Lecture 15: Groups of Objects and Antialiasing

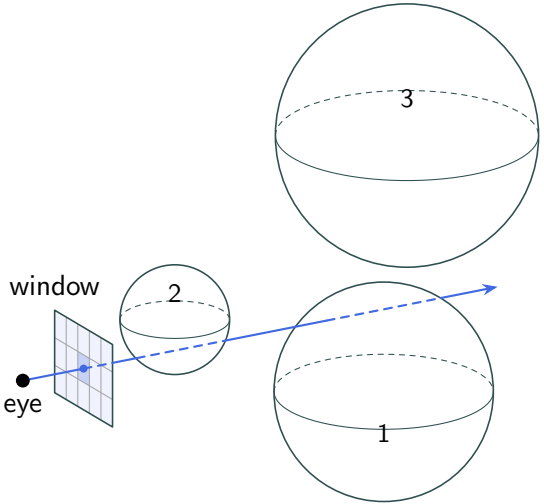
The Nearest Hit in a Scene, and Several Rays per Pixel

CS 453 — Computer Graphics

Fakhir Shaheen

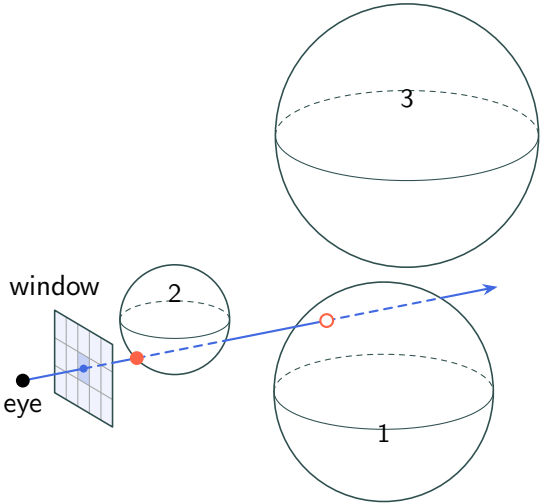
Forman Christian University

Today: which sphere does this pixel show?



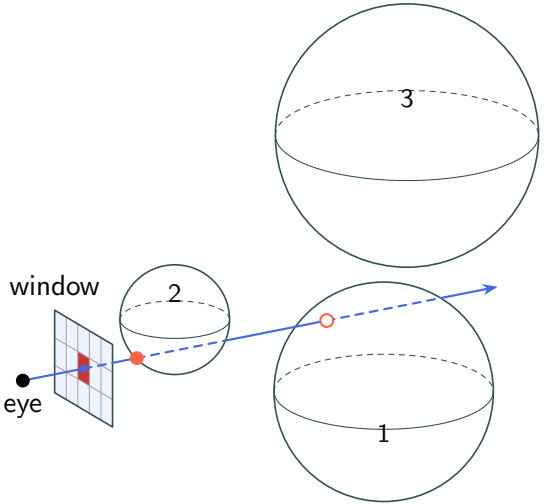
- Three spheres, one ray per pixel

Today: which sphere does this pixel show?



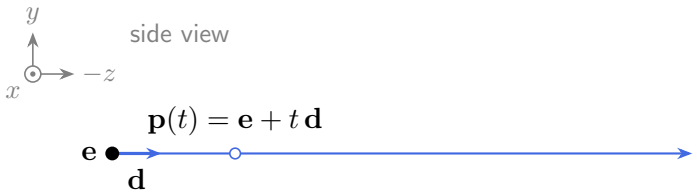
- Three spheres, one ray per pixel
- This ray meets two of them

Today: which sphere does this pixel show?



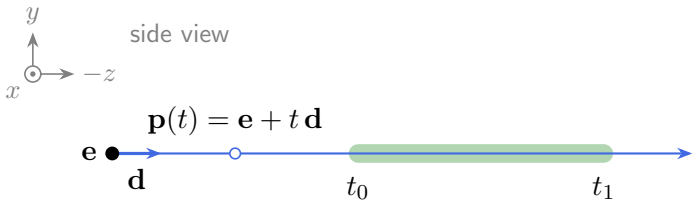
- Three spheres, one ray per pixel
- This ray meets two of them
- The pixel shows the nearest one

Our conventions in one picture



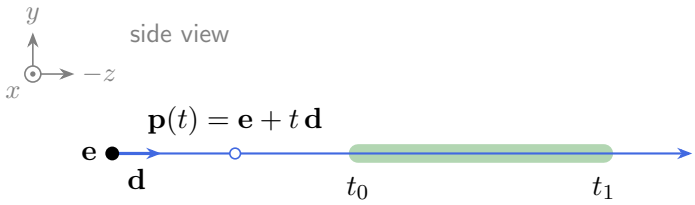
- Ray: start $\mathbf{e}$, direction $\mathbf{d}$

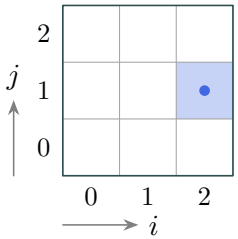
Our conventions in one picture



- Ray: start $\mathbf{e}$, direction $\mathbf{d}$
- Hits count only in $[t_0, t_1]$

Our conventions in one picture

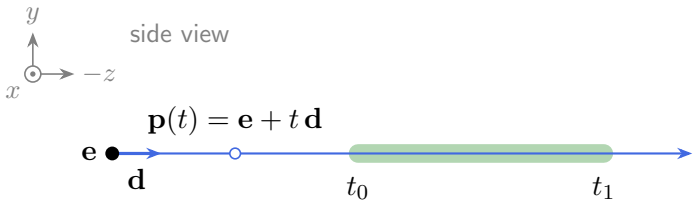


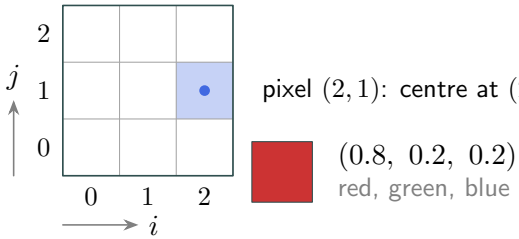


pixel $(2, 1)$: centre at $(2.5,\ 1.5)$

- Ray: start $\mathbf{e}$, direction $\mathbf{d}$
- Hits count only in $[t_0, t_1]$
- Pixel (i, j): j counts from the bottom

Our conventions in one picture



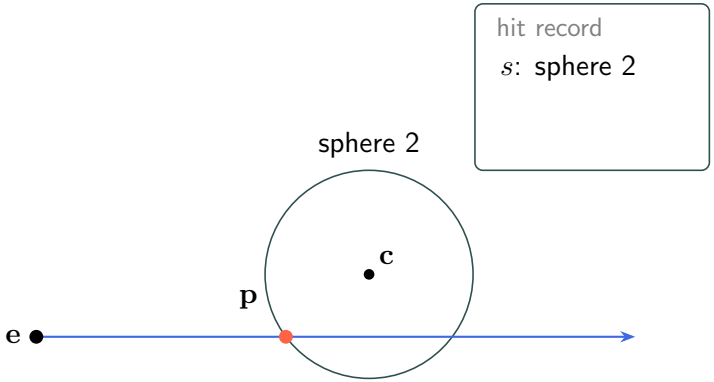


- Ray: start $\mathbf{e}$, direction $\mathbf{d}$
- Hits count only in $[t_0, t_1]$
- Pixel (i, j): j counts from the bottom
- Colour: three numbers, each 0 to 1

What one hit returns

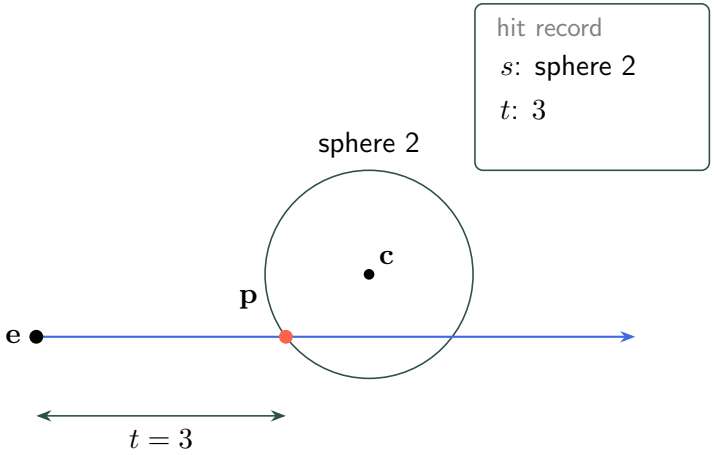
A record: which surface, where, which way

