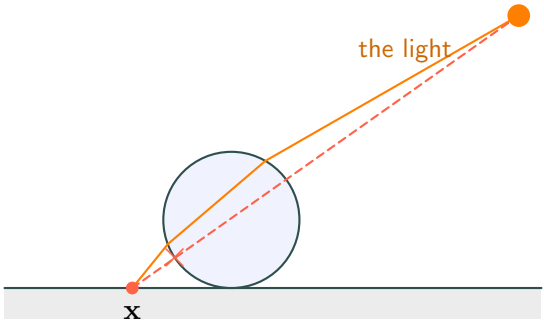
method	**finds**	**misses, for example**
Whitted (lectures 17, 18)	L D S* E	bounced light, caustics
radiosity	L D* E	every mirror and glass
lecture 20, Extra 1	L D (S|D)* E	nothing, in principle
	L S (S|D)* E only by luck	

Older methods, as patterns

method	**finds**	**misses, for example**
Whitted (lectures 17, 18)	L D S* E	bounced light, caustics
radiosity	L D* E	every mirror and glass
lecture 20, Extra 1	L D (S|D)* E	nothing, in principle
	L S (S|D)* E only by luck	

- The weak spot: an S next to the lamp. Then the shadow ray is useless.

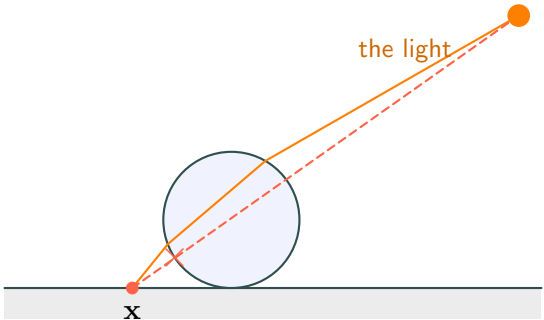
A shadow ray cannot bend



glass: the straight line is blocked

A shadow ray cannot bend

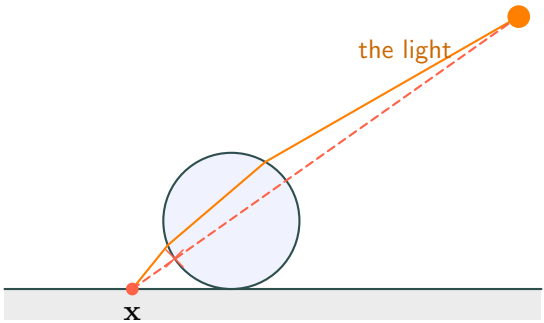
- A join is a straight line: glass blocks it.



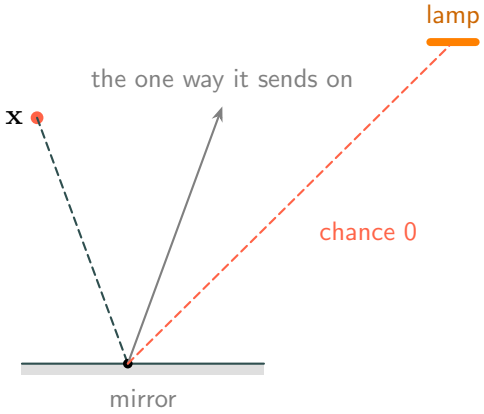
glass: the straight line is blocked

A shadow ray cannot bend

- A join is a straight line: glass blocks it.

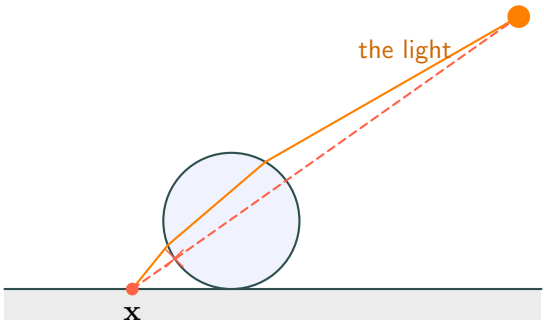


glass: the straight line is blocked



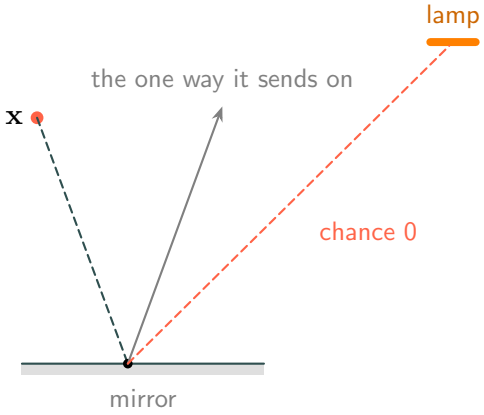
A shadow ray cannot bend

- A join is a straight line: glass blocks it.



glass: the straight line is blocked

- At a spike, a join has chance 0.



Worked: the caustic under the glass ball

Our room: lamp 0.2×0.2, $L_e = 180$; floor $R = 0.75$. At the caustic's brightest point:

shadow ray: blocked every time

lecture 20's bounce: reaches the lamp through the ball 1 time in 21

(on open floor nearby the bare lamp is hit 1 time in 818, but the shadow ray finds it)

Try first: what does one path bring back, on average? Of 320 paths, how many find it?

Worked: the caustic under the glass ball

Our room: lamp 0.2×0.2, $L_e = 180$; floor $R = 0.75$. At the caustic's brightest point:

shadow ray: blocked every time

lecture 20's bounce: reaches the lamp through the ball 1 time in 21

(on open floor nearby the bare lamp is hit 1 time in 818, but the shadow ray finds it)

Try first: what does one path bring back, on average? Of 320 paths, how many find it?

a success brings back $R\,L_e = 0.75 \times 180 = 135$; average $135/21 = 6.4$

Worked: the caustic under the glass ball

Our room: lamp 0.2×0.2, $L_e = 180$; floor $R = 0.75$. At the caustic's brightest point:

shadow ray: blocked every time

lecture 20's bounce: reaches the lamp through the ball 1 time in 21

(on open floor nearby the bare lamp is hit 1 time in 818, but the shadow ray finds it)

Try first: what does one path bring back, on average? Of 320 paths, how many find it?

a success brings back $R\,L_e = 0.75 \times 180 = 135$; average $135/21 = 6.4$

$320/21 \approx 15$ successes; a little off the brightest point 1 in 77: about 4

Worked: the caustic under the glass ball

Our room: lamp 0.2×0.2, $L_e = 180$; floor $R = 0.75$. At the caustic's brightest point:

shadow ray: blocked every time

lecture 20's bounce: reaches the lamp through the ball 1 time in 21

(on open floor nearby the bare lamp is hit 1 time in 818, but the shadow ray finds it)

Try first: what does one path bring back, on average? Of 320 paths, how many find it?

a success brings back $R\,L_e = 0.75 \times 180 = 135$; average $135/21 = 6.4$

$320/21 \approx 15$ successes; a little off the brightest point 1 in 77: about 4

- A few huge values and many zeros: specks. A lamp 10 times smaller: 100 times rarer.

Worked: the caustic under the glass ball

Our room: lamp 0.2×0.2, $L_e = 180$; floor $R = 0.75$. At the caustic's brightest point:

shadow ray: blocked every time

lecture 20's bounce: reaches the lamp through the ball 1 time in 21

(on open floor nearby the bare lamp is hit 1 time in 818, but the shadow ray finds it)

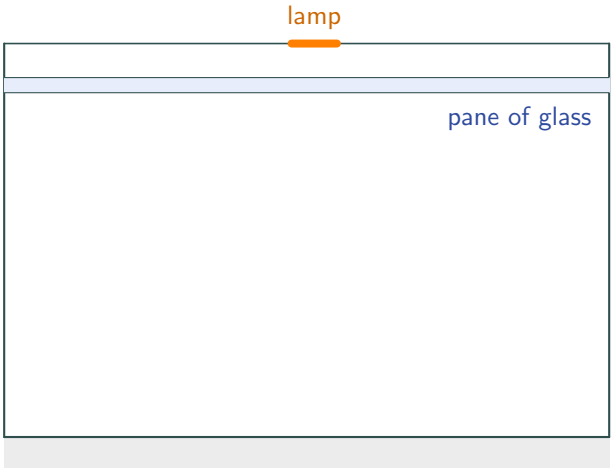
Try first: what does one path bring back, on average? Of 320 paths, how many find it?

a success brings back $R\,L_e = 0.75 \times 180 = 135$; average $135/21 = 6.4$

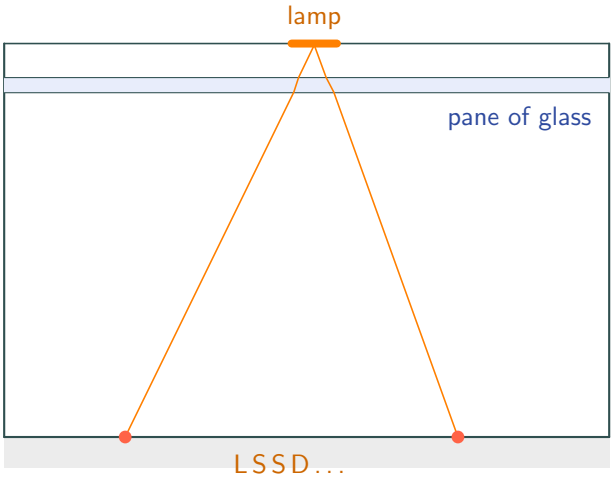
$320/21 \approx 15$ successes; a little off the brightest point 1 in 77: about 4

- A few huge values and many zeros: specks. A lamp 10 times smaller: 100 times rarer.
- A point lamp: chance 0. The caustic is never found.

A pane of glass under the lamp

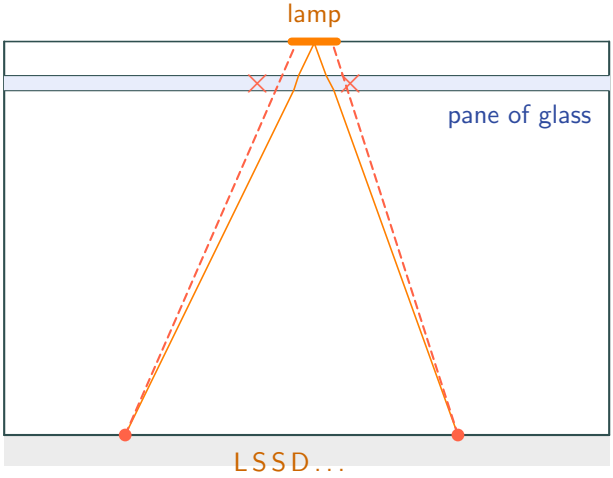


A pane of glass under the lamp



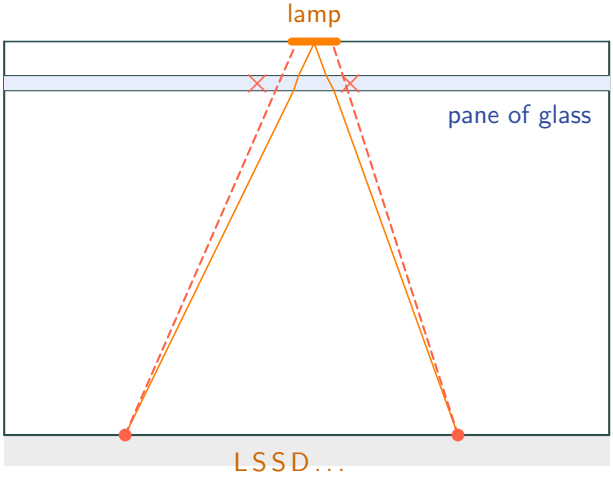
- Every lit point is now a caustic: LSSD...

A pane of glass under the lamp



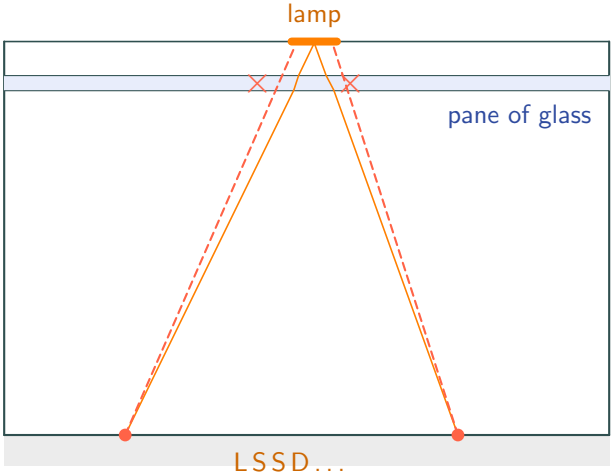
- Every lit point is now a caustic: LSSD...
- Every shadow ray is blocked.

A pane of glass under the lamp



- Every lit point is now a caustic: LSSD...
- Every shadow ray is blocked.
- Only the random bounce finds the lamp: Extra 1's way 1, alone.

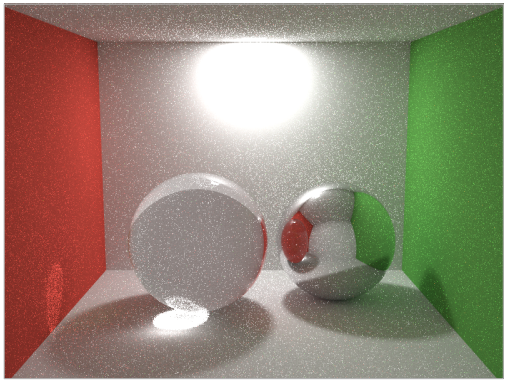
A pane of glass under the lamp



- Every lit point is now a caustic: L S S D . . .
- Every shadow ray is blocked.
- Only the random bounce finds the lamp: Extra 1's way 1, alone.
- Production's cheat: let shadow rays through thin glass.

Rendered: Extra 1's tracer in two rooms

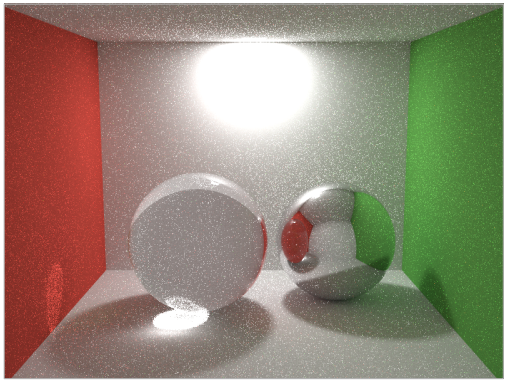
Extra 1's tracer: shadow rays and bounces, balance heuristic; 320 paths per pixel



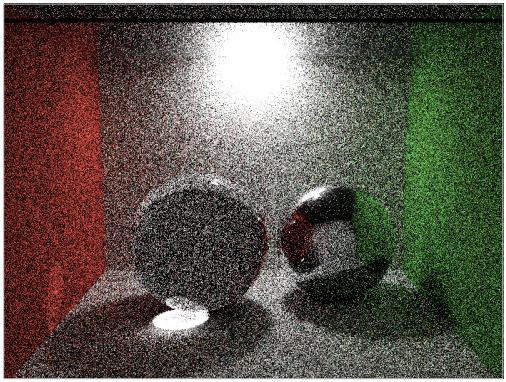
the glass ball and the mirror ball

Rendered: Extra 1's tracer in two rooms

Extra 1's tracer: shadow rays and bounces, balance heuristic; 320 paths per pixel



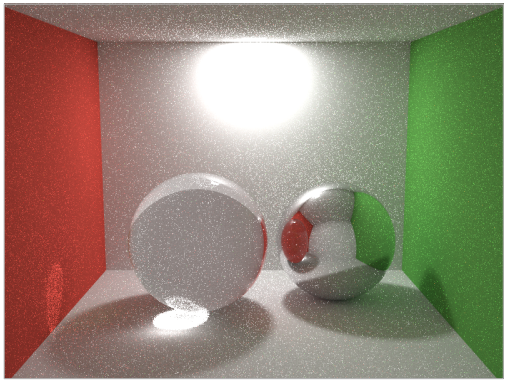
the glass ball and the mirror ball



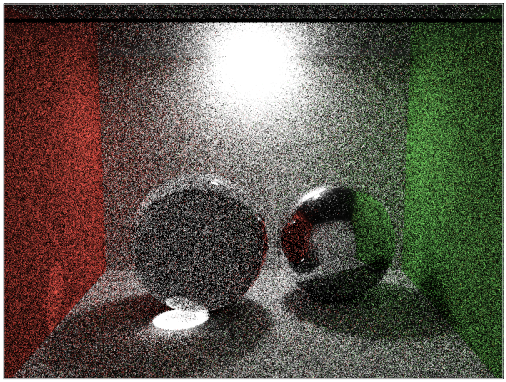
the same, with the pane under the lamp

Rendered: Extra 1's tracer in two rooms

Extra 1's tracer: shadow rays and bounces, balance heuristic; 320 paths per pixel



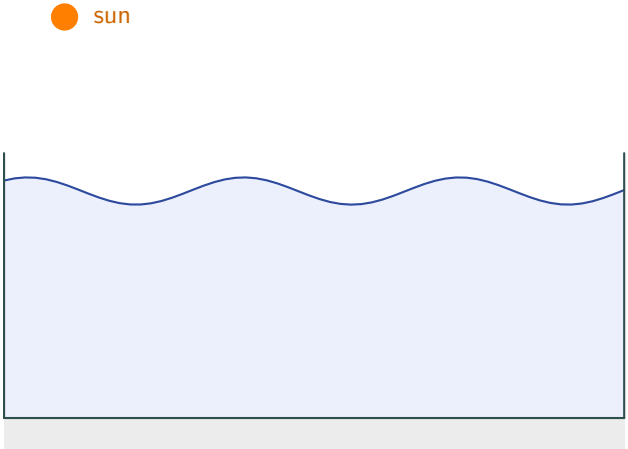
the glass ball and the mirror ball



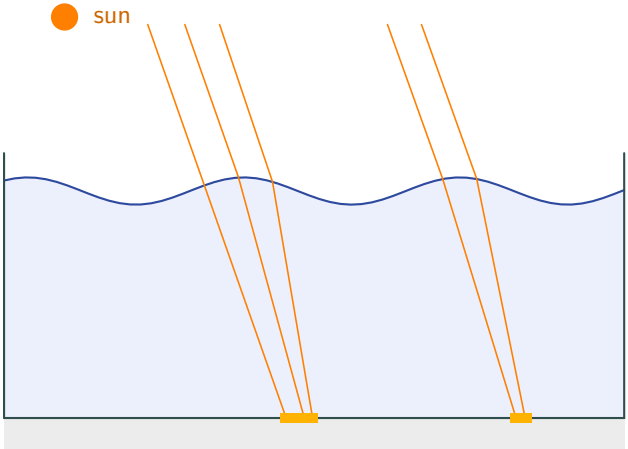
the same, with the pane under the lamp

- White specks: L S D paths found by luck, all over the room.

The pool floor: L S D S E

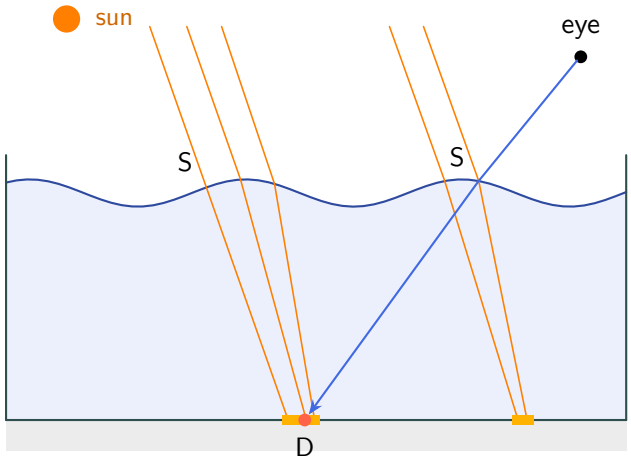


The pool floor: L S D S E



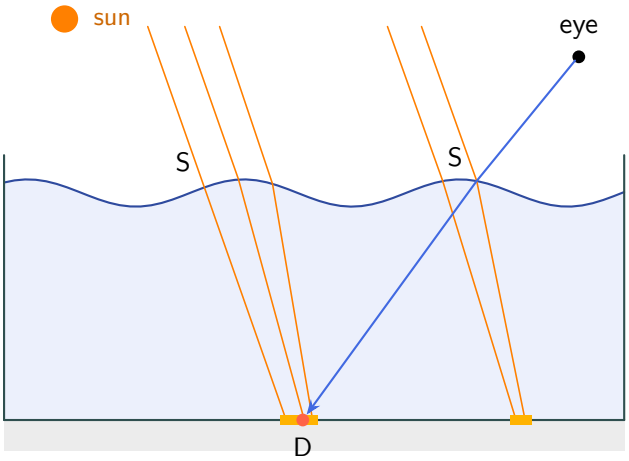
- The waves focus the sun into bright lines: L S D.

The pool floor: L S D S E



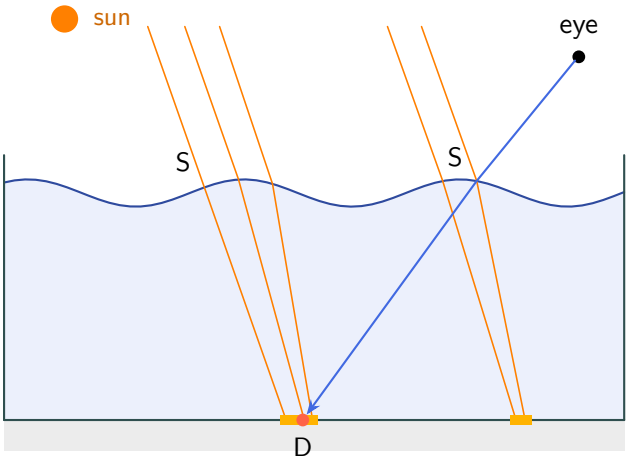
- The waves focus the sun into bright lines: L S D.
- We see the floor through the water: D S E.

The pool floor: L S D S E



- The waves focus the sun into bright lines: L S D.
- We see the floor through the water: D S E.
- L S D S E: a spike on both sides of the chalk.

The pool floor: L S D S E



- The waves focus the sun into bright lines: L S D.
- We see the floor through the water: D S E.
- L S D S E: a spike on both sides of the chalk.
- No straight join is possible anywhere on it.

Recap: who finds which paths

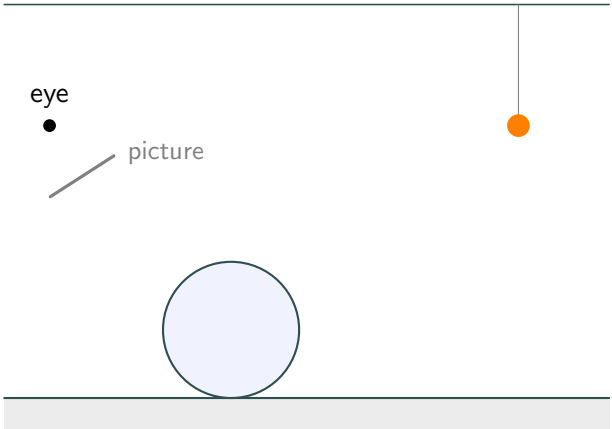
path	for example	Extra 1's tracer
$L\,D\,E$	direct light	✓
$L\,D\,D^{+}\,E$	light bounced off walls	✓
$L\,S^{+}\,D\,E$	caustic: under the glass ball	∼
$L\,S^{+}\,D\,S^{+}\,E$	the pool's floor, through water	×

✓ clean ∼ noisy: found by luck × never, with a point lamp and a pinhole eye

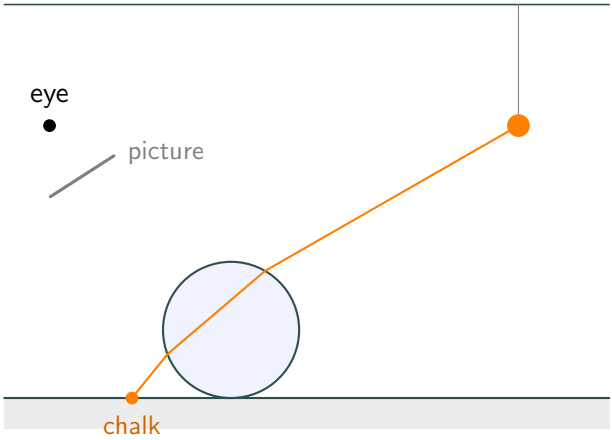
Paths from both ends

Start at the lamp, start at the eye, join them

Light tracing: start at the lamp

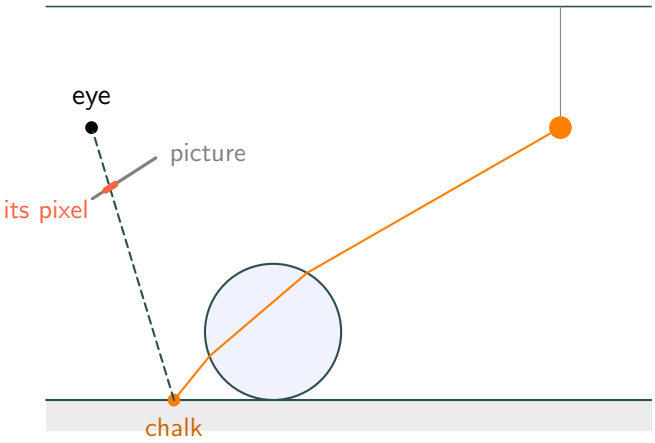


Light tracing: start at the lamp



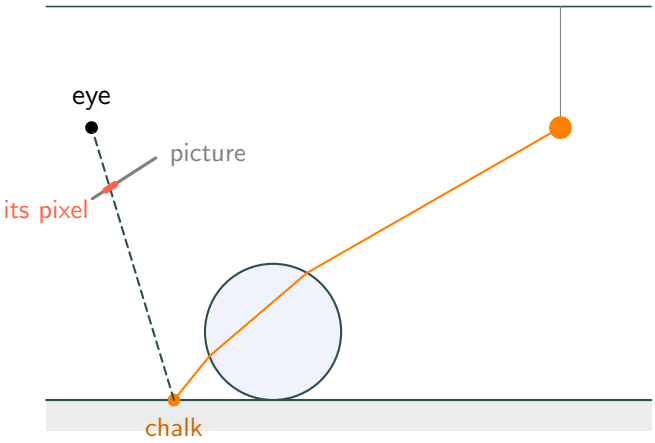
- Shoot a path from the lamp: bounce, bend, as lecture 20 does.

Light tracing: start at the lamp



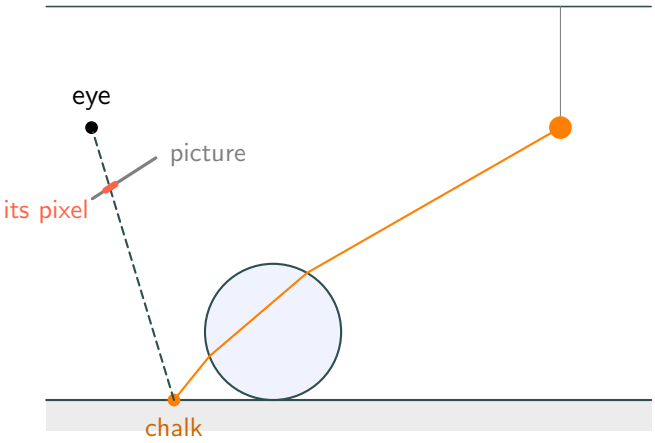
- Shoot a path from the lamp: bounce, bend, as lecture 20 does.
- At each chalk point, join it to the eye; add to the pixel it crosses.

Light tracing: start at the lamp



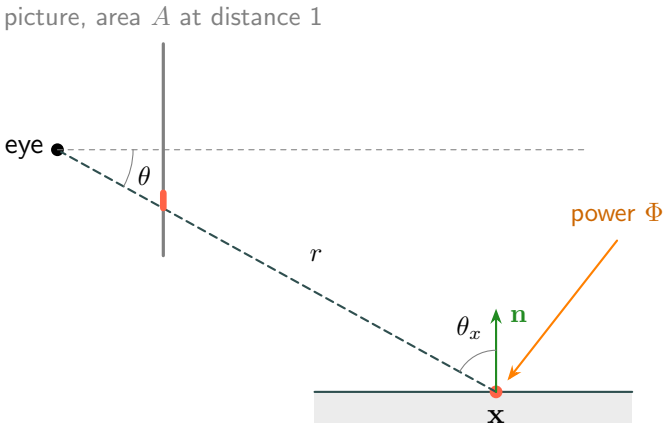
- Shoot a path from the lamp: bounce, bend, as lecture 20 does.
- At each chalk point, join it to the eye; add to the pixel it crosses.
- The caustic is easy now: L S S D, then a join D to E.

Light tracing: start at the lamp

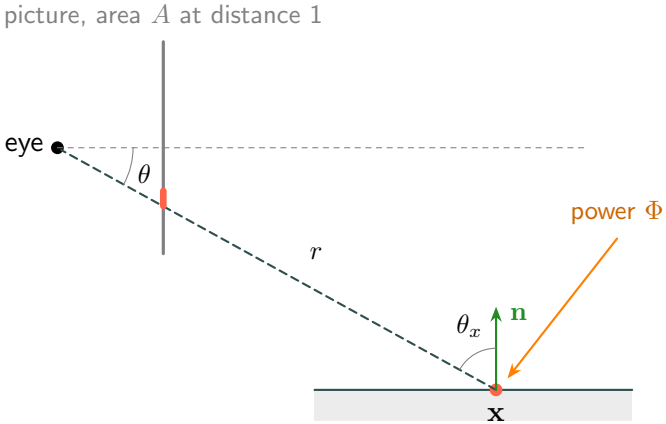


- Shoot a path from the lamp: bounce, bend, as lecture 20 does.
- At each chalk point, join it to the eye; add to the pixel it crosses.
- The caustic is easy now: L S S D, then a join D to E.
- But a mirror the eye looks at (E S. . .) is never joined.

What one light path adds to its pixel

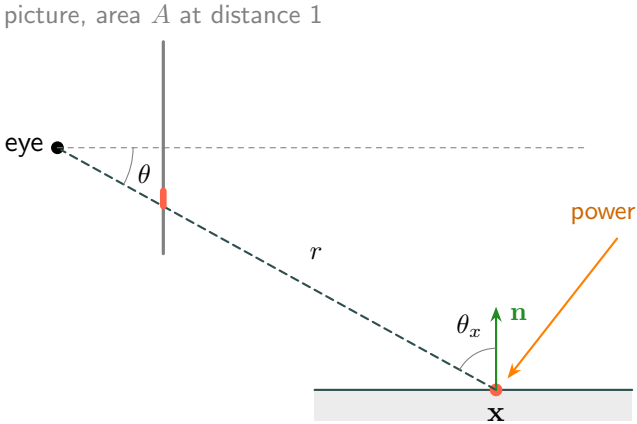


What one light path adds to its pixel



adds $\dfrac{\Phi \cdot \dfrac{R}{\pi} \cdot \cos\theta_x}{r^2} \cdot \dfrac{1}{A\cos^3\theta}$

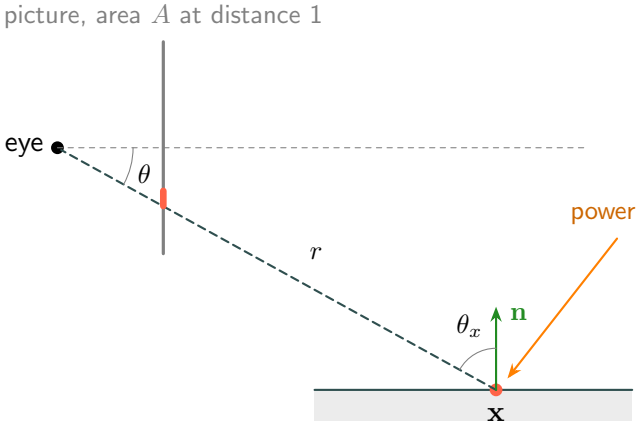
What one light path adds to its pixel



adds $\dfrac{\Phi \cdot \dfrac{R}{\pi} \cdot \cos\theta_x}{r^2} \cdot \dfrac{1}{A\cos^3\theta}$

- Φ, R/π, $\cos\theta_x/r^2$: the light $\mathbf{x}$ sends into the pinhole.

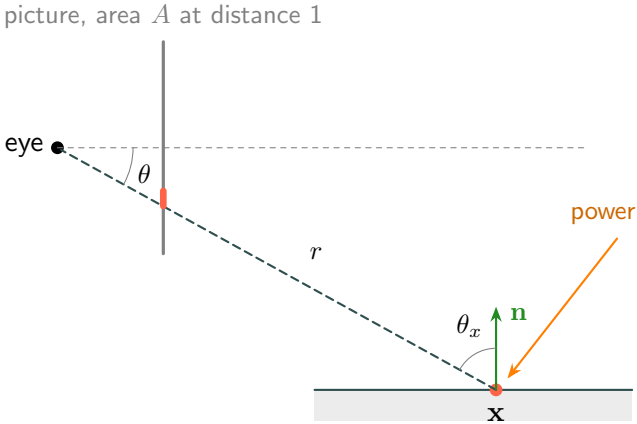
What one light path adds to its pixel



adds $$\frac{\Phi \cdot \dfrac{R}{\pi} \cdot \cos\theta_x}{r^2} \cdot \frac{1}{A\cos^3\theta}$$

- Φ, R/π, $\cos\theta_x/r^2$: the light $\mathbf{x}$ sends into the pinhole.
- $1/(A\cos^3\theta)$: the camera's density, Extra 3's diamond pass.

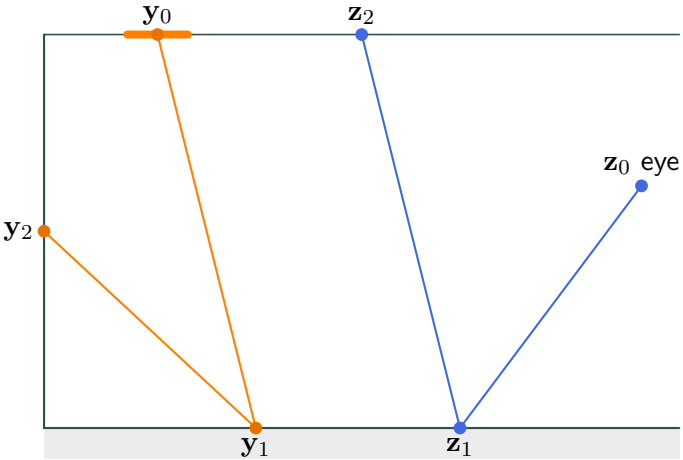
What one light path adds to its pixel



adds $$\frac{\Phi \cdot \frac{R}{\pi} \cdot \cos\theta_x}{r^2} \cdot \frac{1}{A\cos^3\theta}$$

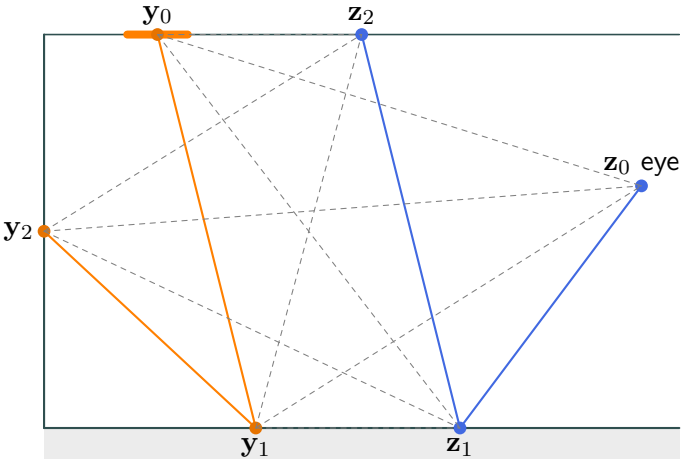
- Φ, R/π, $\cos\theta_x/r^2$: the light $\mathbf{x}$ sends into the pinhole.
- $1/(A\cos^3\theta)$: the camera's density, Extra 3's diamond pass.
- At the end, divide by the paths per pixel.