A hit record holds surface, t, and normal



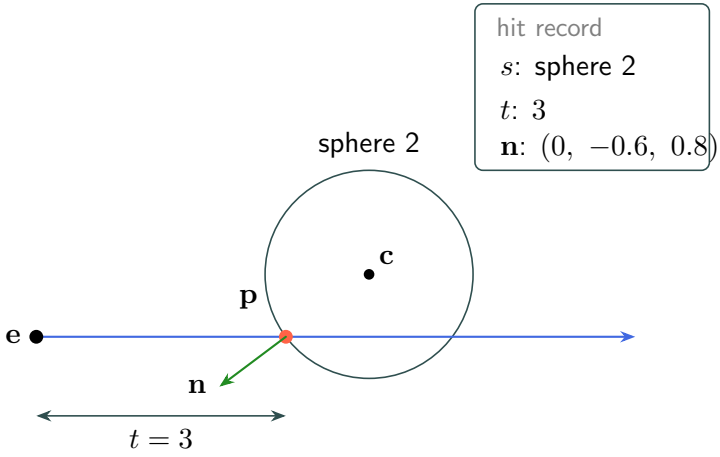
- Which surface was hit: s

A hit record holds surface, t, and normal



- Which surface was hit: s
- Where: its t along the ray

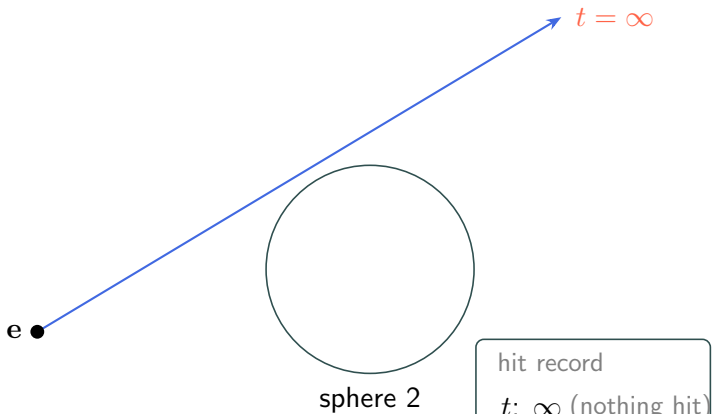
A hit record holds surface, t, and normal



- Which surface was hit: s
- Where: its t along the ray
- Which way it faces: $\mathbf{n}$

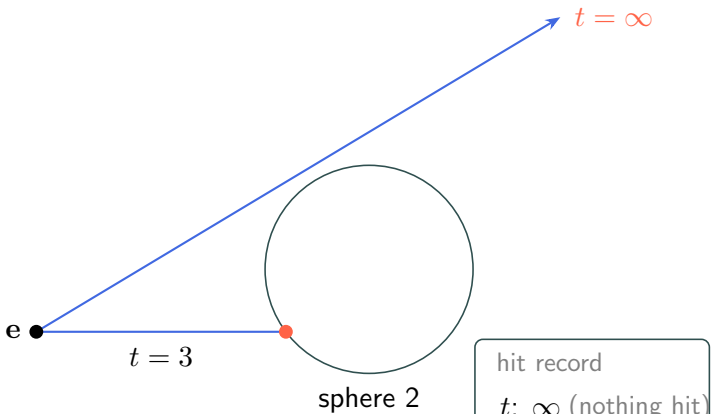
$$\mathbf{n} = (\mathbf{p} - \mathbf{c})/R$$

A miss is a record with $t = \infty$



- No hit: the record says $t = \infty$

A miss is a record with $t = \infty$



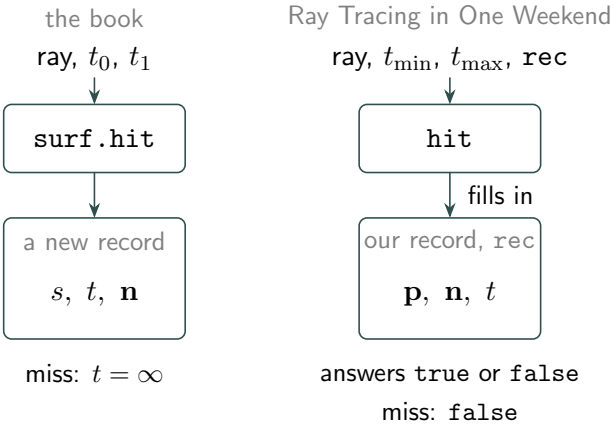
- No hit: the record says $t = \infty$
- Any real hit has a smaller t

Return a new record, or fill one in



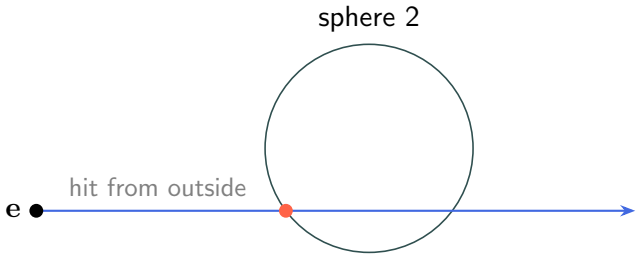
- Book: hands back a new record

Return a new record, or fill one in



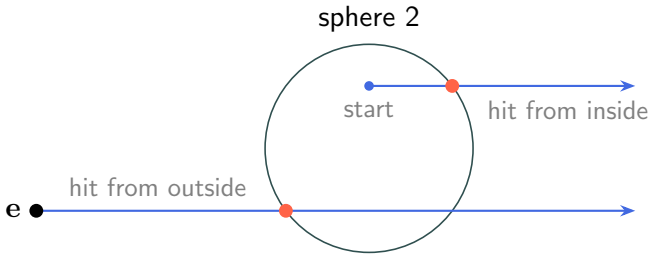
- Book: hands back a new record
- Or: fills in one we pass in

Goal: record which side the ray came from



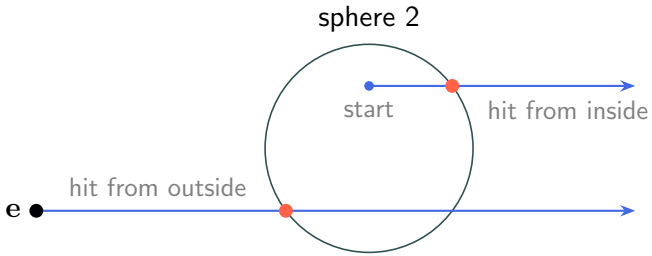
- Our ray hits sphere 2 from outside

Goal: record which side the ray came from



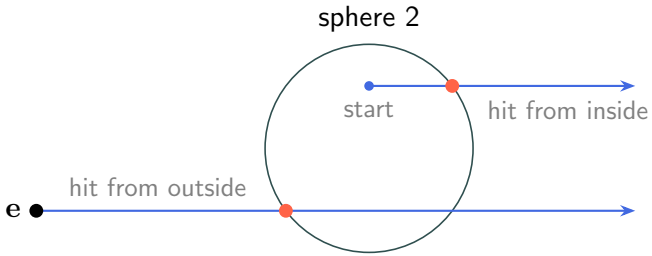
- Our ray hits sphere 2 from outside
- A ray starting inside hits from inside

Goal: record which side the ray came from



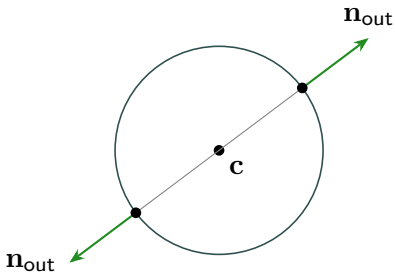
- Our ray hits sphere 2 from outside
- A ray starting inside hits from inside
- Glass (lecture 18) must know which

Goal: record which side the ray came from



- Our ray hits sphere 2 from outside
- A ray starting inside hits from inside
- Glass (lecture 18) must know which
- So the hit record stores the side

How to tell the side: the outward normal

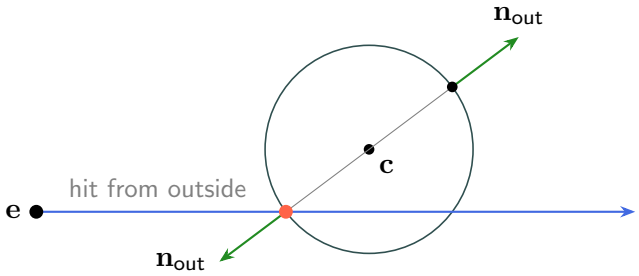


- Outward normal $\mathbf{n}_{\text{out}}$: straight out from $\mathbf{c}$

$$\mathbf{n}_{\text{out}} = (\mathbf{p} - \mathbf{c})/R$$

$\mathbf{p}$: the hit point

How to tell the side: the outward normal



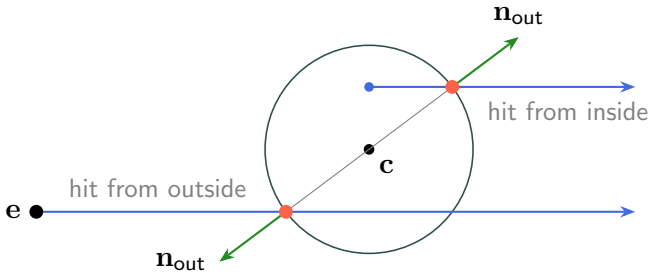
- Outward normal $\mathbf{n}_{\text{out}}$: straight out from $\mathbf{c}$