Two half-paths, joined every way



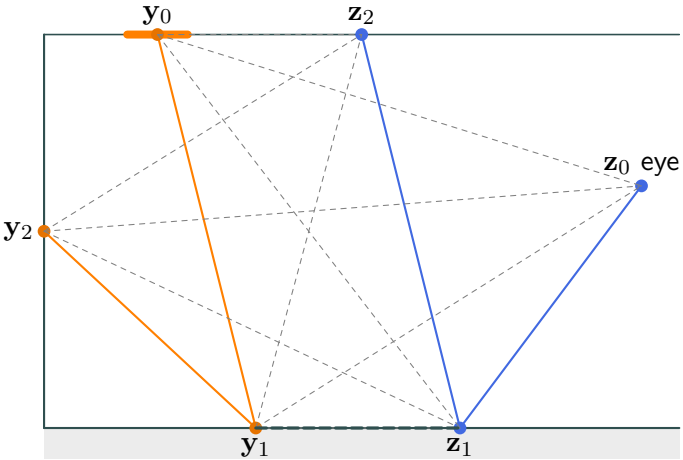
- A light subpath from the lamp, an eye subpath from the eye.

Two half-paths, joined every way



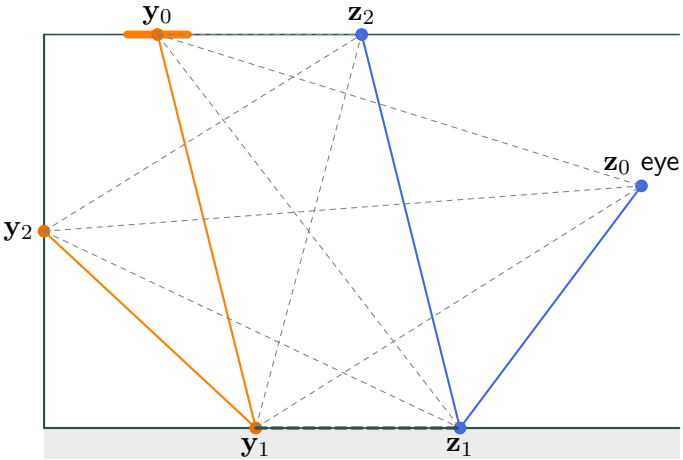
- A light subpath from the lamp, an eye subpath from the eye.
- Join every light vertex to every eye vertex: one shadow ray each.

Two half-paths, joined every way



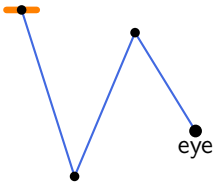
- A light subpath from the lamp, an eye subpath from the eye.
- Join every light vertex to every eye vertex: one shadow ray each.
- Each join is a whole path: $\mathbf{y}_0\,\mathbf{y}_1\,\mathbf{z}_1\,\mathbf{z}_0$ is L D D E.

Two half-paths, joined every way



- A light subpath from the lamp, an eye subpath from the eye.
- Join every light vertex to every eye vertex: one shadow ray each.
- Each join is a whole path: $\mathbf{y}_0\,\mathbf{y}_1\,\mathbf{z}_1\,\mathbf{z}_0$ is L D D E.
- Nine paths, of several lengths, from two half-paths.

One path, $k + 2$ ways to make it

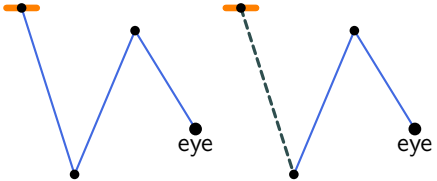


$s = 0, \ \ t = 4$

lecture 20:
it hits the lamp

- k edges, $k + 1$ vertices: s made from the lamp, $t = k + 1 - s$ from the eye.

One path, $k + 2$ ways to make it



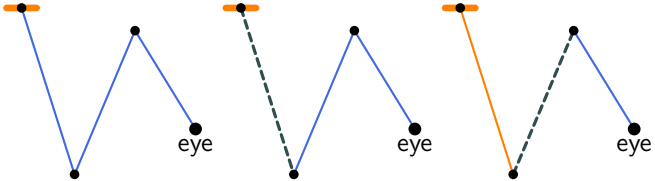
$s = 0,\ t = 4$ $s = 1,\ t = 3$

lecture 20: it hits the lamp

Extra 1's shadow ray

- k edges, $k + 1$ vertices: s made from the lamp, $t = k + 1 - s$ from the eye.

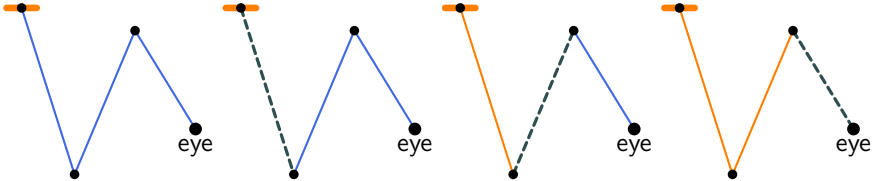
One path, $k + 2$ ways to make it



$s = 0,\ t = 4$	$s = 1,\ t = 3$	$s = 2,\ t = 2$
lecture 20: it hits the lamp	Extra 1's shadow ray	new: a join in the middle

- k edges, $k + 1$ vertices: s made from the lamp, $t = k + 1 - s$ from the eye.

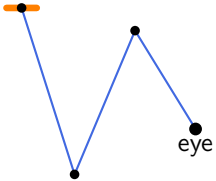
One path, $k + 2$ ways to make it



$s = 0,\ t = 4$	$s = 1,\ t = 3$	$s = 2,\ t = 2$	$s = 3,\ t = 1$
lecture 20: it hits the lamp	Extra 1's shadow ray	new: a join in the middle	light tracing (C10)

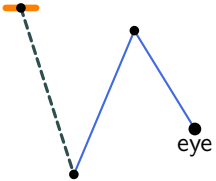
- k edges, $k + 1$ vertices: s made from the lamp, $t = k + 1 - s$ from the eye.

One path, $k + 2$ ways to make it



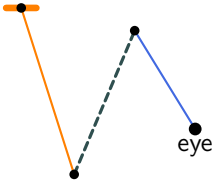
$s = 0,\ t = 4$

lecture 20:
it hits the lamp



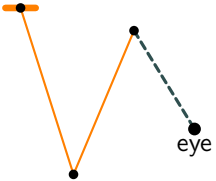
$s = 1,\ t = 3$

Extra 1's
shadow ray



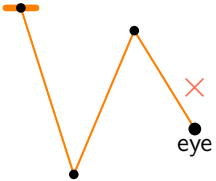
$s = 2,\ t = 2$

new: a join
in the middle



$s = 3,\ t = 1$

light tracing
(C10)

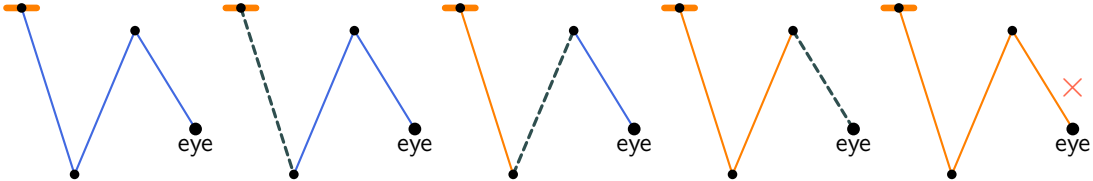


$s = 4,\ t = 0$

the light path
hits the eye

- k edges, $k + 1$ vertices: s made from the lamp, $t = k + 1 - s$ from the eye.

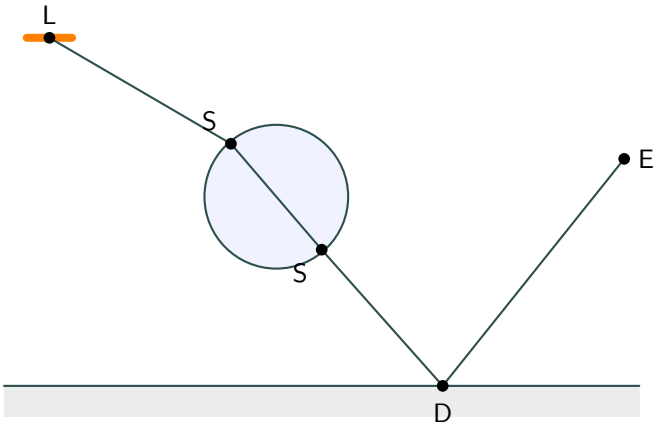
One path, $k + 2$ ways to make it



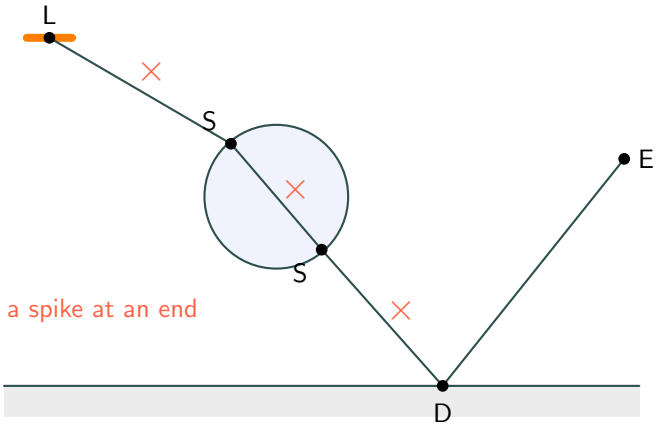
$s = 0,\ t = 4$	$s = 1,\ t = 3$	$s = 2,\ t = 2$	$s = 3,\ t = 1$	$s = 4,\ t = 0$
lecture 20: it hits the lamp	Extra 1's shadow ray	new: a join in the middle	light tracing (C10)	the light path hits the eye

- k edges, $k + 1$ vertices: s made from the lamp, $t = k + 1 - s$ from the eye.
- $s = 0, 1, \ldots, k + 1$: $k + 2$ ways. A pinhole eye is never hit: $k + 1$ are left.

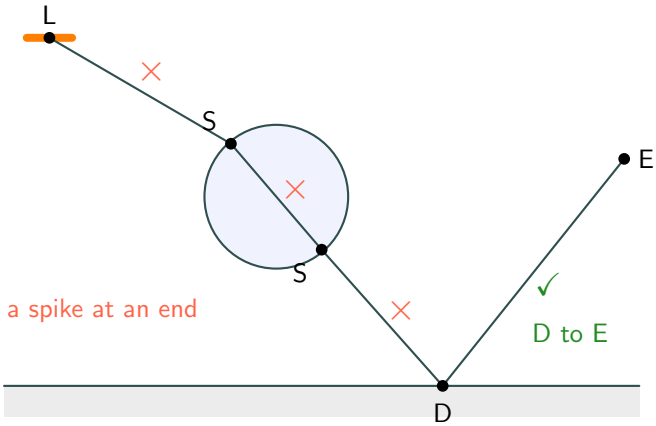
A join that touches a spike counts nothing



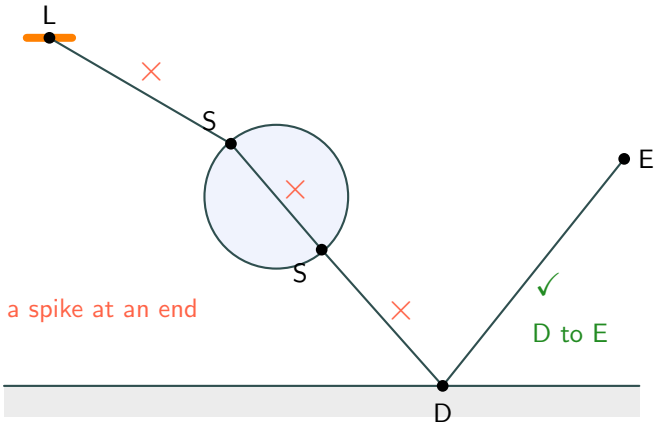
A join that touches a spike counts nothing



A join that touches a spike counts nothing



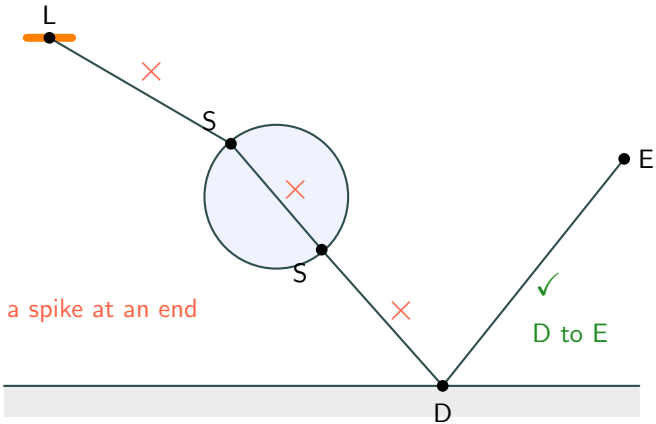
A join that touches a spike counts nothing



$k = 4$: six ways

$s = 0$: bounce hits lamp	luck
$s = 1, 2, 3$: spike joins	0
$s = 4$: light tracing	easy
$s = 5$: hit the pinhole	0

A join that touches a spike counts nothing

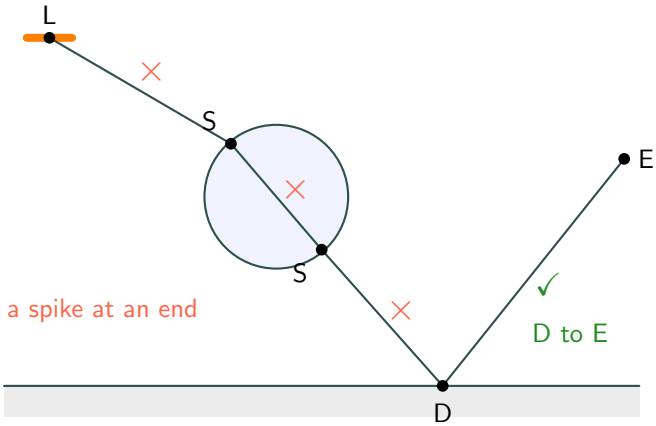


$k = 4$: six ways

$s = 0$: bounce hits lamp	luck
$s = 1, 2, 3$: spike joins	0
$s = 4$: light tracing	easy
$s = 5$: hit the pinhole	0

- A join needs chalk, the lamp or the eye at both ends.

A join that touches a spike counts nothing

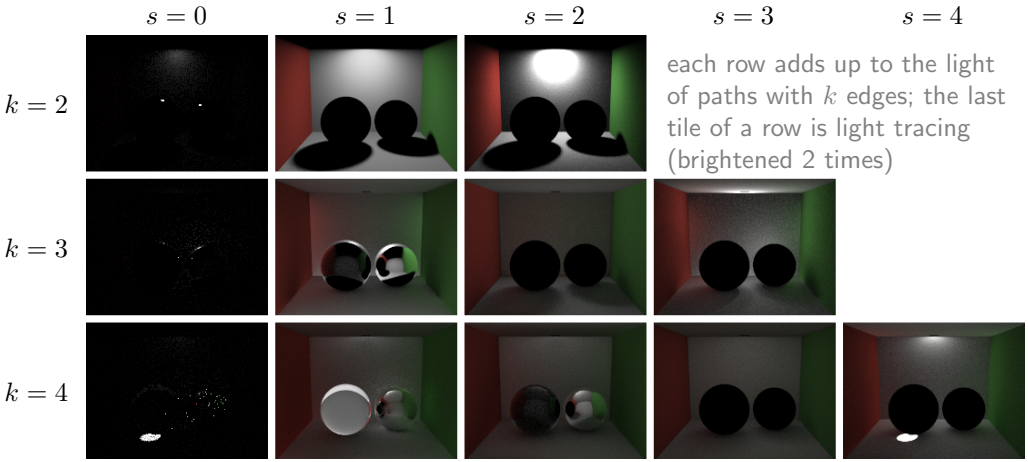


$k = 4$: six ways

$s = 0$: bounce hits lamp	luck
$s = 1, 2, 3$: spike joins	0
$s = 4$: light tracing	easy
$s = 5$: hit the pinhole	0

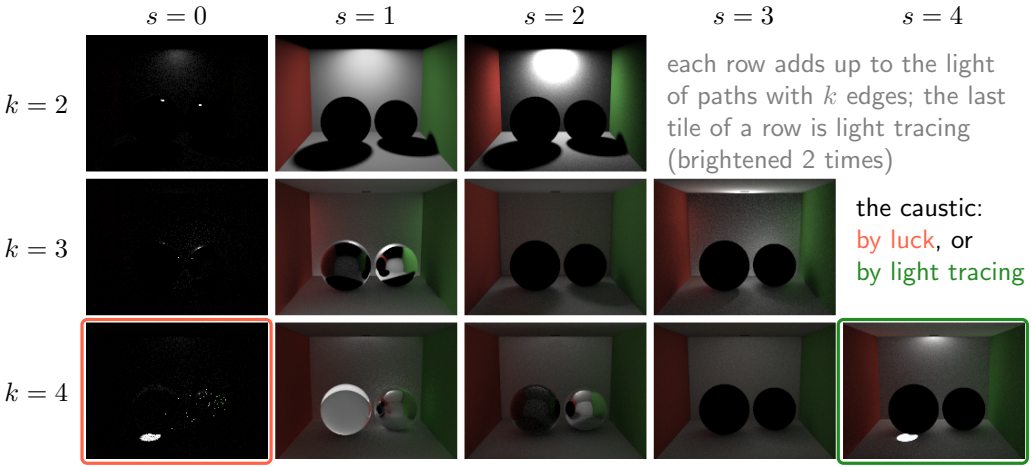
- A join needs chalk, the lamp or the eye at both ends.
- The caustic: light tracing makes it, and wins its weight.

Rendered: each way's share



each row adds up to the light of paths with k edges; the last tile of a row is light tracing (brightened 2 times)

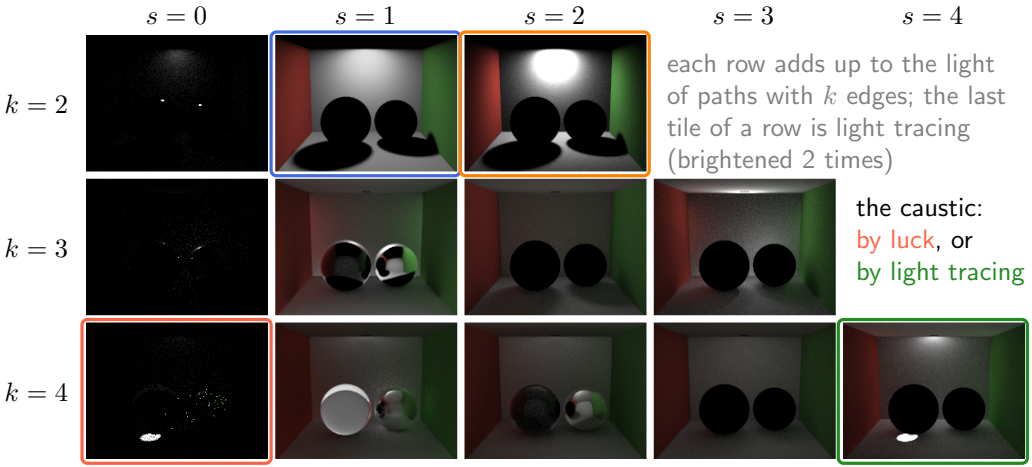
Rendered: each way's share



each row adds up to the light of paths with k edges; the last tile of a row is light tracing (brightened 2 times)

the caustic:
by luck, or
by light tracing

Rendered: each way's share



each row adds up to the light of paths with k edges; the last tile of a row is light tracing (brightened 2 times)

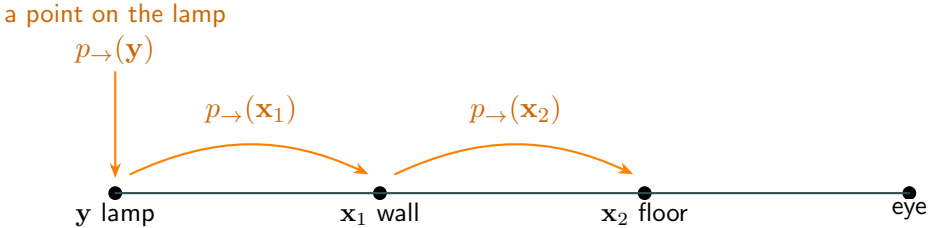
the caustic: by luck, or by light tracing

direct light: shadow rays far from the lamp, light tracing near it

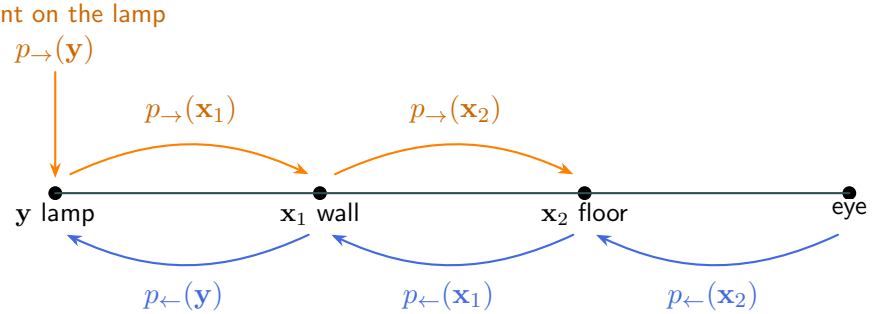
Two densities at every vertex



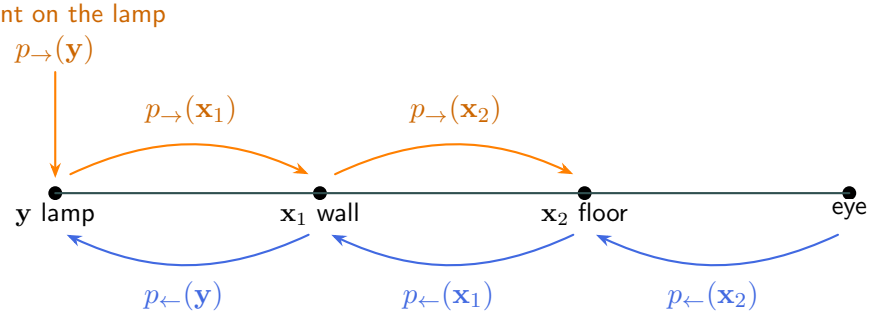
Two densities at every vertex



Two densities at every vertex

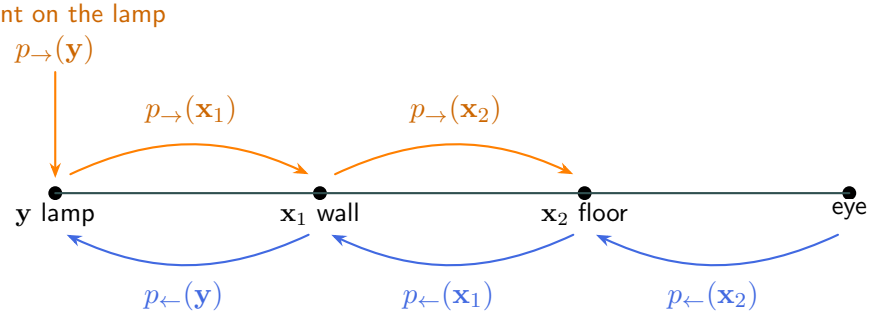


Two densities at every vertex



way s: $p_s =$ the first s of the $p_{\rightarrow}$'s $\times$ the rest of the $p_{\leftarrow}$'s, e.g. $p_1 = p_{\rightarrow}(\mathbf{y})\, p_{\leftarrow}(\mathbf{x}_1)\, p_{\leftarrow}(\mathbf{x}_2)$

Two densities at every vertex



way s: $p_s =$ the first s of the $p_{\rightarrow}$'s $\times$ the rest of the $p_{\leftarrow}$'s, e.g. $p_1 = p_{\rightarrow}(\mathbf{y})\, p_{\leftarrow}(\mathbf{x}_1)\, p_{\leftarrow}(\mathbf{x}_2)$

weight of way s: $\dfrac{p_s}{p_0 + p_1 + \ldots}$ (Extra 1's balance heuristic); every density per unit area

Worked: weighing four ways

Our room: lamp centre, the green wall, the floor, the eye. Per unit area:

$\mathbf{y}$: $p_{\rightarrow} = 1/A = 25$, $p_{\leftarrow} = 0.031$ $\mathbf{x}_1$: $p_{\rightarrow} = 0.031$, $p_{\leftarrow} = 0.049$ $\mathbf{x}_2$: $p_{\rightarrow} = 0.049$, $p_{\leftarrow} = 0.073$

Try first: p_0 to p_3, and the four weights. Which way wins?

Worked: weighing four ways

Our room: lamp centre, the green wall, the floor, the eye. Per unit area:

$\mathbf{y}$: $p_{\rightarrow} = 1/A = 25$, $p_{\leftarrow} = 0.031$ $\mathbf{x}_1$: $p_{\rightarrow} = 0.031$, $p_{\leftarrow} = 0.049$ $\mathbf{x}_2$: $p_{\rightarrow} = 0.049$, $p_{\leftarrow} = 0.073$

Try first: p_0 to p_3, and the four weights. Which way wins?

$$p_0 = 0.031 \times 0.049 \times 0.073 = 0.00011$$

$$p_1 = 25 \times 0.049 \times 0.073 = 0.089$$

$$p_2 = 25 \times 0.031 \times 0.073 = 0.057$$

$$p_3 = 25 \times 0.031 \times 0.049 = 0.038$$

Worked: weighing four ways

Our room: lamp centre, the green wall, the floor, the eye. Per unit area:

$\mathbf{y}$: $p_{\rightarrow} = 1/A = 25$, $p_{\leftarrow} = 0.031$ $\mathbf{x}_1$: $p_{\rightarrow} = 0.031$, $p_{\leftarrow} = 0.049$ $\mathbf{x}_2$: $p_{\rightarrow} = 0.049$, $p_{\leftarrow} = 0.073$

Try first: p_0 to p_3, and the four weights. Which way wins?

$$p_0 = 0.031 \times 0.049 \times 0.073 = 0.00011$$

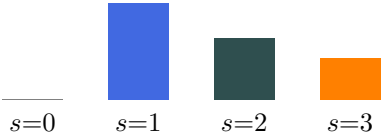
$$p_1 = 25 \times 0.049 \times 0.073 = 0.089$$

$$p_2 = 25 \times 0.031 \times 0.073 = 0.057$$

$$p_3 = 25 \times 0.031 \times 0.049 = 0.038$$

sum 0.184

weights 0.001, 0.486, 0.307, 0.206



Worked: weighing four ways

Our room: lamp centre, the green wall, the floor, the eye. Per unit area:

$\mathbf{y}$: $p_{\rightarrow} = 1/A = 25$, $p_{\leftarrow} = 0.031$ $\mathbf{x}_1$: $p_{\rightarrow} = 0.031$, $p_{\leftarrow} = 0.049$ $\mathbf{x}_2$: $p_{\rightarrow} = 0.049$, $p_{\leftarrow} = 0.073$

Try first: p_0 to p_3, and the four weights. Which way wins?

$$p_0 = 0.031 \times 0.049 \times 0.073 = 0.00011$$

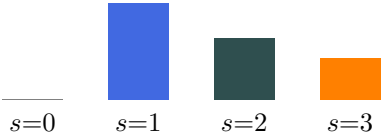
$$p_1 = 25 \times 0.049 \times 0.073 = 0.089$$

$$p_2 = 25 \times 0.031 \times 0.073 = 0.057$$

$$p_3 = 25 \times 0.031 \times 0.049 = 0.038$$

sum 0.184

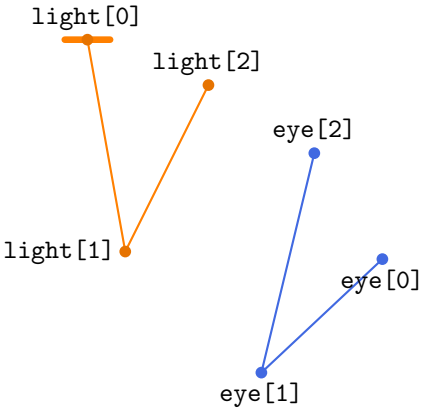
weights 0.001, 0.486, 0.307, 0.206



- No way wins outright: the shadow ray takes half, the two new ways the rest.

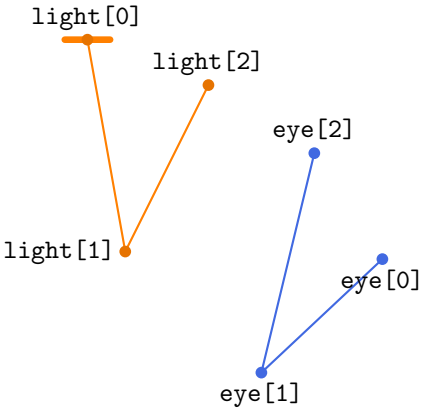
Bidirectional, in Python

```
def bdpt_sample(scene, pixel):
    light = light_subpath(scene)          # y0, y1, ...
    eye = eye_subpath(scene, pixel)       # z0, z1, ...
```



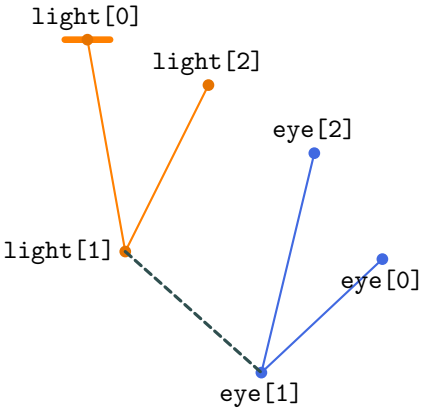
Bidirectional, in Python

```
def bdpt_sample(scene, pixel):
    light = light_subpath(scene)          # y0, y1, ...
    eye = eye_subpath(scene, pixel)       # z0, z1, ...

    for t in range(1, len(eye) + 1):
        for s in range(0, len(light) + 1):
```



Bidirectional, in Python

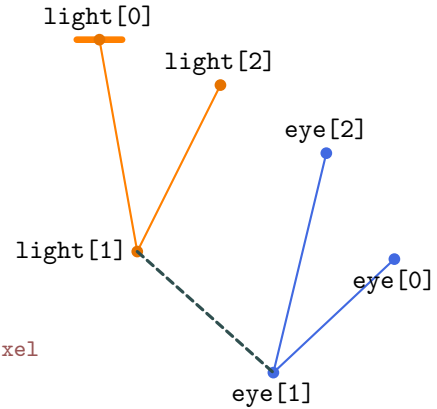
```
def bdpt_sample(scene, pixel):
    light = light_subpath(scene)          # y0, y1, ...
    eye = eye_subpath(scene, pixel)    # z0, z1, ...

    for t in range(1, len(eye) + 1):
        for s in range(0, len(light) + 1):

            c, where = connect(light[:s], eye[:t])
            if c is None:
                continue      # blocked, or a spike
```



$s = 2$, $t = 2$: light[1] to eye[1]

Bidirectional, in Python

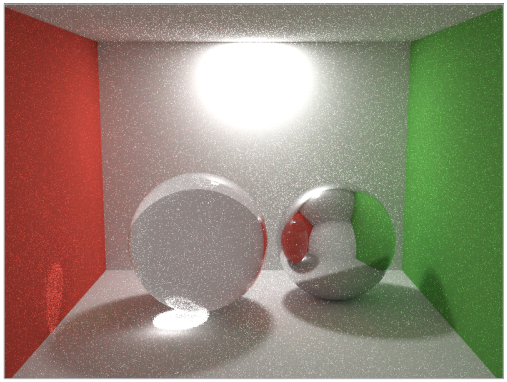
```
def bdpt_sample(scene, pixel):
    light = light_subpath(scene)          # y0, y1, ...
    eye = eye_subpath(scene, pixel)    # z0, z1, ...

    for t in range(1, len(eye) + 1):
        for s in range(0, len(light) + 1):

            c, where = connect(light[:s], eye[:t])
            if c is None:
                continue      # blocked, or a spike

            w = mis_weight(light, eye, s, t)
            image.add(where, w * c)  # t == 1:  any pixel
```



$s = 2, t = 2$: `light[1]` to `eye[1]`

Rendered: same time, two tracers

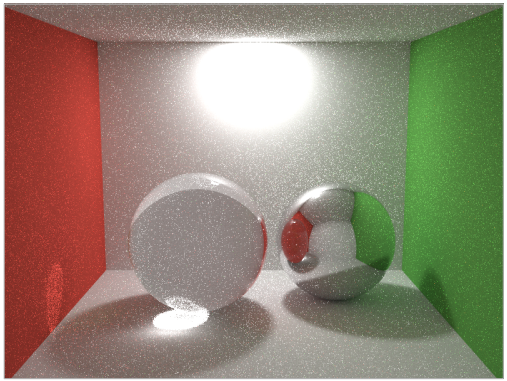
the same time on 12 cores for both pictures of a pair



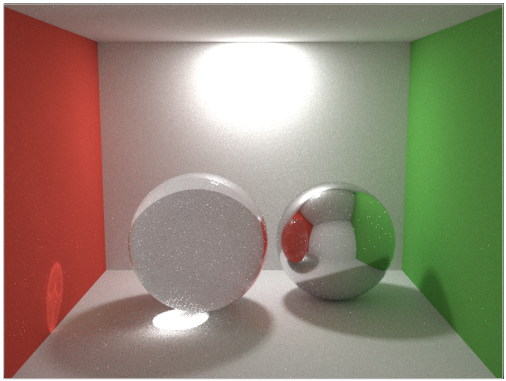
Extra 1's tracer, 320 per pixel

Rendered: same time, two tracers

the same time on 12 cores for both pictures of a pair



Extra 1's tracer, 320 per pixel

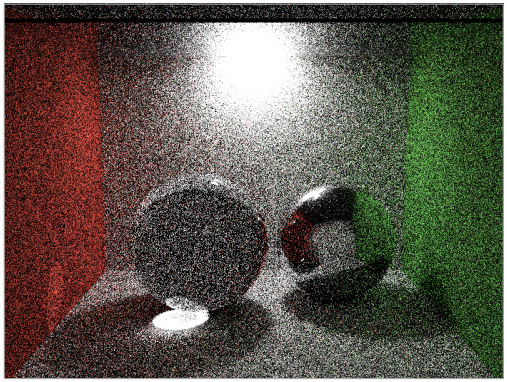


BDPT, 24 per pixel

- The specks are gone. Inside the glass ball some stay: L S S D S S E.

Rendered: same time, two tracers

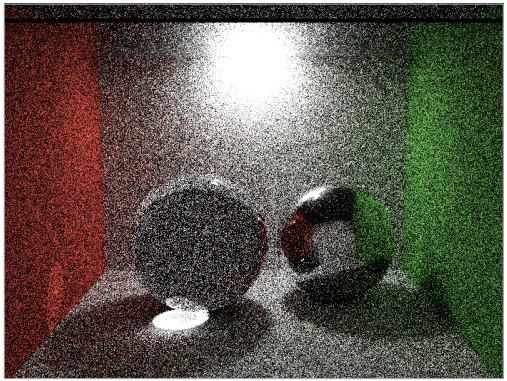
the same time on 12 cores for both pictures of a pair



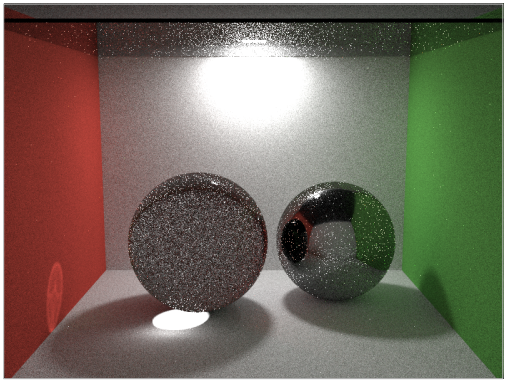
Extra 1's tracer, with the pane

Rendered: same time, two tracers

the same time on 12 cores for both pictures of a pair



Extra 1's tracer, with the pane



BDPT, with the pane

- Through the pane: easy from the lamp's side. Inside the ball: specks again.

Quick check 1: paths and joins

L, S, D, E; k edges, $k+2$ ways; weight $p_s/(p_0 + p_1 + \dots)$

1. Lamp, mirror, floor, wall, eye. Its string?
2. A path has 3 edges. How many ways? How many with a pinhole?
3. L S D E: on which edge can a join work?
4. Two ways can make a path, densities 0.06 and 0.02. Weights?

- Joins need chalk (or the lamp or the eye) at both ends; the weights share out each path.

Quick check 1: paths and joins

L, S, D, E; k edges, $k+2$ ways; weight $p_s/(p_0 + p_1 + \dots)$

1. Lamp, mirror, floor, wall, eye. Its string? L S D D E
2. A path has 3 edges. How many ways? How many with a pinhole? 5; 4
3. L S D E: on which edge can a join work? D to E
4. Two ways can make a path, densities 0.06 and 0.02. Weights? 0.75, 0.25

- Joins need chalk (or the lamp or the eye) at both ends; the weights share out each path.