  $\mathbf{n}_{\text{out}} = (\mathbf{p} - \mathbf{c})/R$

  $\mathbf{p}$: the hit point

- From outside: it points against the ray

How to tell the side: the outward normal



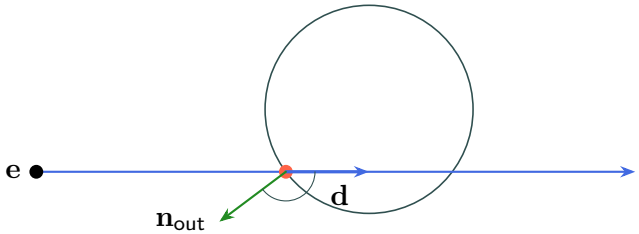
- Outward normal $\mathbf{n}_{\text{out}}$: straight out from $\mathbf{c}$

  $\mathbf{n}_{\text{out}} = (\mathbf{p} - \mathbf{c})/R$

  $\mathbf{p}$: the hit point

- From outside: it points against the ray
- From inside: it points along the ray

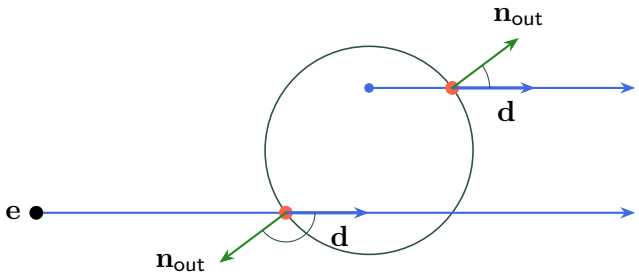
The sign of $\mathbf{d} \cdot \mathbf{n}_{\text{out}}$ tells the side



- $\mathbf{d} \cdot \mathbf{n}_{\text{out}} < 0$: from outside

$$(0, 0, -1) \cdot (0, -0.6, 0.8) = -0.8$$

The sign of $\mathbf{d} \cdot \mathbf{n}_{\text{out}}$ tells the side



- $\mathbf{d} \cdot \mathbf{n}_{\text{out}} < 0$: from outside

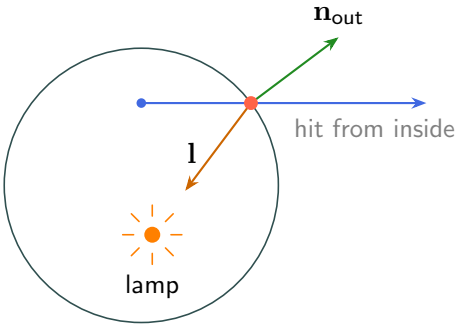
$$(0, 0, -1) \cdot (0, -0.6, 0.8) = -0.8$$

- $\mathbf{d} \cdot \mathbf{n}_{\text{out}} > 0$: from inside

$$(0, 0, -1) \cdot (0, 0.6, -0.8) = 0.8$$

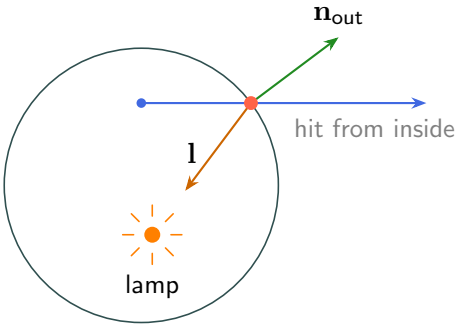
An inside hit needs the normal flipped

Lighting (lecture 16): brightness $= \max(0,\ \mathbf{n} \cdot \mathbf{l})$, $\mathbf{l}$ points to the lamp



An inside hit needs the normal flipped

Lighting (lecture 16): brightness $= \max(0,\ \mathbf{n} \cdot \mathbf{l})$, $\mathbf{l}$ points to the lamp

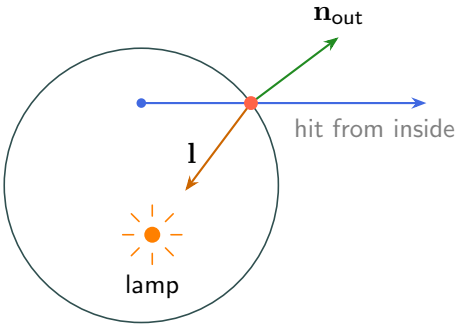


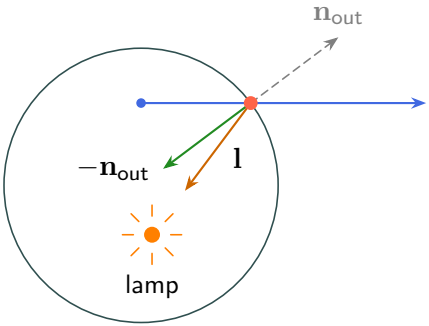
- $\mathbf{n}_{\text{out}} \cdot \mathbf{l} = -0.96$: brightness 0

  the lamp counts as behind the wall: black

An inside hit needs the normal flipped

Lighting (lecture 16): brightness $= \max(0,\ \mathbf{n} \cdot \mathbf{l})$, $\mathbf{l}$ points to the lamp





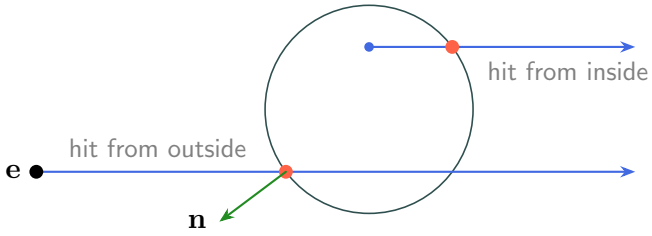
- $\mathbf{n}_{\text{out}} \cdot \mathbf{l} = -0.96$: brightness 0

  the lamp counts as behind the wall: black

- $(-\mathbf{n}_{\text{out}}) \cdot \mathbf{l} = 0.96$: the wall is lit

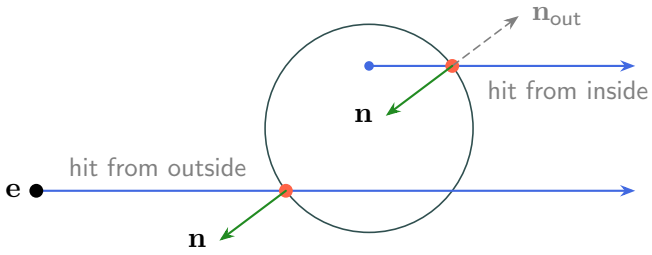
  the lamp is in front of the wall

Store the normal of the side that was hit



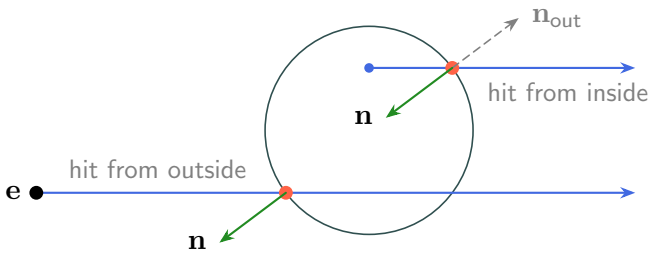
- The outside faces out:
  $\mathbf{n} = \mathbf{n}_{\text{out}}$

Store the normal of the side that was hit



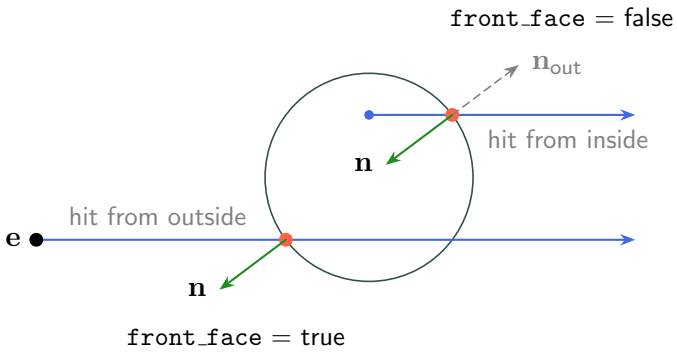
- The outside faces out:
  $\mathbf{n} = \mathbf{n}_{\text{out}}$
- The inside faces in:
  $\mathbf{n} = -\mathbf{n}_{\text{out}}$

Store the normal of the side that was hit



- The outside faces out: $\mathbf{n} = \mathbf{n}_{\text{out}}$
- The inside faces in: $\mathbf{n} = -\mathbf{n}_{\text{out}}$
- So each $\mathbf{n}$ points against its own ray

Store the normal of the side that was hit



- The outside faces out: $\mathbf{n} = \mathbf{n}_{\text{out}}$
- The inside faces in: $\mathbf{n} = -\mathbf{n}_{\text{out}}$
- So each $\mathbf{n}$ points against its own ray
- That hides the side, so store it too

$$\texttt{front_face} = (\mathbf{d} \cdot \mathbf{n}_{\text{out}} < 0)$$

Recap: the hit record and its side

hit record

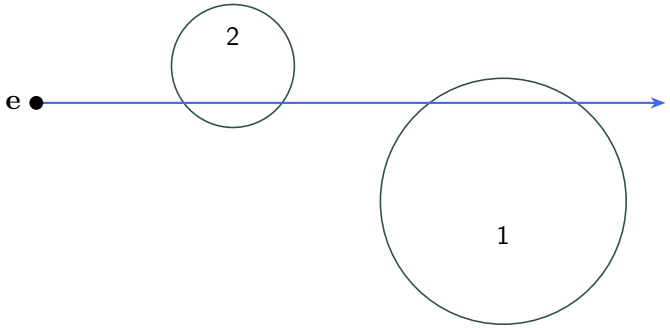
s	which surface was hit
t	where along the ray; $t = \infty$ for a miss
$\mathbf{n}$	which way the side that was hit faces
`front_face`	true: the ray came from outside

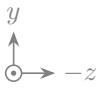
A group of surfaces

Many objects, one nearest hit

Our ray crosses two of the three spheres

side view: the slice $x = 0$



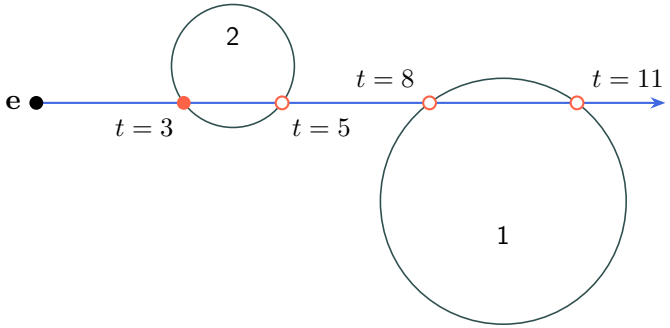


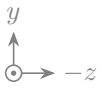
sphere 3 sits at $x = -5$: off this slice, off this ray

- Sphere 3 lies off this ray

Our ray crosses two of the three spheres

side view: the slice $x = 0$



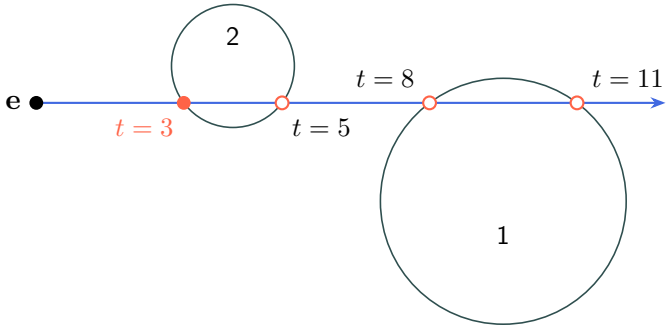


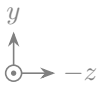
sphere 3 sits at $x = -5$: off this slice, off this ray

- Sphere 3 lies off this ray
- It crosses at $t = 3, 5, 8, 11$

Our ray crosses two of the three spheres

side view: the slice $x = 0$

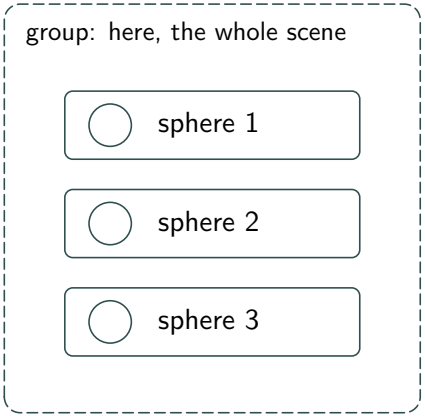




sphere 3 sits at $x = -5$: off this slice, off this ray

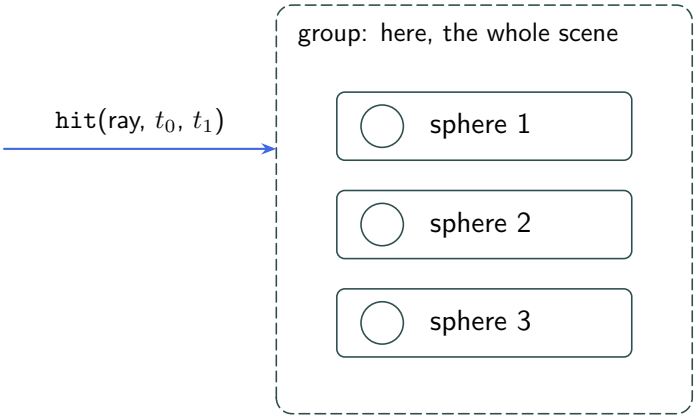
- Sphere 3 lies off this ray
- It crosses at $t = 3, 5, 8, 11$
- Lecture 10: the smallest t wins

A group of surfaces is itself a surface



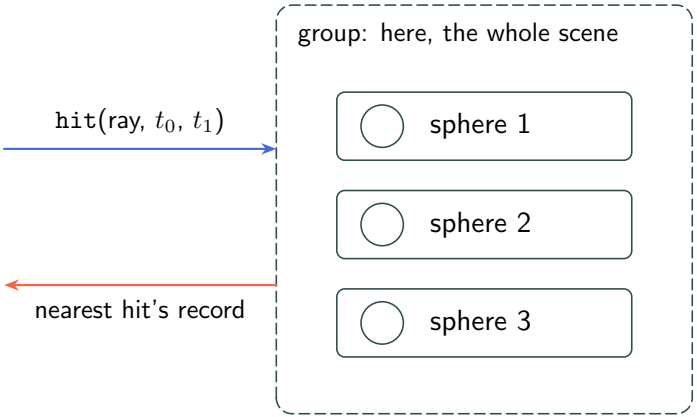
- A group: a list of surfaces

A group of surfaces is itself a surface



- A group: a list of surfaces
- Asked like any one surface

A group of surfaces is itself a surface



- A group: a list of surfaces
- Asked like any one surface
- Answers with the nearest hit

How a group finds the nearest hit

```
class Group(Surface):                          # a group is a surface too
    def __init__(self):
        self.surface_list = []                 # the list of its surfaces
    def add(self, surface):
        self.surface_list.append(surface)      # one more surface
    def hit(self, ray, t0 = 0.00001, t1 = math.inf):
```

- A group answers like any surface

How a group finds the nearest hit

```
class Group(Surface):                              # a group is a surface too
    def __init__(self):
        self.surface_list = []                     # the list of its surfaces
    def add(self, surface):
        self.surface_list.append(surface)          # one more surface
    def hit(self, ray, t0 = 0.00001, t1 = math.inf):
        closest_hit = HitRecord(None, math.inf, Vec3(0,0,0), True)
```

- A group answers like any surface

How a group finds the nearest hit

```
class Group(Surface):                              # a group is a surface too
    def __init__(self):
        self.surface_list = []                     # the list of its surfaces
    def add(self, surface):
        self.surface_list.append(surface)          # one more surface
    def hit(self, ray, t0 = 0.00001, t1 = math.inf):
        closest_hit = HitRecord(None, math.inf, Vec3(0,0,0), True)
        for surf in self.surface_list:             # ask each surface
            rec = surf.hit(ray, t0, t1)            # only hits in [t0, t1]
```

- A group answers like any surface

How a group finds the nearest hit

```
class Group(Surface):                          # a group is a surface too
    def __init__(self):
        self.surface_list = []                 # the list of its surfaces
    def add(self, surface):
        self.surface_list.append(surface)      # one more surface
    def hit(self, ray, t0 = 0.00001, t1 = math.inf):
        closest_hit = HitRecord(None, math.inf, Vec3(0,0,0), True)
        for surf in self.surface_list:         # ask each surface
            rec = surf.hit(ray, t0, t1)        # only hits in [t0, t1]
            if rec.t < math.inf:               # something was hit
                closest_hit = rec              # keep this record
```

- A group answers like any surface

How a group finds the nearest hit

```
class Group(Surface):                                  # a group is a surface too
    def __init__(self):
        self.surface_list = []                         # the list of its surfaces
    def add(self, surface):
        self.surface_list.append(surface)              # one more surface
    def hit(self, ray, t0 = 0.00001, t1 = math.inf):
        closest_hit = HitRecord(None, math.inf, Vec3(0,0,0), True)
        for surf in self.surface_list:                 # ask each surface
            rec = surf.hit(ray, t0, t1)                # only hits in [t0, t1]
            if rec.t < math.inf:                       # something was hit
                closest_hit = rec                      # keep this record
                t1 = rec.t                             # shrink the interval
```