Recap: who finds which paths

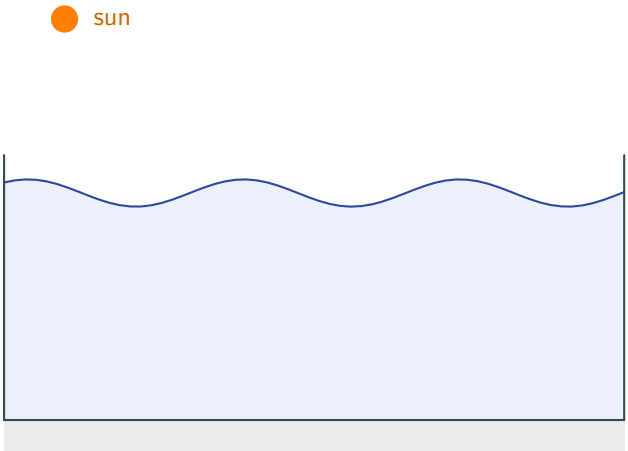
path	for example	Extra 1's tracer	BDPT
$L\,D\,E$	direct light	✓	✓
$L\,D\,D^{+}\,E$	light bounced off walls	✓	✓
$L\,S^{+}\,D\,E$	caustic: under the glass ball	∼	✓
$L\,S^{+}\,D\,S^{+}\,E$	the pool's floor, through water	×	×

✓ clean ∼ noisy: found by luck × never, with a point lamp and a pinhole eye

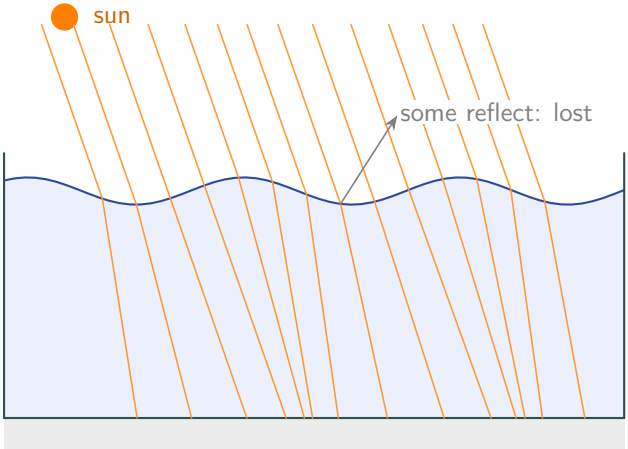
Photon maps

Store the lamp's light, look it up later

Pass 1: shoot photons, keep where they land

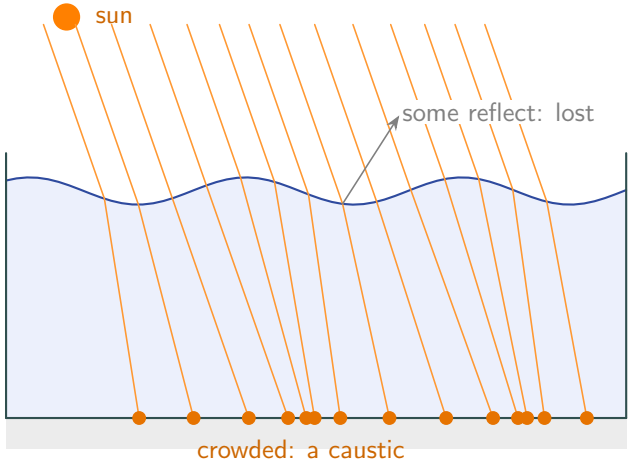


Pass 1: shoot photons, keep where they land



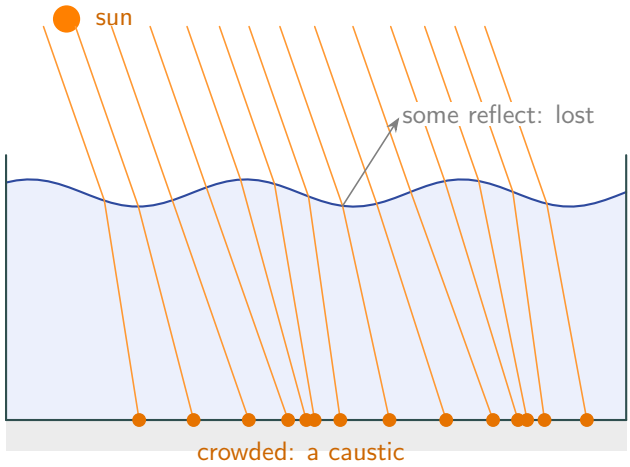
- Light paths from the sun, through the water, to the first chalk point.

Pass 1: shoot photons, keep where they land



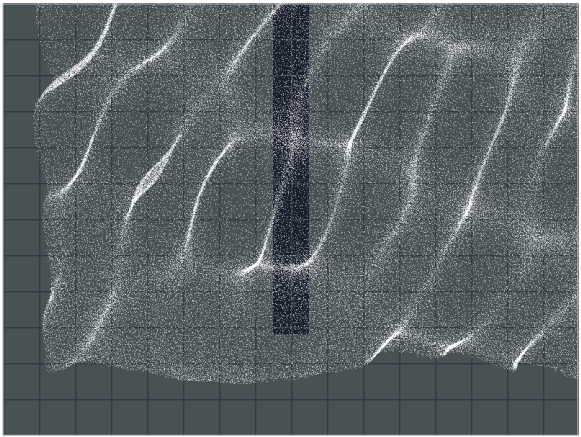
- Light paths from the sun, through the water, to the first chalk point.
- Keep a photon there: its position and its power Φ. Stop.

Pass 1: shoot photons, keep where they land



- Light paths from the sun, through the water, to the first chalk point.
- Keep a photon there: its position and its power Φ. Stop.
- The sun sends 60 into the water: 600 000 photons of 0.0001.

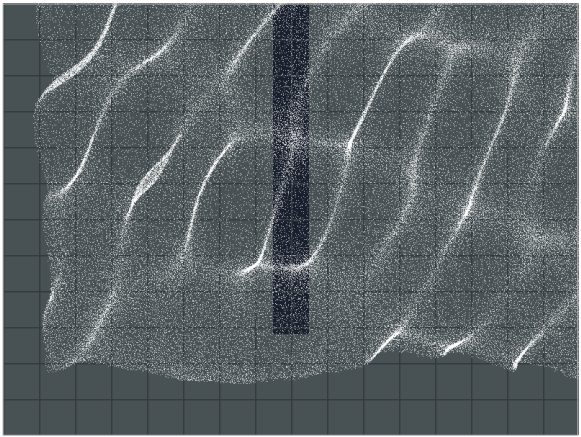
Rendered: the photon map



the near half of the floor, from above, water taken away

- One photon in three, as a white dot.

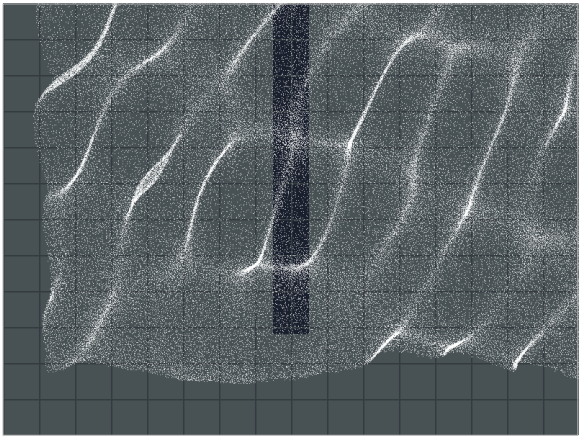
Rendered: the photon map



the near half of the floor, from above, water taken away

- One photon in three, as a white dot.
- Where the waves focus, photons crowd: the bright lines of the floor.

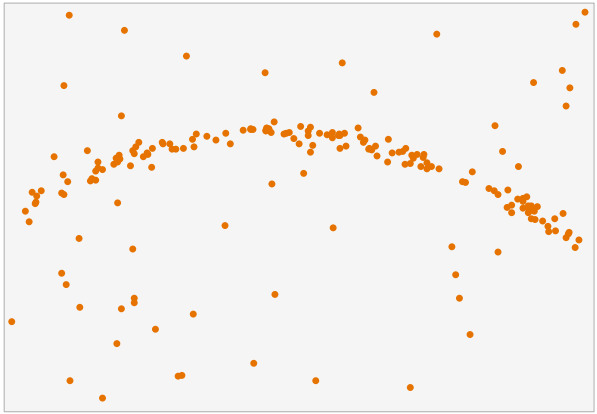
Rendered: the photon map



the near half of the floor, from above, water taken away

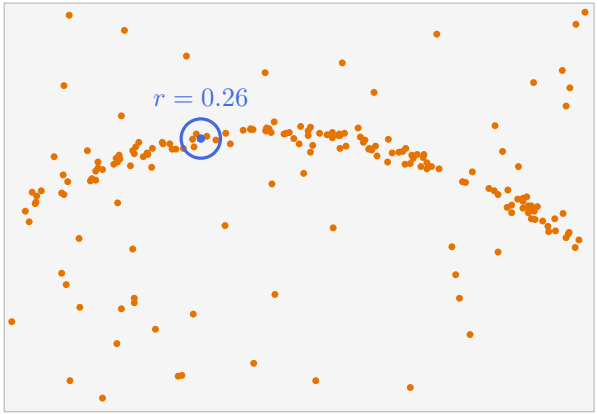
- One photon in three, as a white dot.
- Where the waves focus, photons crowd: the bright lines of the floor.
- The dark strip: the lane line. The empty bands: the walls' shadows.

Pass 2: count the photons near a point



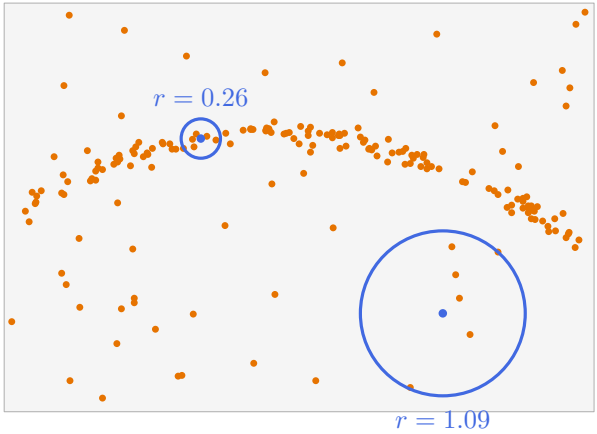
- The eye's path, to its first chalk point: lecture 20's.

Pass 2: count the photons near a point



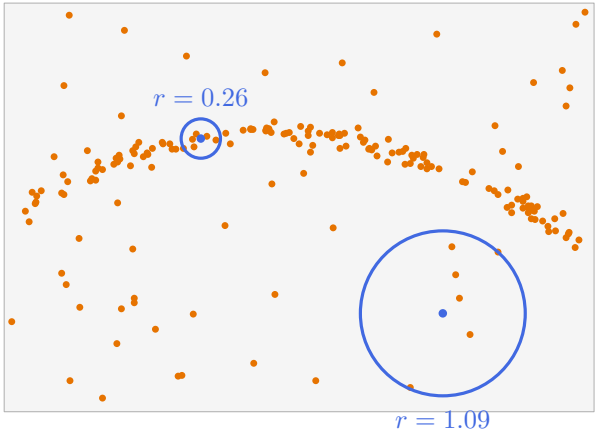
- The eye's path, to its first chalk point: lecture 20's.
- There, find the k nearest photons ($k = 6$ here); r reaches the k-th.

Pass 2: count the photons near a point



- The eye's path, to its first chalk point: lecture 20's.
- There, find the k nearest photons ($k = 6$ here); r reaches the k-th.

Pass 2: count the photons near a point

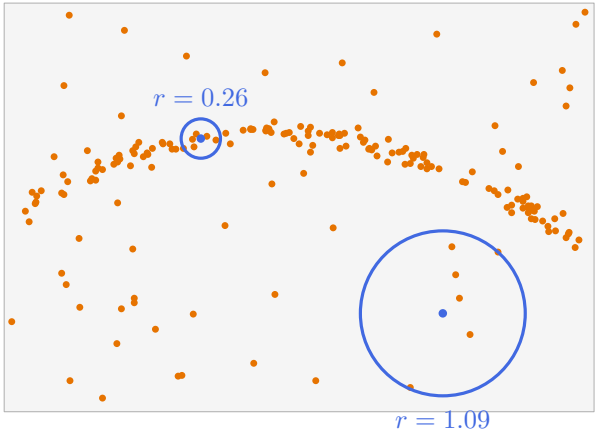


- The eye's path, to its first chalk point: lecture 20's.
- There, find the k nearest photons ($k = 6$ here); r reaches the k-th.

$$E \approx \frac{\Phi_1 + \Phi_2 + \cdots + \Phi_k}{\pi r^2}$$

$$L = \frac{R}{\pi} E$$

Pass 2: count the photons near a point



- The eye's path, to its first chalk point: lecture 20's.
- There, find the k nearest photons ($k = 6$ here); r reaches the k-th.

$$E \approx \frac{\Phi_1 + \Phi_2 + \cdots + \Phi_k}{\pi r^2}$$

$$L = \frac{R}{\pi} E$$

- Same k, a quarter of the radius: about 18 times the light.

Worked: light from 60 photons

Our pool: every photon carries 0.0001; the tiles' albedo $R = 0.62$; $k = 60$.

Try first: E and L where the 60 lie within 3 cm; then within 1 cm.

Worked: light from 60 photons

Our pool: every photon carries 0.0001; the tiles' albedo $R = 0.62$; $k = 60$.

Try first: E and L where the 60 lie within 3 cm; then within 1 cm.

$$r = 0.03\text{:}\quad E = \frac{60 \times 0.0001}{\pi \times 0.03^2} = \frac{0.006}{0.00283} = 2.12, \qquad L = \frac{0.62}{\pi} \times 2.12 = 0.42$$

Worked: light from 60 photons

Our pool: every photon carries 0.0001; the tiles' albedo $R = 0.62$; $k = 60$.

Try first: E and L where the 60 lie within 3 cm; then within 1 cm.

$$r = 0.03\text{:}\quad E = \frac{60 \times 0.0001}{\pi \times 0.03^2} = \frac{0.006}{0.00283} = 2.12, \qquad L = \frac{0.62}{\pi} \times 2.12 = 0.42$$

$$r = 0.01\text{:}\quad E = \frac{0.006}{\pi \times 0.01^2} = 19.1, \qquad L = 0.197 \times 19.1 = 3.77$$

Worked: light from 60 photons

Our pool: every photon carries 0.0001; the tiles' albedo $R = 0.62$; $k = 60$.

Try first: E and L where the 60 lie within 3 cm; then within 1 cm.

$$r = 0.03\text{:}\quad E = \frac{60 \times 0.0001}{\pi \times 0.03^2} = \frac{0.006}{0.00283} = 2.12, \qquad L = \frac{0.62}{\pi} \times 2.12 = 0.42$$

$$r = 0.01\text{:}\quad E = \frac{0.006}{\pi \times 0.01^2} = 19.1, \qquad L = 0.197 \times 19.1 = 3.77$$

- A third of the radius, nine times the light: the bright line clips to white.

Worked: light from 60 photons

Our pool: every photon carries 0.0001; the tiles' albedo $R = 0.62$; $k = 60$.

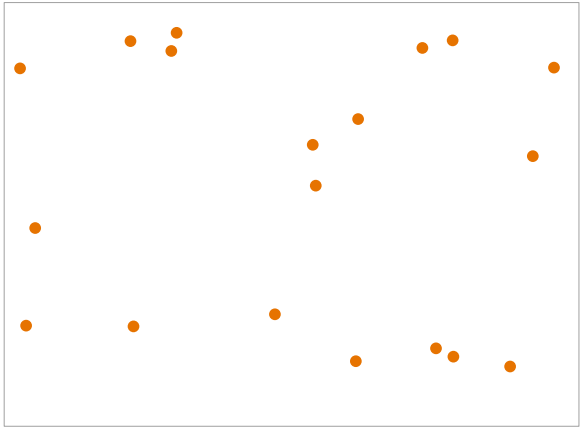
Try first: E and L where the 60 lie within 3 cm; then within 1 cm.

$$r = 0.03\text{:}\quad E = \frac{60 \times 0.0001}{\pi \times 0.03^2} = \frac{0.006}{0.00283} = 2.12, \qquad L = \frac{0.62}{\pi} \times 2.12 = 0.42$$

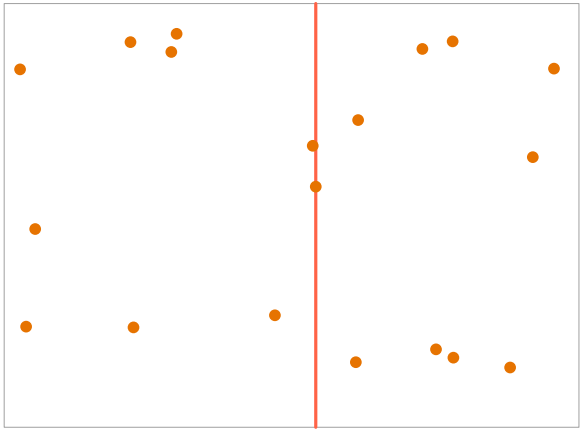
$$r = 0.01\text{:}\quad E = \frac{0.006}{\pi \times 0.01^2} = 19.1, \qquad L = 0.197 \times 19.1 = 3.77$$

- A third of the radius, nine times the light: the bright line clips to white.
- Measured on our floor: r is 3.3 cm at the middle, 1.1 cm on the brightest lines.