- A group answers like any surface
- Every hit found shrinks t_1

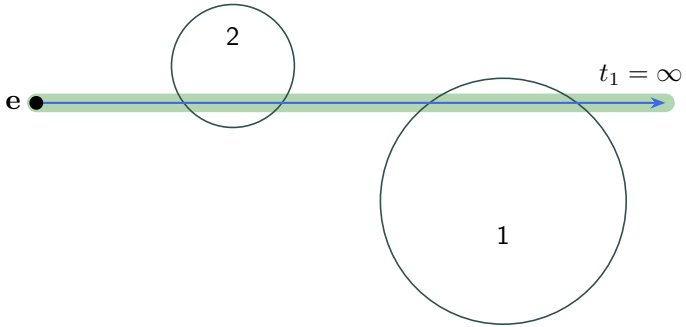
How a group finds the nearest hit

```
class Group(Surface):                          # a group is a surface too
    def __init__(self):
        self.surface_list = []                 # the list of its surfaces
    def add(self, surface):
        self.surface_list.append(surface)      # one more surface
    def hit(self, ray, t0 = 0.00001, t1 = math.inf):
        closest_hit = HitRecord(None, math.inf, Vec3(0,0,0), True)
        for surf in self.surface_list:         # ask each surface
            rec = surf.hit(ray, t0, t1)        # only hits in [t0, t1]
            if rec.t < math.inf:               # something was hit
                closest_hit = rec              # keep this record
                t1 = rec.t                     # shrink the interval
        return closest_hit                     # nearest hit, or a miss
```

- A group answers like any surface
- Every hit found shrinks t_1

Worked: t_1 shrinks to 8, then to 3

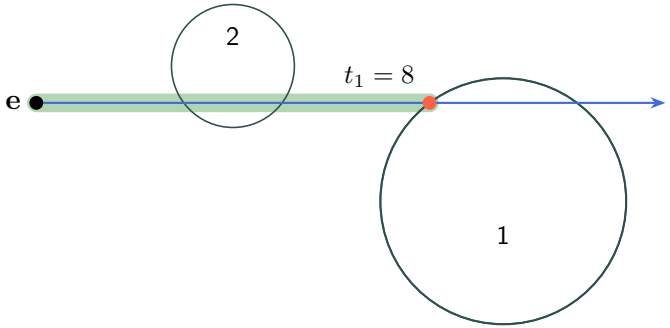
list order: 1, 2, 3



- Start: a miss, $t_1 = \infty$

Worked: t_1 shrinks to 8, then to 3

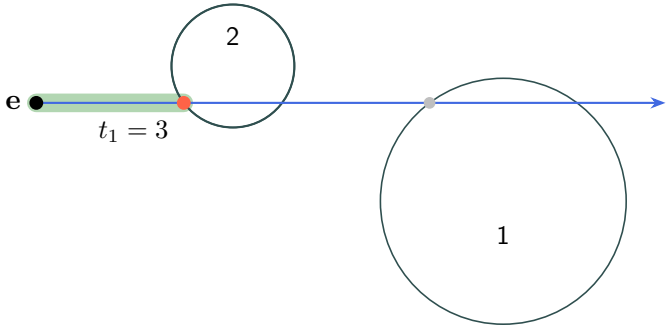
list order: 1, 2, 3



- Start: a miss, $t_1 = \infty$
- Sphere 1: hit at 8, so $t_1 = 8$

Worked: t_1 shrinks to 8, then to 3

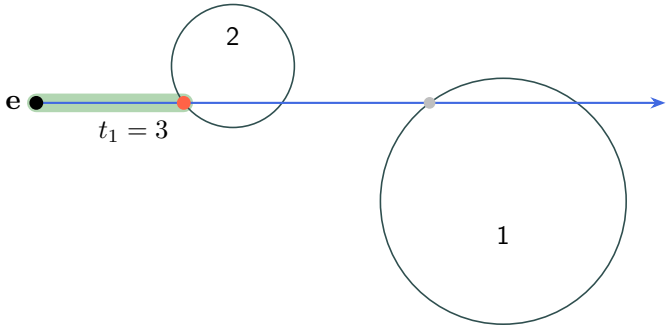
list order: 1, 2, 3



- Start: a miss, $t_1 = \infty$
- Sphere 1: hit at 8, so $t_1 = 8$
- Sphere 2: hit at 3, so $t_1 = 3$

Worked: sphere 3 misses; sphere 2's record returns

list order: 1, 2, 3

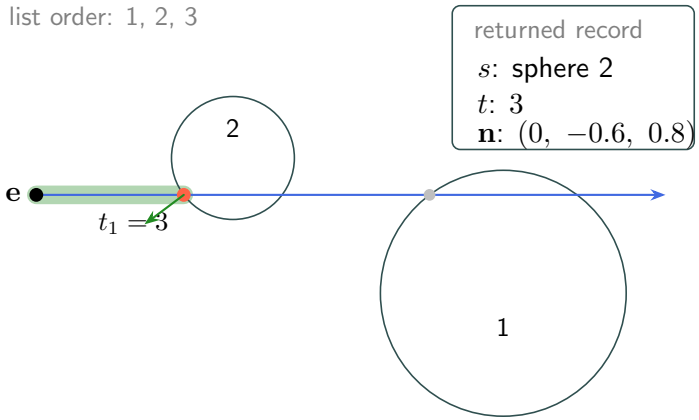


- Sphere 3: no hit, t_1 stays 3

sphere 3: no hit, $t = \infty$; t_1 stays 3

Worked: sphere 3 misses; sphere 2's record returns

list order: 1, 2, 3

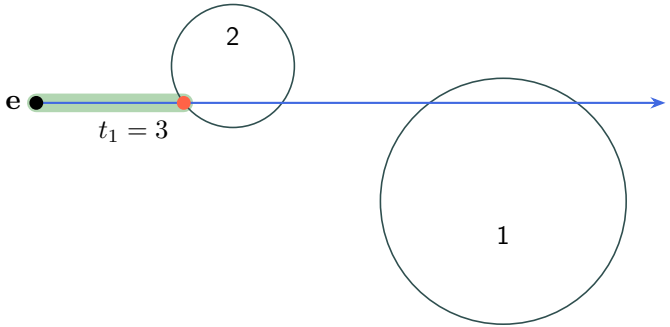


- Sphere 3: no hit, t_1 stays 3
- Return sphere 2's record

sphere 3: no hit, $t = \infty$; t_1 stays 3

After a hit, farther surfaces answer "miss"

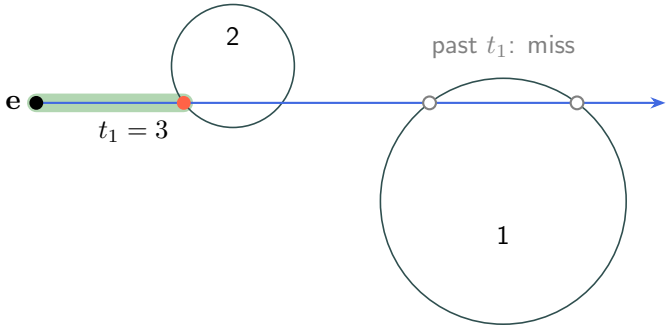
list order: 2, 1, 3



- Order 2, 1, 3: $t_1 = 3$ at once

After a hit, farther surfaces answer "miss"

list order: 2, 1, 3



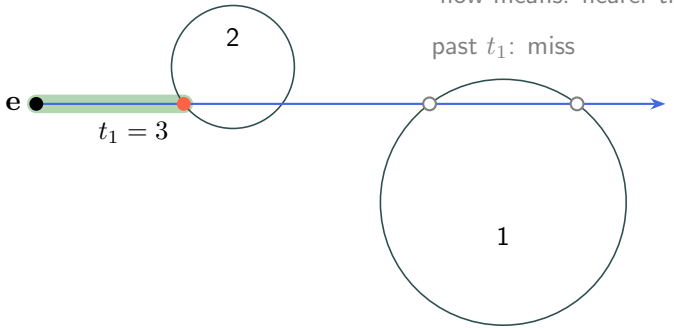
- Order 2, 1, 3: $t_1 = 3$ at once
- Sphere 1's crossings lie past t_1