Finding the nearest photons: a kd-tree

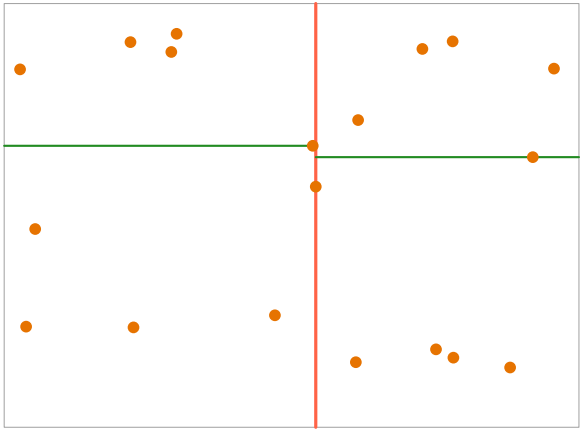


Finding the nearest photons: a kd-tree



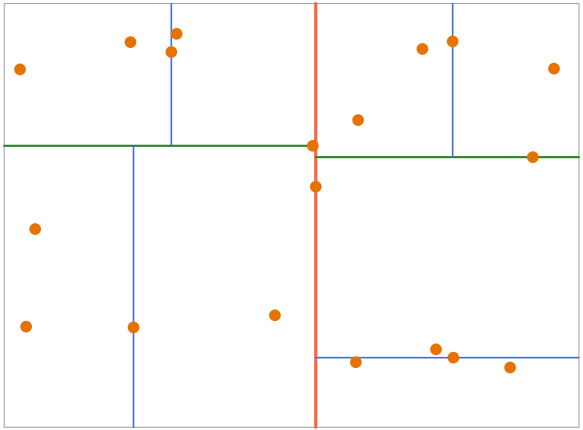
- Build: split the box at its middle photon, across its wider side.

Finding the nearest photons: a kd-tree



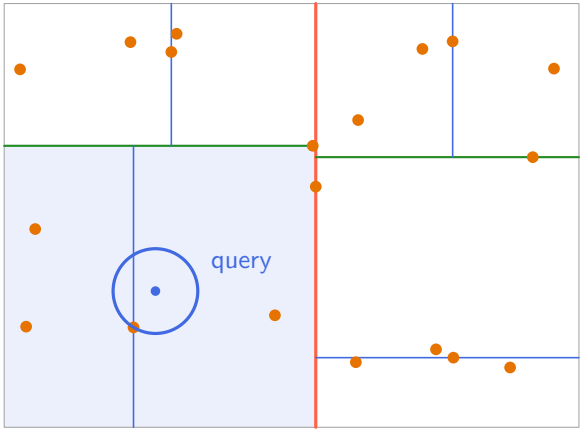
- Build: split the box at its middle photon, across its wider side.
- Split each half the same way, and so on, down to one photon.

Finding the nearest photons: a kd-tree



- Build: split the box at its middle photon, across its wider side.
- Split each half the same way, and so on, down to one photon.
- 600 000 photons: about 20 levels.

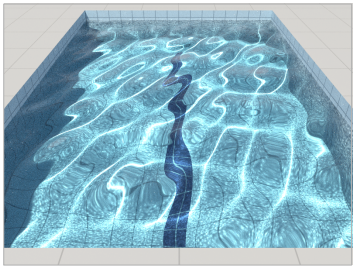
Finding the nearest photons: a kd-tree



- Build: split the box at its middle photon, across its wider side.
- Split each half the same way, and so on, down to one photon.
- 600 000 photons: about 20 levels.
- Search: go down to the query's box; open another box only if the circle reaches it.

Rendered: few photons or many

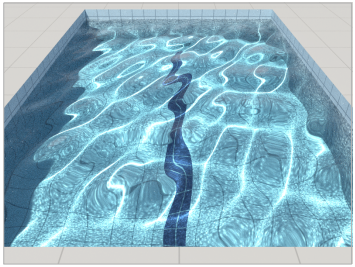
the same 600 000 photons; only k, the number gathered, changes



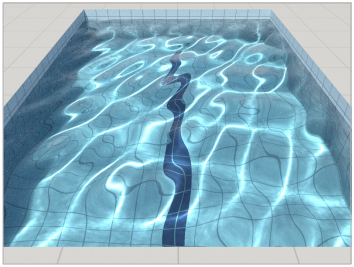
$k = 6$

Rendered: few photons or many

the same 600 000 photons; only k, the number gathered, changes



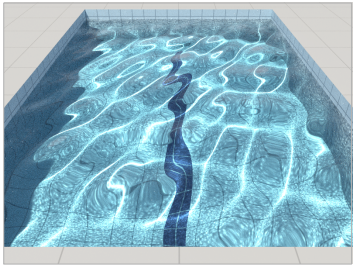
$k = 6$



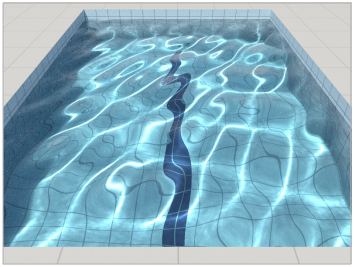
$k = 60$

Rendered: few photons or many

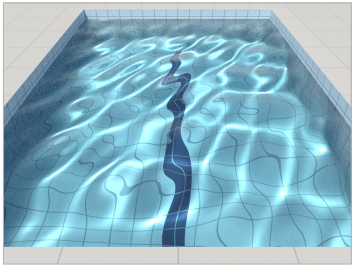
the same 600 000 photons; only k, the number gathered, changes



$k = 6$



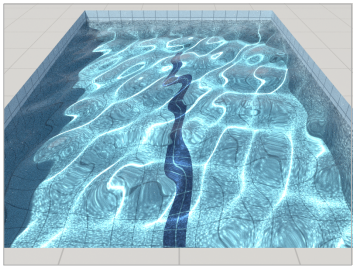
$k = 60$



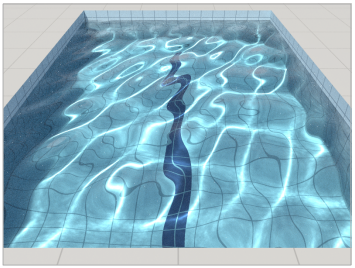
$k = 600$

Rendered: few photons or many

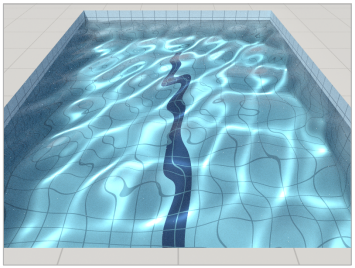
the same 600 000 photons; only k, the number gathered, changes



$k = 6$



$k = 60$

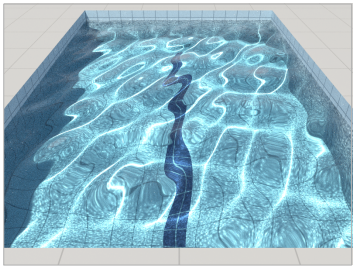


$k = 600$

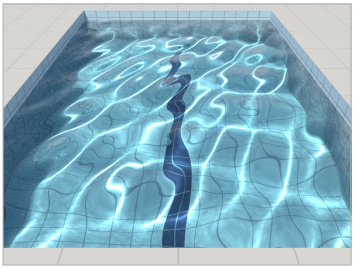
- Small k: the count itself is noisy, a mottle over the floor.

Rendered: few photons or many

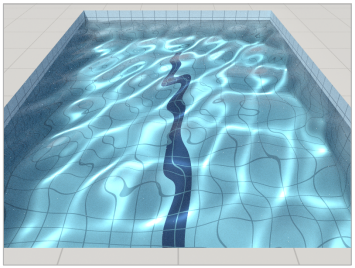
the same 600 000 photons; only k, the number gathered, changes



$k = 6$



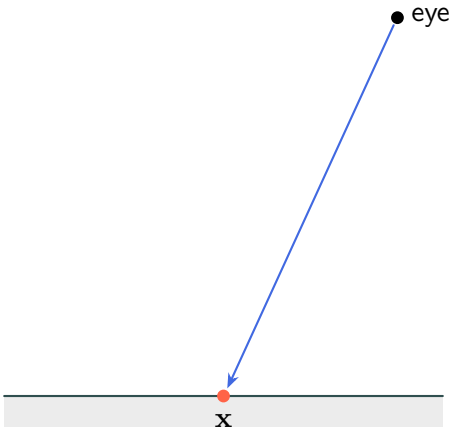
$k = 60$



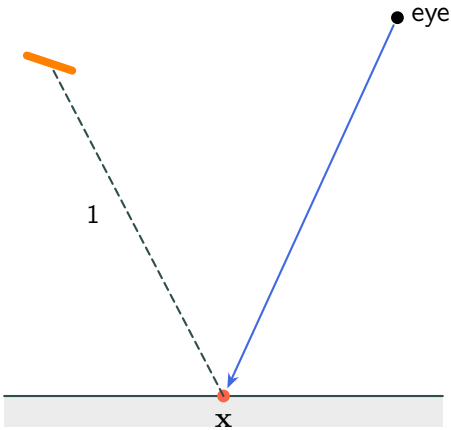
$k = 600$

- Small k: the count itself is noisy, a mottle over the floor.
- Large k: r grows and the lines smear: smooth, but wrong. More photons fix both.

Jensen's recipe: path tracing plus a caustic map

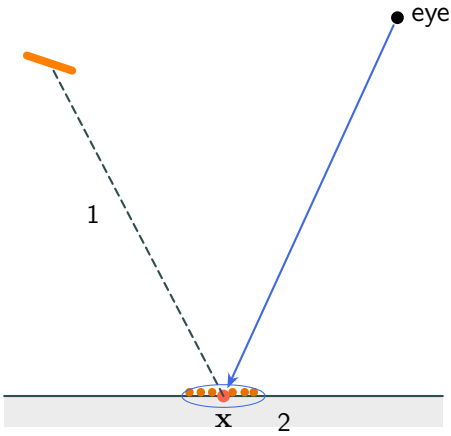


Jensen's recipe: path tracing plus a caustic map



1 L D, straight from the lamp: shadow rays

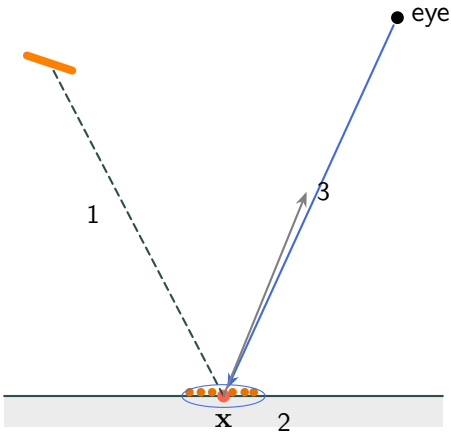
Jensen's recipe: path tracing plus a caustic map



1 L D, straight from the lamp: shadow rays

2 L S^+ D, through glass: the map

Jensen's recipe: path tracing plus a caustic map

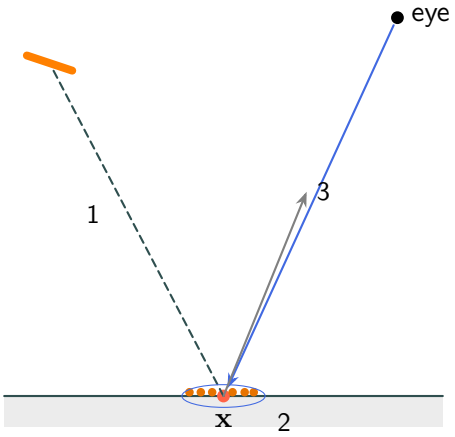


1 L D, straight from the lamp: shadow rays

2 L S^+ D, through glass: the map

3 the rest: bounce on (lecture 20)

Jensen's recipe: path tracing plus a caustic map



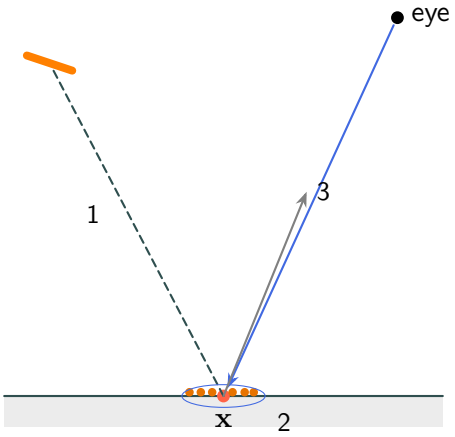
1 L D, straight from the lamp: shadow rays

2 L S^+ D, through glass: the map

3 the rest: bounce on (lecture 20)

a bounce that finds the lamp through glass is not counted: the map has it

Jensen's recipe: path tracing plus a caustic map



1 L D, straight from the lamp: shadow rays

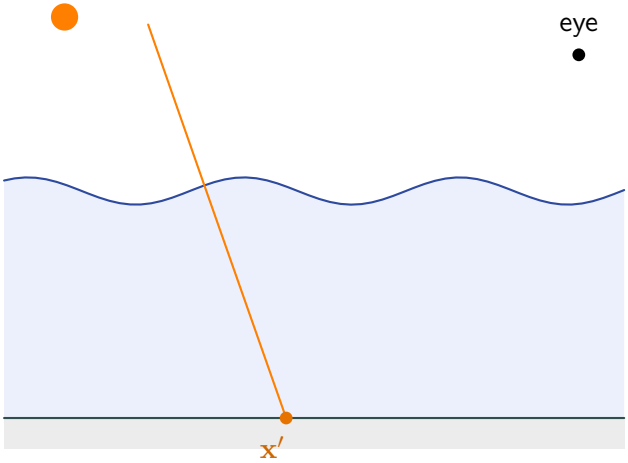
2 L S^+ D, through glass: the map

3 the rest: bounce on (lecture 20)

a bounce that finds the lamp through glass is not counted: the map has it

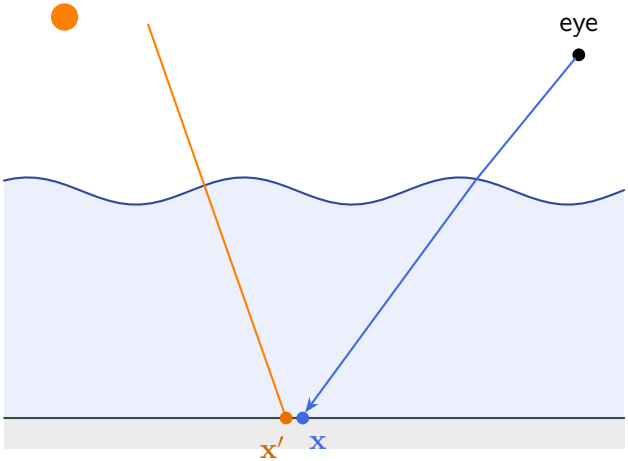
- Each path is counted once. The map is used only where only it works.

Why the map can make L S D S E



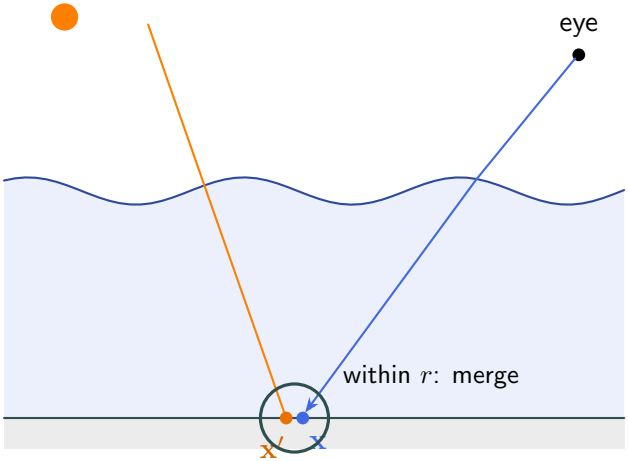
- The light path ends at $\mathbf{x}'$: L S D.

Why the map can make L S D S E



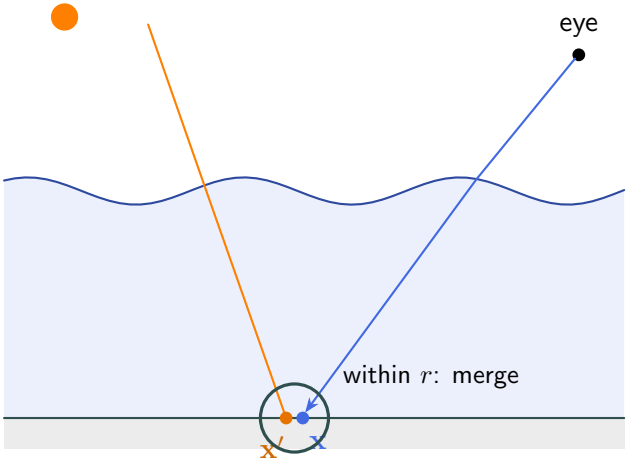
- The light path ends at $\mathbf{x}'$: L S D.
- The eye path ends at $\mathbf{x}$, near it: D S E.

Why the map can make L S D S E



- The light path ends at $\mathbf{x}'$: L S D.
- The eye path ends at $\mathbf{x}$, near it: D S E.
- Not joined by a line: merged, as if $\mathbf{x}' = \mathbf{x}$.

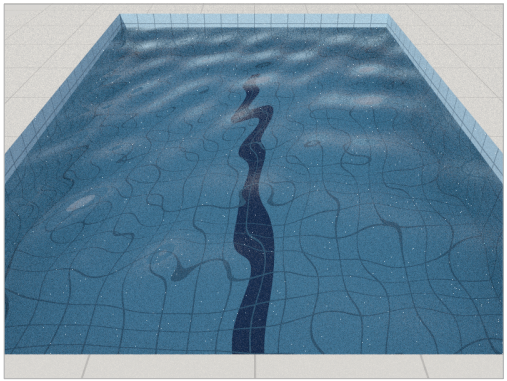
Why the map can make L S D S E



- The light path ends at $\mathbf{x}'$: L S D.
- The eye path ends at $\mathbf{x}$, near it: D S E.
- Not joined by a line: merged, as if $\mathbf{x}' = \mathbf{x}$.
- The price: light moved by up to r, the blur of C25.