After a hit, farther surfaces answer "miss"

list order: 2, 1, 3

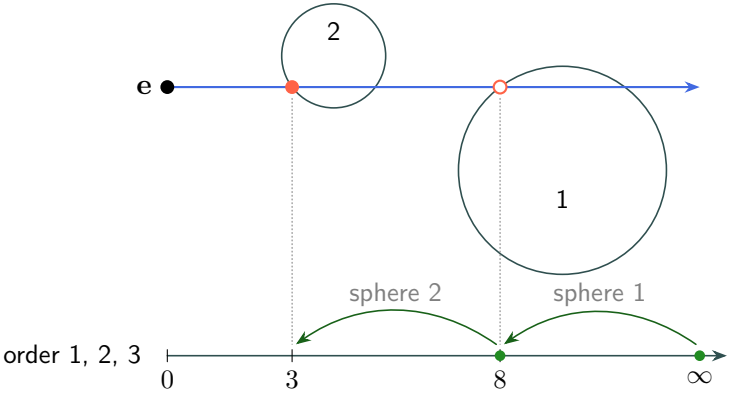
if rec.$t < \infty$

now means: nearer than all before



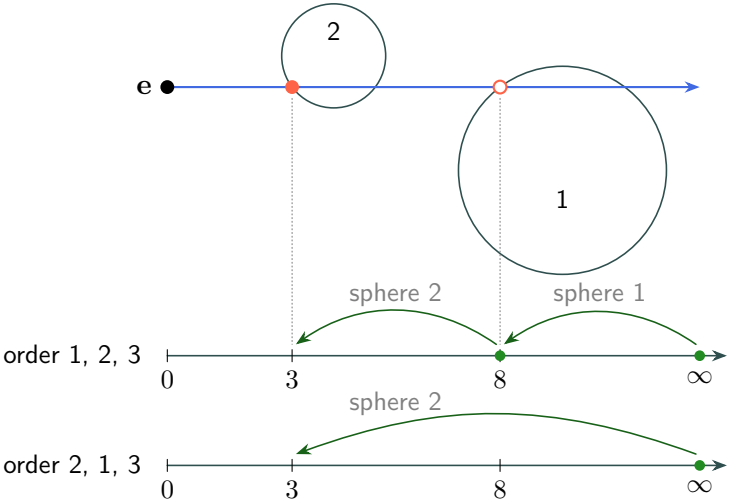
- Order 2, 1, 3: $t_1 = 3$ at once
- Sphere 1's crossings lie past t_1
- So each kept hit is nearer

Any order gives the same nearest hit



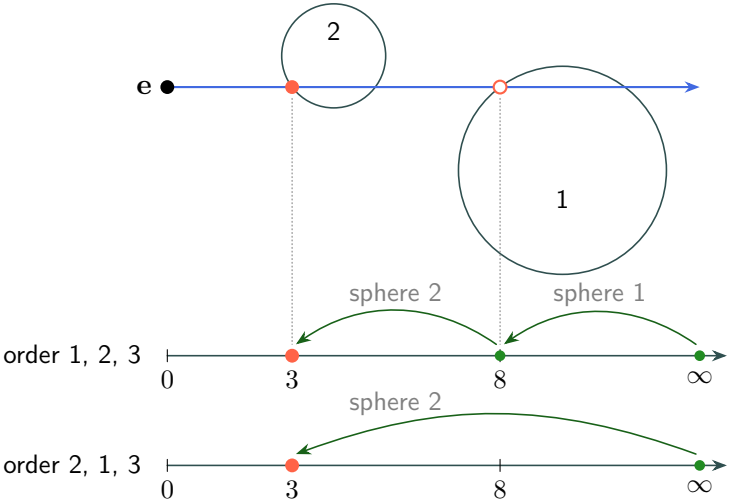
- Order 1, 2, 3: t_1 moves twice

Any order gives the same nearest hit



- Order 1, 2, 3: t_1 moves twice
- Order 2, 1, 3: t_1 moves once

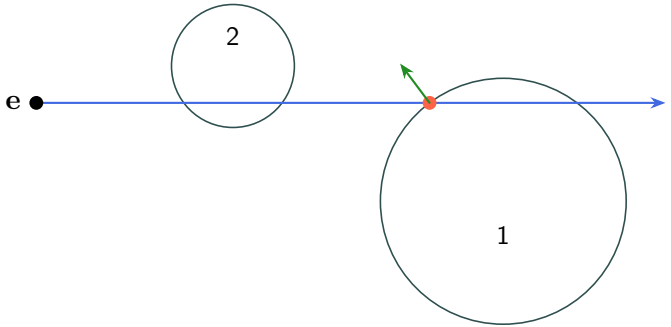
Any order gives the same nearest hit



- Order 1, 2, 3: t_1 moves twice
- Order 2, 1, 3: t_1 moves once
- Same record either way

Replaced records: their normals were wasted work

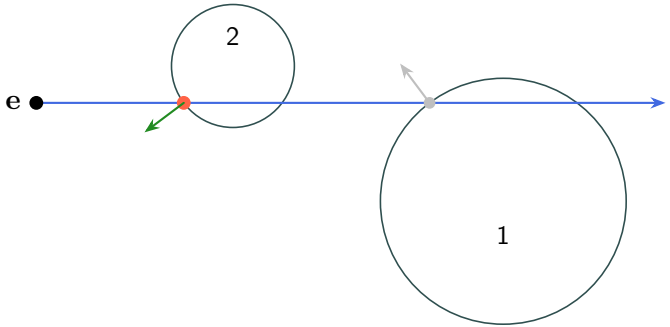
list order: 1, 2, 3



- Sphere 1's record: normal included

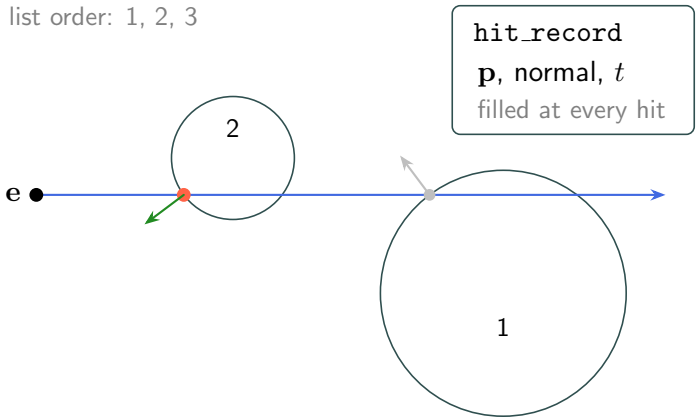
Replaced records: their normals were wasted work

list order: 1, 2, 3



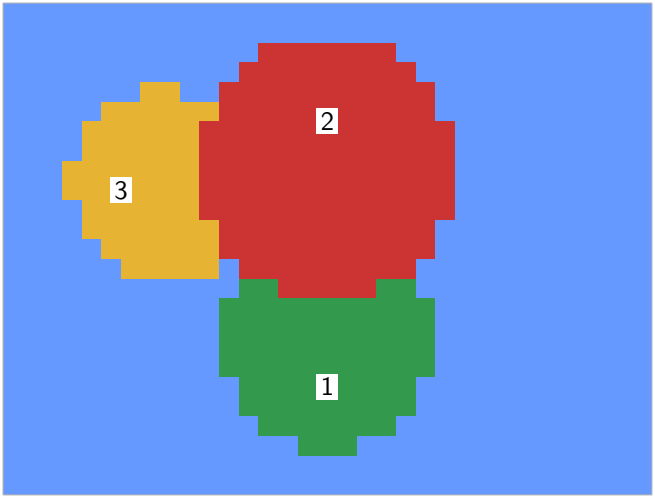
- Sphere 1's record: normal included
- Replaced: that normal went unused

Replaced records: their normals were wasted work



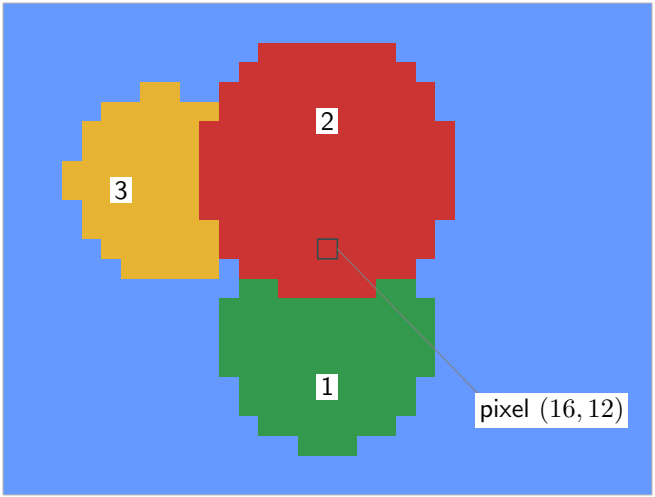
- Sphere 1's record: normal included
- Replaced: that normal went unused
- Simplest: fill it at every hit

First picture: each pixel takes its sphere's colour



- Each pixel: its nearest sphere's colour
- No hit: the sky's colour

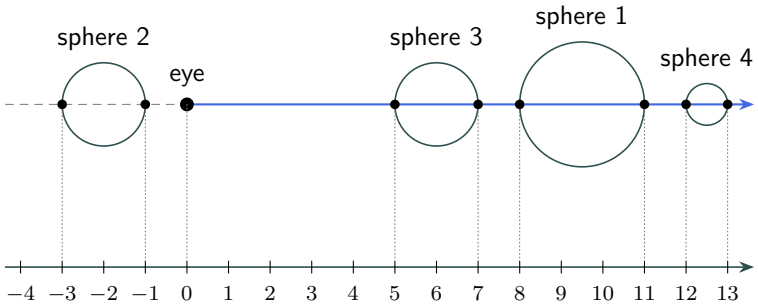
First picture: each pixel takes its sphere's colour



- Each pixel: its nearest sphere's colour
- No hit: the sky's colour
- Our ray: pixel $(16, 12)$, sphere 2

Quick check 1: trace a group's loop

one ray from the eye, four spheres on its line



a dot: the ray crosses a sphere; read its t below it

list order: 1, 2, 3, 4

start: $t_0 = 0$, $t_1 = \infty$

t_1 after each sphere:

____ ____ ____ ____

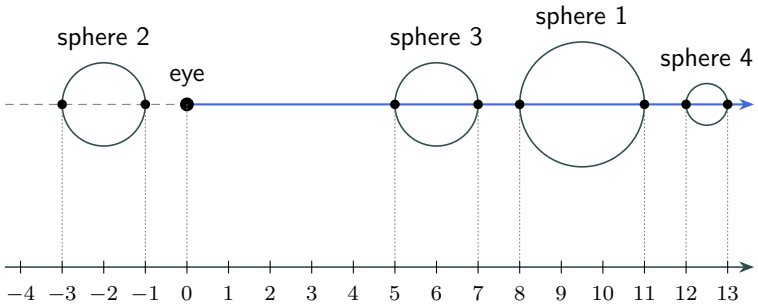
the group returns:

_____

- Ask the spheres in list order
- Write t_1 after each one

Quick check 1: trace a group's loop

one ray from the eye, four spheres on its line



list order: 1, 2, 3, 4

start: $t_0 = 0$, $t_1 = \infty$

t_1 after each sphere:

$8, \quad 8, \quad 5, \quad 5$

the group returns:

sphere 3, $t = 5$

a dot: the ray crosses a sphere; read its t below it

- Ask the spheres in list order
- Write t_1 after each one

Recap: keep each hit, shrink t_1

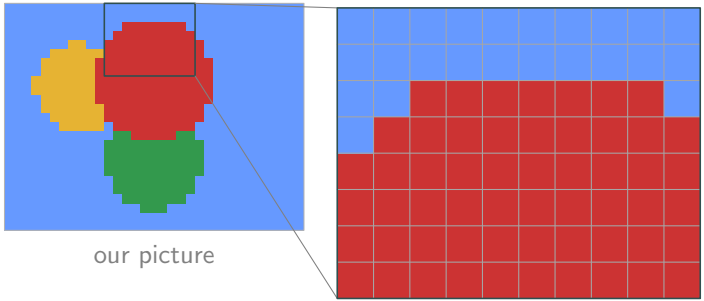


if rec.$t < \infty$: keep rec, $t_1 = \text{rec}.t$

Antialiasing

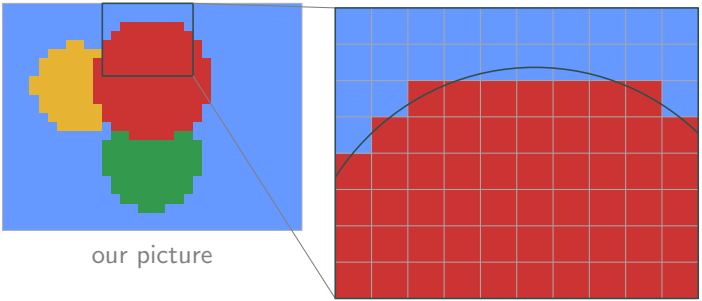
Several rays per pixel, averaged

Zoom in: our picture has stair steps



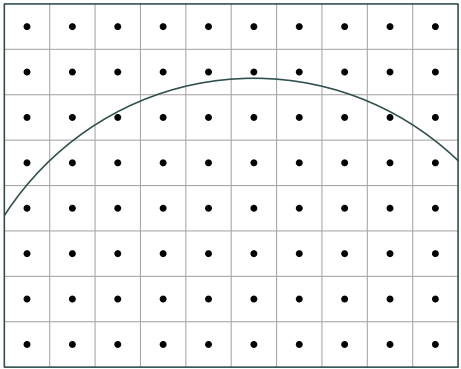
- Stair steps: aliasing, or jaggies

Zoom in: our picture has stair steps



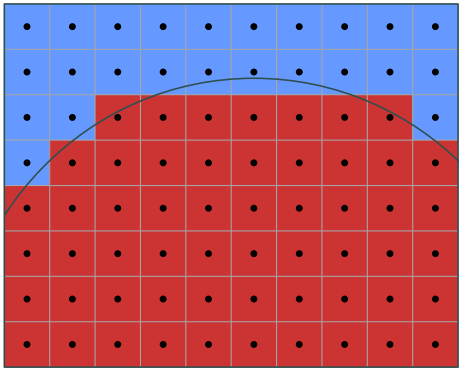
- Stair steps: aliasing, or jaggies
- The true edge: a smooth curve

One ray through the centre: a point sample



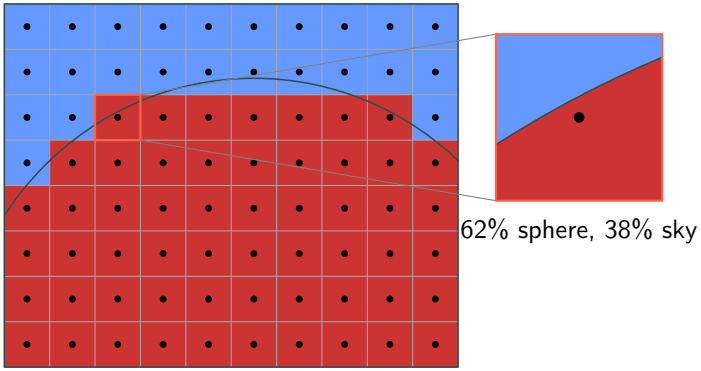
- One ray through each pixel's centre

One ray through the centre: a point sample



- One ray through each pixel's centre
- The centre decides: point sampling

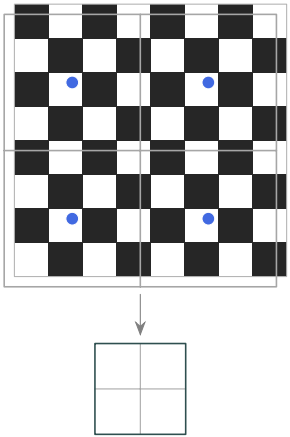
One ray through the centre: a point sample



- One ray through each pixel's centre
- The centre decides: point sampling
- This pixel: 62% sphere, all red

A far checkerboard fools four point samples

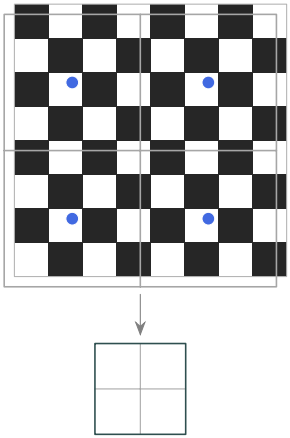
a far checkerboard, 8×8 tiles



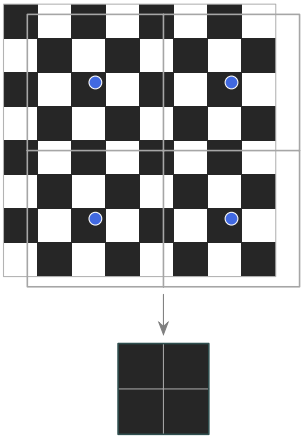
- 8×8 tiles, but only 4 rays

A far checkerboard fools four point samples

a far checkerboard, 8×8 tiles



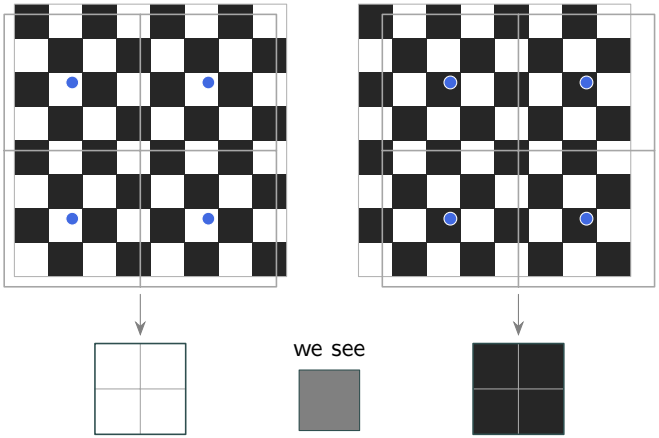
the same board, one tile left



- 8×8 tiles, but only 4 rays
- All white, or all black

A far checkerboard fools four point samples

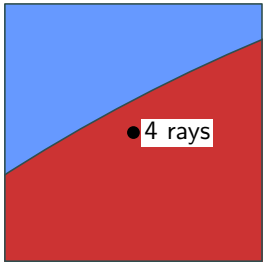
a far checkerboard, 8×8 tiles the same board, one tile left