Rendered: the pool, two ways

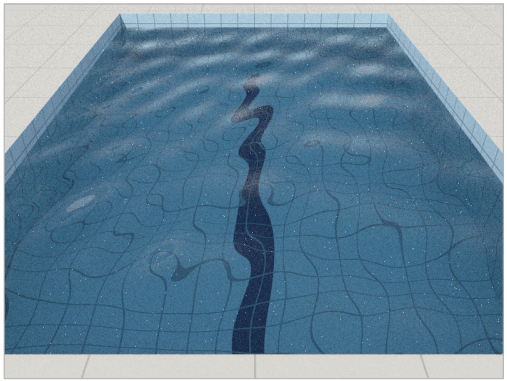
the pool: a sun 0.5 degrees in radius, waves on the water, 600 000 photons



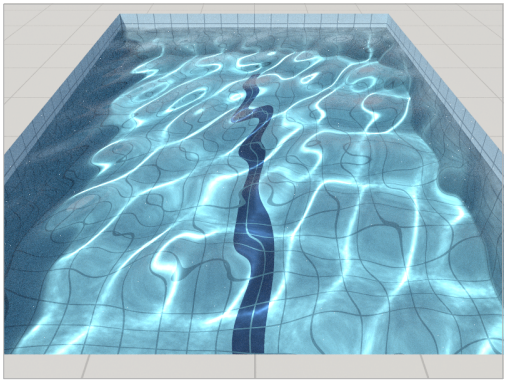
BDPT

Rendered: the pool, two ways

the pool: a sun 0.5 degrees in radius, waves on the water, 600 000 photons



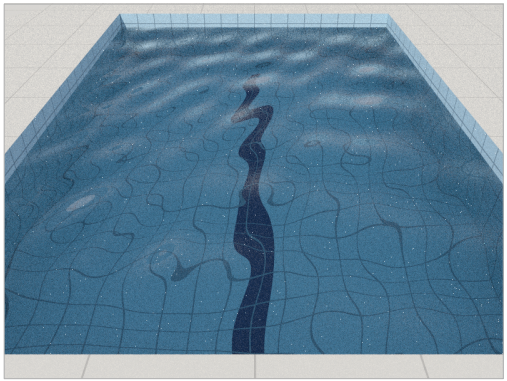
BDPT



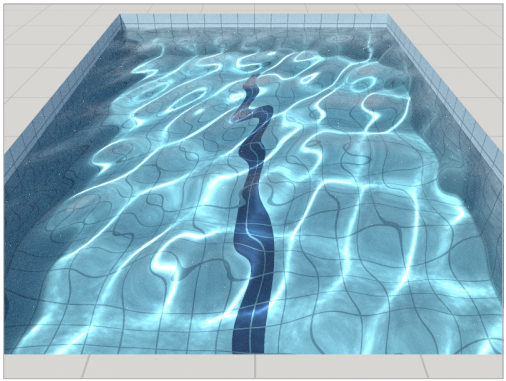
path tracing with the caustic map

Rendered: the pool, two ways

the pool: a sun 0.5 degrees in radius, waves on the water, 600 000 photons



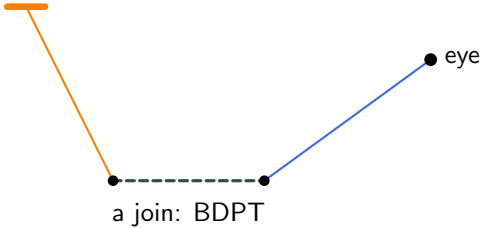
BDPT



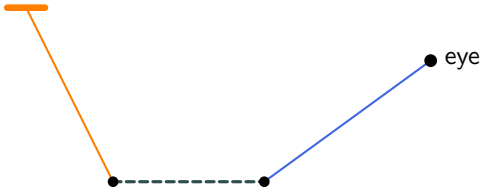
path tracing with the caustic map

- BDPT: no bright lines, only specks of sun. The map: the floor as we see it.

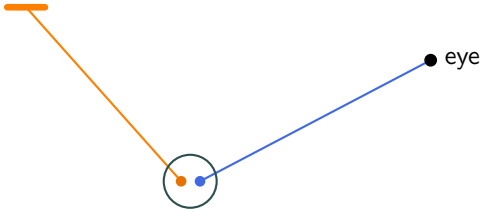
Joining and merging, weighed together



Joining and merging, weighed together

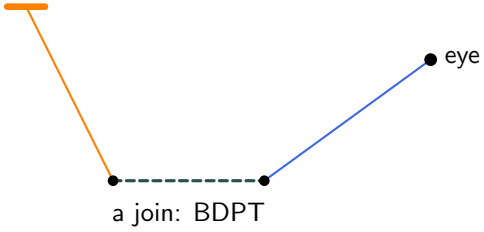


a join: BDPT



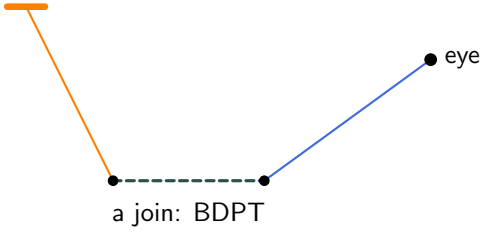
a merge: the photon map

Joining and merging, weighed together



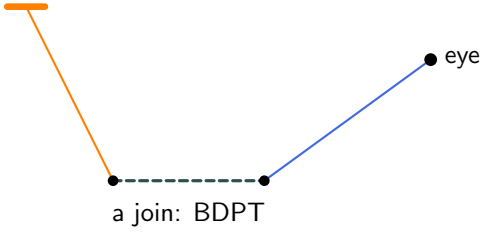
- A merge is one more way to make a path, with its own density.

Joining and merging, weighed together



- A merge is one more way to make a path, with its own density.
- Weigh joins and merges together: joins where they work, merges where only they do.

Joining and merging, weighed together



- A merge is one more way to make a path, with its own density.
- Weigh joins and merges together: joins where they work, merges where only they do.
- Georgiev et al., Hachisuka et al., 2012; RenderMan's VCM.

Quick check 2: photons

Φ **= power over photons;** $E \approx (\Phi_1 + \cdots + \Phi_k)/(\pi r^2)$**;** $L = (R/\pi)\, E$

1. A lamp of power 50 sends 1 000 000 photons. The power of each?
2. 20 photons of 0.001 lie within $r = 0.1$. What is E?
3. The same 20 photons, spread over twice the radius. E?
4. Which can only the map make: L S D E or L S D S E?

- Few photons gathered: noise. Many: blur. Merging makes what no join can.

Quick check 2: photons

Φ = power over photons; $E \approx (\Phi_1 + \cdots + \Phi_k)/(\pi r^2)$**;** $L = (R/\pi)\, E$

1. A lamp of power 50 sends 1 000 000 photons. The power of each? 0.00005
2. 20 photons of 0.001 lie within $r = 0.1$. What is E? 0.64
3. The same 20 photons, spread over twice the radius. E? 0.16
4. Which can only the map make: L S D E or L S D S E? L S D S E

- Few photons gathered: noise. Many: blur. Merging makes what no join can.

Recap: who finds which paths

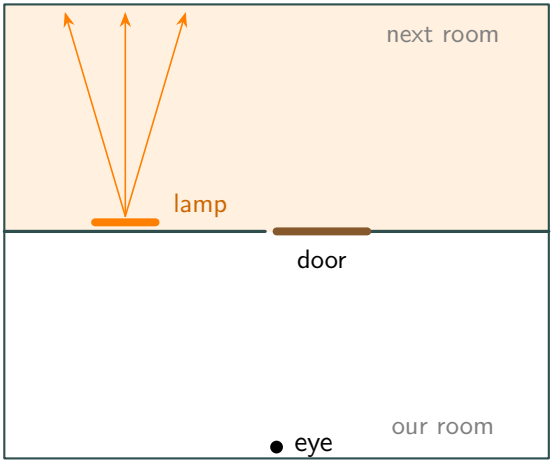
path	for example	Extra 1's tracer	BDPT	photon map
$L D E$	direct light	✓	✓	✓
$L D D^{+} E$	light bounced off walls	✓	✓	✓
$L S^{+} D E$	caustic: under the glass ball	∼	✓	✓
$L S^{+} D S^{+} E$	the pool's floor, through water	×	×	✓

✓ clean ∼ noisy: found by luck × never, with a point lamp and a pinhole eye

Metropolis light transport

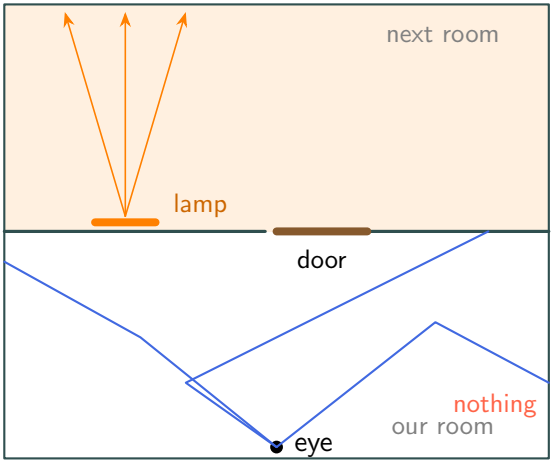
Found a good path? Change it a little

Light through a door left open a crack



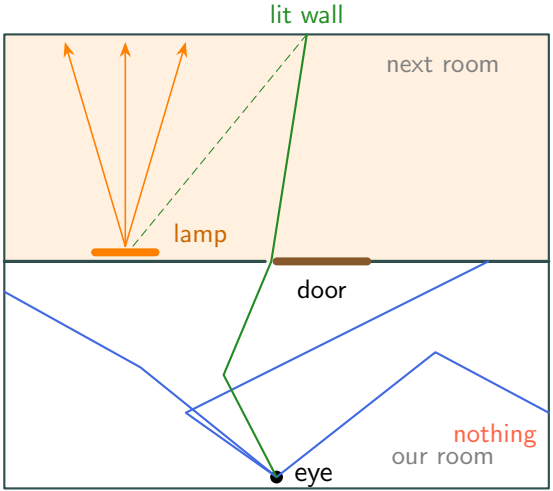
- Our room gets only light that bounced next door and came through the crack.

Light through a door left open a crack



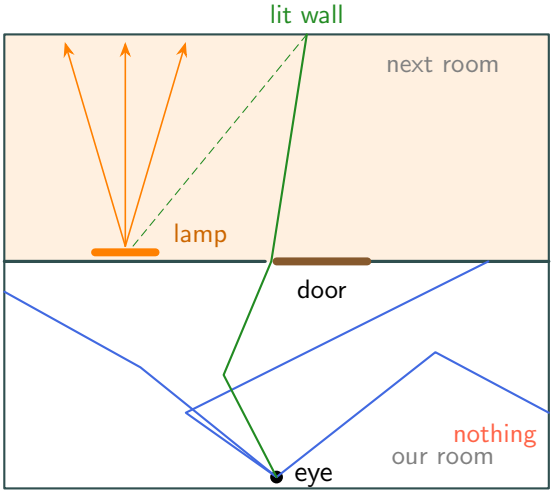
- Our room gets only light that bounced next door and came through the crack.
- From our room, 1 path in 101 brings back any light.

Light through a door left open a crack



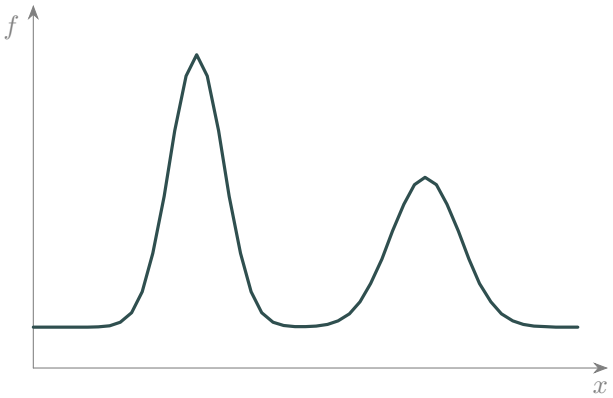
- Our room gets only light that bounced next door and came through the crack.
- From our room, 1 path in 101 brings back any light.
- Once one is found, its neighbours go through the crack too.

Light through a door left open a crack



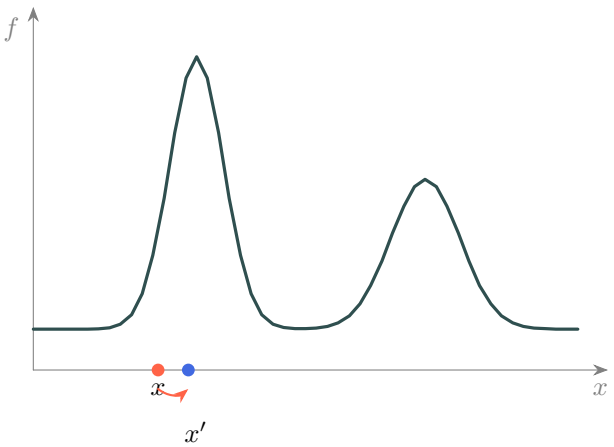
- Our room gets only light that bounced next door and came through the crack.
- From our room, 1 path in 101 brings back any light.
- Once one is found, its neighbours go through the crack too.
- BDPT does not help: light paths must find the crack too.

Metropolis on a line



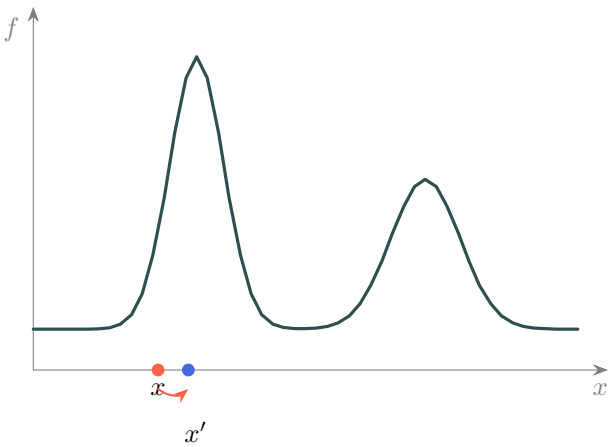
- Goal: visit each x as often as $f(x)$ is high.

Metropolis on a line



- Goal: visit each x as often as $f(x)$ is high.
- Propose a step: x' near x.

Metropolis on a line



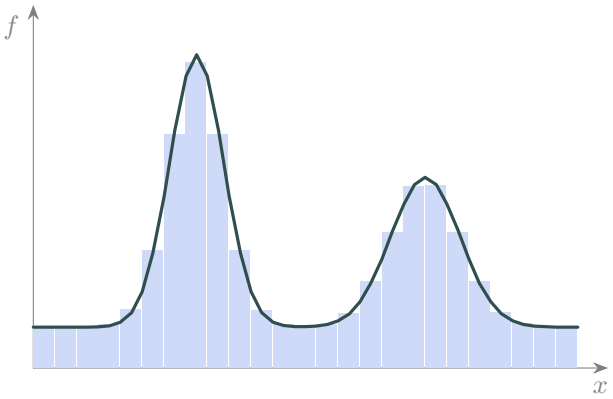
- Goal: visit each x as often as $f(x)$ is high.
- Propose a step: x' near x.

  move with chance

  $$a = \min\left(1,\ \frac{f(x')}{f(x)}\right)$$

  else stay, and count x once more

Metropolis on a line



- Goal: visit each x as often as $f(x)$ is high.
- Propose a step: x' near x.

  move with chance

  $$a = \min\left(1,\ \frac{f(x')}{f(x)}\right)$$

  else stay, and count x once more
- The visits pile up in the shape of f: 400 000 steps.

Worked: stay or move?

The walk is at x with $f(x) = 0.8$.

Try first: the chance of moving, for each proposal.

(a) $f(x') = 0.2$

(b) $f(x') = 1.2$

(c) $f(x') = 0$: no light at all

Worked: stay or move?

The walk is at x with $f(x) = 0.8$.

Try first: the chance of moving, for each proposal.

(a) $f(x') = 0.2$ $\qquad a = 0.2/0.8 = 0.25$

(b) $f(x') = 1.2$

(c) $f(x') = 0$: no light at all

Worked: stay or move?

The walk is at x with $f(x) = 0.8$.

Try first: the chance of moving, for each proposal.

(a) $f(x') = 0.2$ $\qquad a = 0.2/0.8 = 0.25$

(b) $f(x') = 1.2$ $\qquad a = \min(1, 1.5) = 1$

(c) $f(x') = 0$: no light at all

Worked: stay or move?

The walk is at x with $f(x) = 0.8$.

Try first: the chance of moving, for each proposal.

(a) $f(x') = 0.2$ $\qquad a = 0.2/0.8 = 0.25$

(b) $f(x') = 1.2$ $\qquad a = \min(1, 1.5) = 1$

(c) $f(x') = 0$: no light at all $\qquad a = 0$: never

Worked: stay or move?

The walk is at x with $f(x) = 0.8$.

Try first: the chance of moving, for each proposal.

(a) $f(x') = 0.2$ $\qquad a = 0.2/0.8 = 0.25$

(b) $f(x') = 1.2$ $\qquad a = \min(1, 1.5) = 1$

(c) $f(x') = 0$: no light at all $\qquad a = 0$: never

- Once on a path that carries light, the walk never steps onto a dark one.

Worked: stay or move?

The walk is at x with $f(x) = 0.8$.

Try first: the chance of moving, for each proposal.

(a) $f(x') = 0.2$ $\qquad a = 0.2/0.8 = 0.25$

(b) $f(x') = 1.2$ $\qquad a = \min(1, 1.5) = 1$

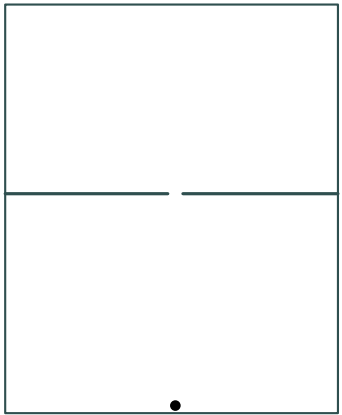
(c) $f(x') = 0$: no light at all $\qquad a = 0$: never

- Once on a path that carries light, the walk never steps onto a dark one.
- In (a), three times in four it stays: x is counted again.

A path is a list of numbers: nudge them

the numbers $\xi_1, \xi_2, \ldots$ lecture 20 drew for one path

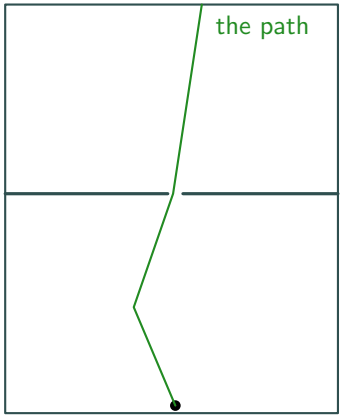
0.41	0.73	0.12	0.88	0.56	0.30	$\cdots$



A path is a list of numbers: nudge them

the numbers $\xi_1, \xi_2, \ldots$ lecture 20 drew for one path

0.41	0.73	0.12	0.88	0.56	0.30	$\cdots$



- The numbers decide the path: keep the list.

A path is a list of numbers: nudge them

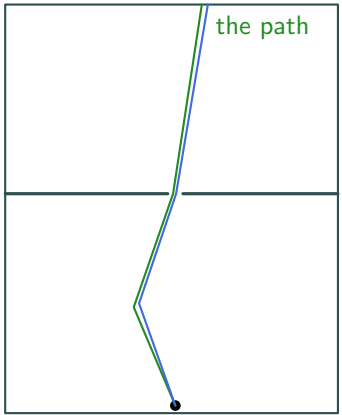
the numbers $\xi_1, \xi_2, \ldots$ lecture 20 drew for one path

0.41	0.73	0.12	0.88	0.56	0.30	$\cdots$

0.42	0.72	0.13	0.87	0.57	0.31	$\cdots$

small step: each moved by about 0.01

- The numbers decide the path: keep the list.
- Nudge every number a little: a path next to it.



A path is a list of numbers: nudge them

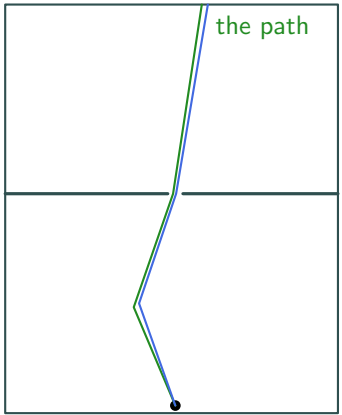
the numbers $\xi_1, \xi_2, \ldots$ lecture 20 drew for one path

0.41	0.73	0.12	0.88	0.56	0.30	$\cdots$

0.42	0.72	0.13	0.87	0.57	0.31	$\cdots$

small step: each moved by about 0.01

- The numbers decide the path: keep the list.
- Nudge every number a little: a path next to it.
- Three steps in ten: all new numbers, a jump anywhere.



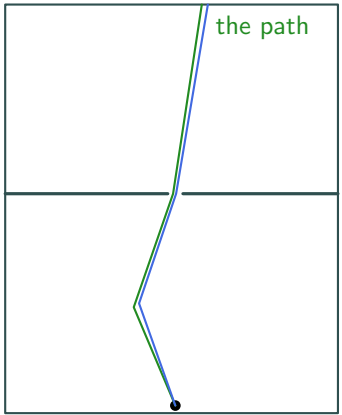
A path is a list of numbers: nudge them

the numbers $\xi_1, \xi_2, \ldots$ lecture 20 drew for one path

0.41	0.73	0.12	0.88	0.56	0.30	$\cdots$

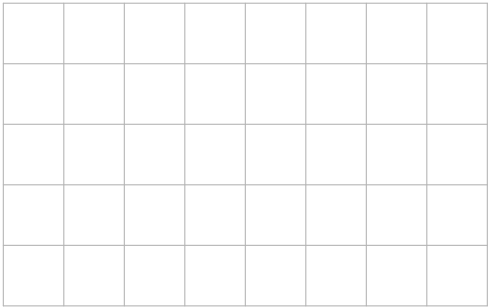
0.42	0.72	0.13	0.87	0.57	0.31	$\cdots$

small step: each moved by about 0.01



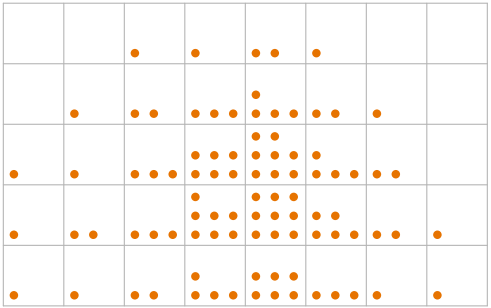
- The numbers decide the path: keep the list.
- Nudge every number a little: a path next to it.
- Three steps in ten: all new numbers, a jump anywhere.
- f = the path's brightness; then C31's rule.

The picture counts the visits



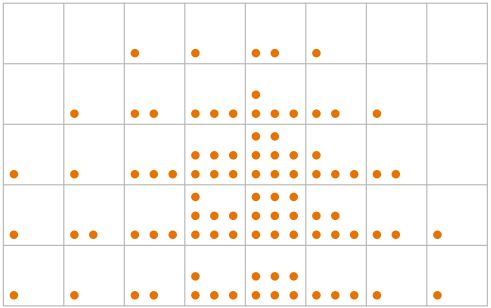
- Every step adds its path's colour to its pixel, scaled to the same size.

The picture counts the visits



- Every step adds its path's colour to its pixel, scaled to the same size.
- Brightness comes from how often a pixel is visited.

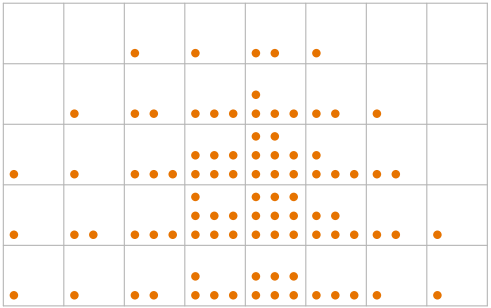
The picture counts the visits



- Every step adds its path's colour to its pixel, scaled to the same size.
- Brightness comes from how often a pixel is visited.

$$\text{each step adds } \frac{b}{\text{steps per pixel}} \cdot \frac{\text{colour}}{f}$$

The picture counts the visits



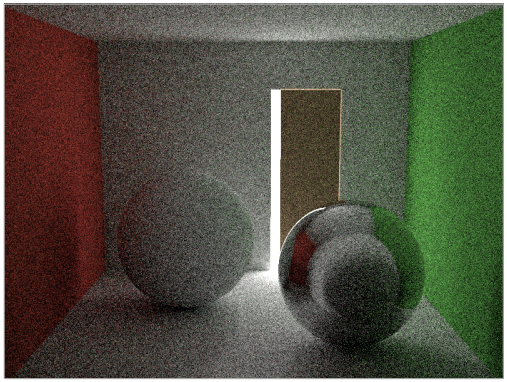
- Every step adds its path's colour to its pixel, scaled to the same size.
- Brightness comes from how often a pixel is visited.

$$\text{each step adds } \frac{b}{\text{steps per pixel}} \cdot \frac{\text{colour}}{f}$$

- b: the average f of fresh paths, measured first (200 000 of them).

Rendered: the door, same time

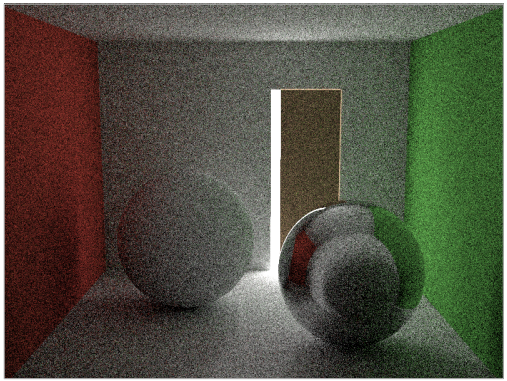
Extra 1's room; the next room's light comes only through the crack



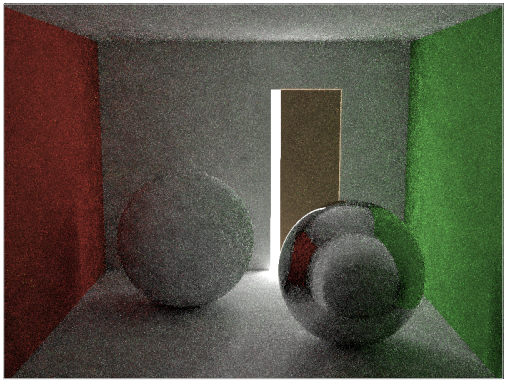
Extra 1's tracer, 400 per pixel

Rendered: the door, same time

Extra 1's room; the next room's light comes only through the crack



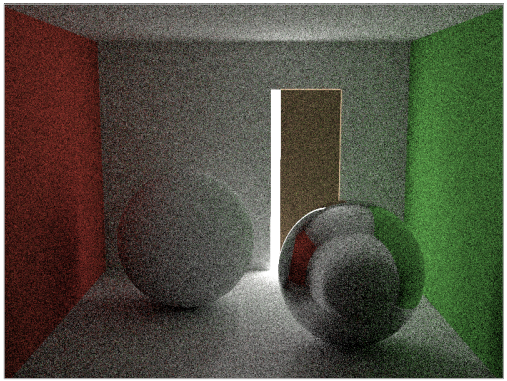
Extra 1's tracer, 400 per pixel



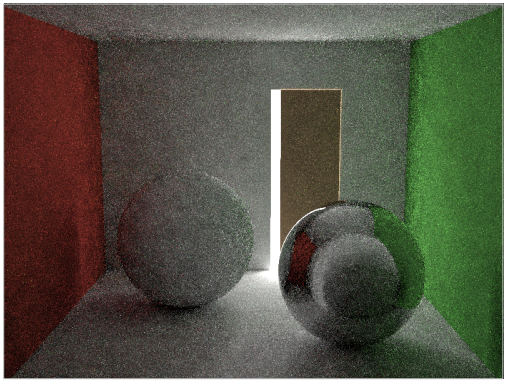
Metropolis, 256 steps per pixel

Rendered: the door, same time

Extra 1's room; the next room's light comes only through the crack



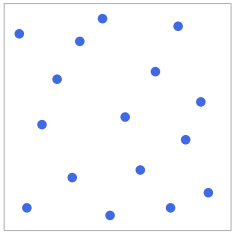
Extra 1's tracer, 400 per pixel



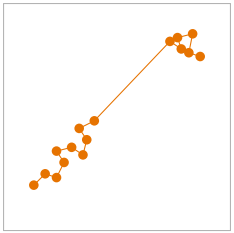
Metropolis, 256 steps per pixel

- Same time: finer grain, and the walls' shading shows. The noise is uneven.

Why nobody ships it

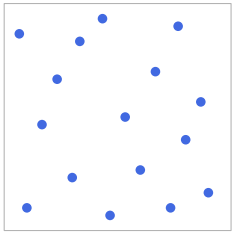


fresh paths:
even, but often dark

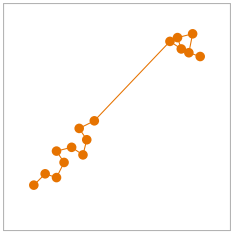


one walk:
bright, but in clumps

Why nobody ships it



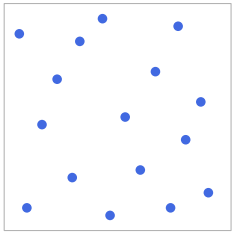
fresh paths:
even, but often dark



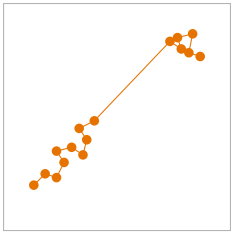
one walk:
bright, but in clumps

- Uneven: some places well visited, some not yet. Blotches, not grain.

Why nobody ships it



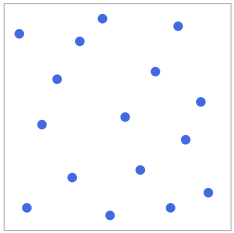
fresh paths:
even, but often dark



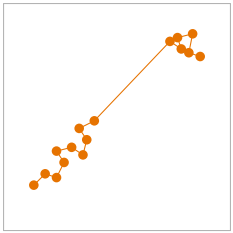
one walk:
bright, but in clumps

- Uneven: some places well visited, some not yet. Blotches, not grain.
- In a film, each frame's walk wanders differently: flicker.

Why nobody ships it



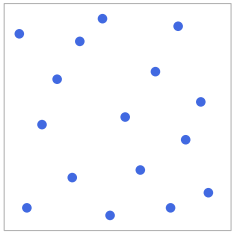
fresh paths:
even, but often dark



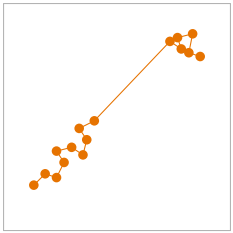
one walk:
bright, but in clumps

- Uneven: some places well visited, some not yet. Blotches, not grain.
- In a film, each frame's walk wanders differently: flicker.
- A GPU wants thousands of separate paths, not a few long walks.

Why nobody ships it



fresh paths:
even, but often dark

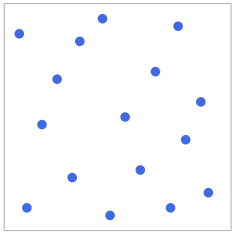


one walk:
bright, but in clumps

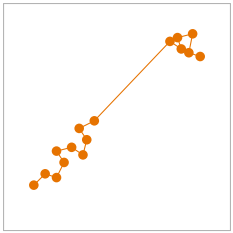
- Uneven: some places well visited, some not yet. Blotches, not grain.
- In a film, each frame's walk wanders differently: flicker.
- A GPU wants thousands of separate paths, not a few long walks.

- Where it wins: light that is hard to find but easy to follow. Doors, keyholes, caustics.

Why nobody ships it



fresh paths:
even, but often dark



one walk:
bright, but in clumps

- Uneven: some places well visited, some not yet. Blotches, not grain.
- In a film, each frame's walk wanders differently: flicker.
- A GPU wants thousands of separate paths, not a few long walks.

- Where it wins: light that is hard to find but easy to follow. Doors, keyholes, caustics.
- What ships instead: path guiding, many-light sampling, ReSTIR (Extra 6).

Recap: who finds which paths

path	for example	Extra 1's tracer	BDPT	photon map	Metropolis
$L\,D\,E$	direct light	✓	✓	✓	✓
$L\,D\,D^{+}\,E$	light bounced off walls	✓	✓	✓	✓
$L\,S^{+}\,D\,E$	caustic: under the glass ball	∼	✓	✓	✓
$L\,S^{+}\,D\,S^{+}\,E$	the pool's floor, through water	×	×	✓	×
$L\,D\,D^{+}\,E$	through a crack: hard to hit	∼	∼	∼	✓

✓ clean ∼ noisy: found by luck × never, with a point lamp and a pinhole eye

Summary

1 Naming paths

$$\mathsf{L}\,(\mathsf{S}|\mathsf{D})^*\,\mathsf{E}$$

a join needs chalk at both ends

2 Paths from both ends

$k+2$ ways, weight $\dfrac{p_s}{p_0 + p_1 + \dots}$

light tracing makes the caustics

3 Photon maps

$$E \approx \frac{\Phi_1 + \dots + \Phi_k}{\pi r^2}$$

merging makes L S D S E; the price is blur

4 Metropolis

move with chance $\min\left(1, \dfrac{f'}{f}\right)$

wins where light is hard to find; uneven

Reading

Pharr, Jakob and Humphreys, *Physically Based Rendering*, 3rd edition (2016)

chapter 16: photon mapping, BDPT, Metropolis; free at pbr-book.org

Veach, PhD thesis, Stanford 1997: chapter 10 (bidirectional), chapter 11 (Metropolis)

Veach and Guibas, "Metropolis light transport", SIGGRAPH 1997

Jensen, *Realistic Image Synthesis Using Photon Mapping*, 2001

Heckbert, "Adaptive radiosity textures for bidirectional ray tracing", SIGGRAPH 1990

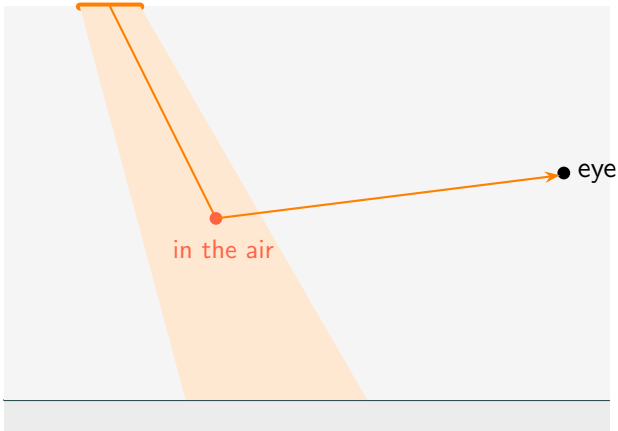
Kelemen et al., "A simple and robust mutation strategy for the Metropolis light transport algorithm", Eurographics 2002

Georgiev et al., "Light transport simulation with vertex connection and merging", SIGGRAPH Asia 2012

Next: Extra 5, volumes

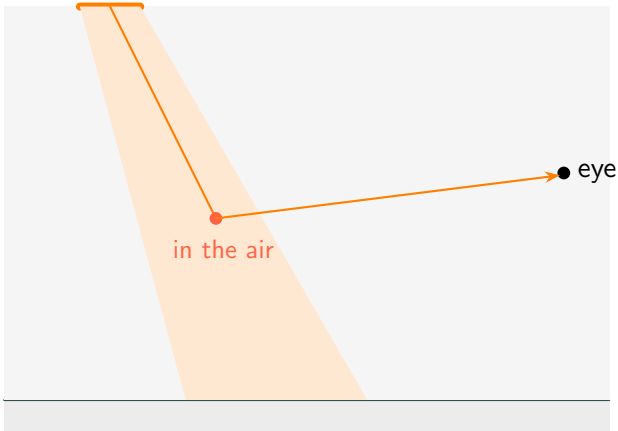


Next: Extra 5, volumes



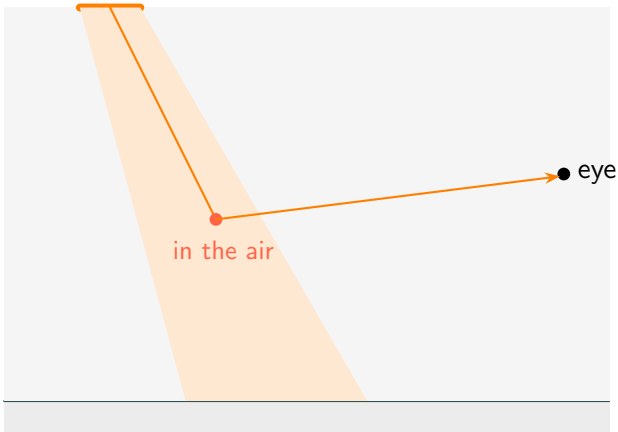
- Fog, smoke, clouds, skin: light turns inside a medium too.

Next: Extra 5, volumes



- Fog, smoke, clouds, skin: light turns inside a medium too.
- A vertex can be anywhere along a ray, not only on a surface.

Next: Extra 5, volumes



- Fog, smoke, clouds, skin: light turns inside a medium too.
- A vertex can be anywhere along a ray, not only on a surface.
- Extra 5: lecture 18's Beer's law, grown into light transport.