- 8×8 tiles, but only 4 rays
- All white, or all black
- Our eyes see gray

Spread the rays over the pixel

pixel $(13, 22)$

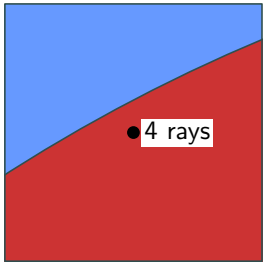


same point: sphere, 4 times

- Same point four times: same answer

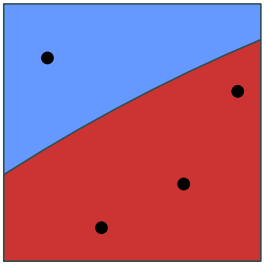
Spread the rays over the pixel

pixel $(13, 22)$



same point: sphere, 4 times

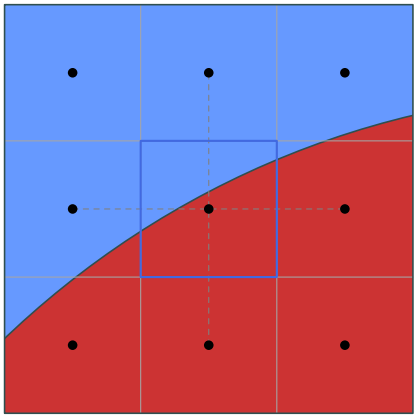
pixel $(13, 22)$



spread: 3 sphere, 1 sky

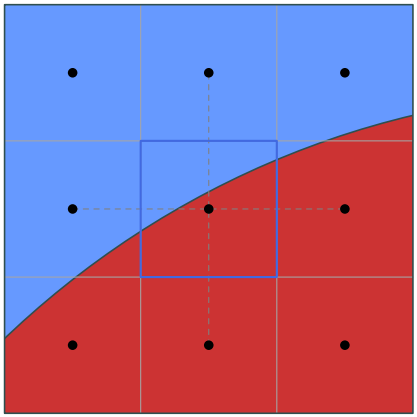
- Same point four times: same answer
- Spread out: some sphere, some sky

Sample the square around the pixel's centre



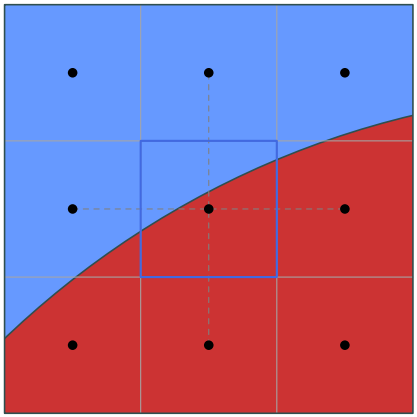
- The square: halfway to each neighbour

Sample the square around the pixel's centre



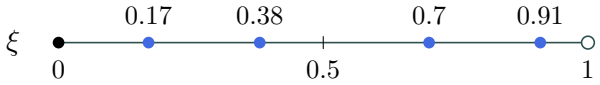
- The square: halfway to each neighbour
- Simplest choice: all its points count equally

Sample the square around the pixel's centre



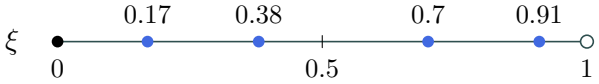
- The square: halfway to each neighbour
- Simplest choice: all its points count equally
- Better: points near the centre count more

Random numbers come from $[0, 1)$



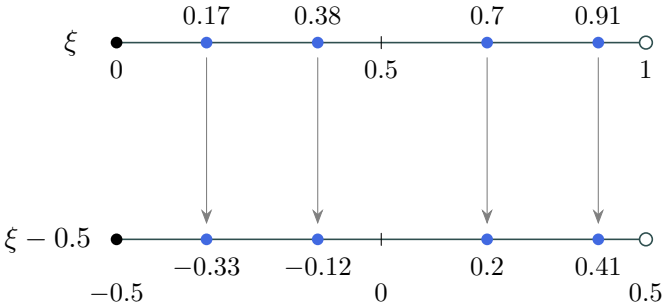
- $0 \leq \xi < 1$: 1 never comes out

Random numbers come from $[0, 1)$



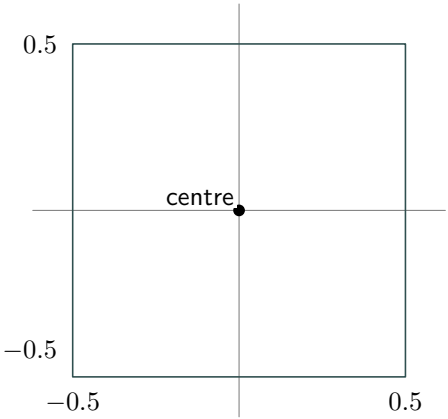
- $0 \leq \xi < 1$: 1 never comes out
- Map to any range: $\min + (\max - \min)\,\xi$

Random numbers come from $[0, 1)$



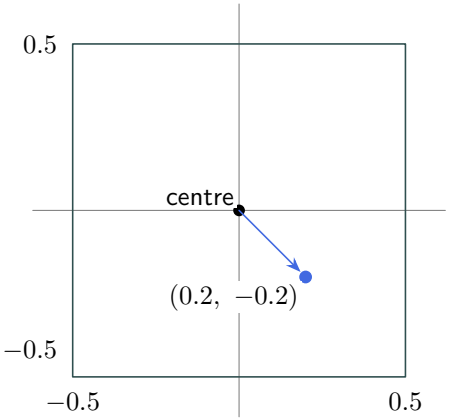
- $0 \leq \xi < 1$: 1 never comes out
- Map to any range: $\min + (\max - \min)\,\xi$
- Here: $-0.5 + \xi$, from -0.5 to 0.5

Two random numbers give an offset



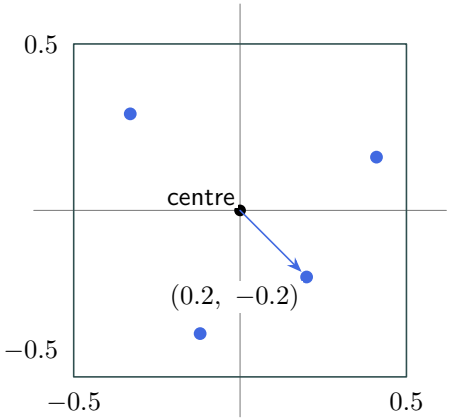
- Two random numbers, minus 0.5 each
  offset $= (\xi_1 - 0.5,\ \xi_2 - 0.5)$

Two random numbers give an offset



- Two random numbers, minus 0.5 each
  offset $= (\xi_1 - 0.5,\ \xi_2 - 0.5)$
- $\xi = (0.7,\ 0.3)$ gives $(0.2,\ -0.2)$

Two random numbers give an offset

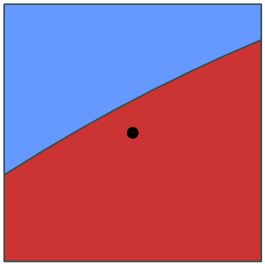


- Two random numbers, minus 0.5 each
  offset $= (\xi_1 - 0.5,\ \xi_2 - 0.5)$
- $\xi\ =\ (0.7,\ 0.3)$ gives $(0.2,\ -0.2)$
- Anywhere in the pixel's square

Sample point: pixel centre plus the offset

Lecture 13: $u = l + (i + 0.5) \times$ cell width

centre $(-0.15,\ 0.5)$

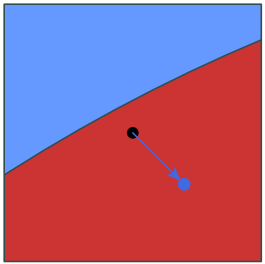


pixel $(13, 22)$; $\xi = (0.7,\ 0.3)$; $l = -0.825$, $b = -0.625$, cell 0.05

- Lecture 13: centre at $i + 0.5$

Sample point: pixel centre plus the offset

centre $(-0.15,\ 0.5)$



sample $(-0.14,\ 0.49)$

Lecture 13: $u = l + (i + 0.5) \times$ cell width

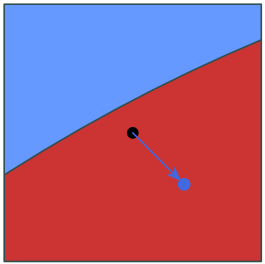
centre + offset: $(i + 0.5) + (\xi_1 - 0.5) = i + \xi_1$

pixel $(13, 22)$; $\xi = (0.7,\ 0.3)$; $l = -0.825$, $b = -0.625$, cell 0.05

- Lecture 13: centre at $i + 0.5$ • Now $i + \xi_1$: a random spot

Sample point: pixel centre plus the offset

centre $(-0.15,\ 0.5)$



sample $(-0.14,\ 0.49)$

Lecture 13: $u = l + (i + 0.5) \times$ cell width

centre + offset: $(i + 0.5) + (\xi_1 - 0.5) = i + \xi_1$

$u = l + (i + \xi_1) \times$ cell width

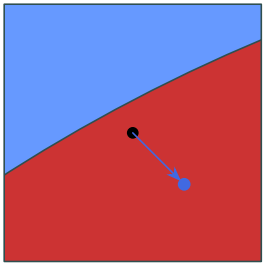
$= -0.825 + (13 + 0.7) \times 0.05 = -0.14$

pixel $(13, 22)$; $\xi = (0.7,\ 0.3)$; $l = -0.825$, $b = -0.625$, cell 0.05

- Lecture 13: centre at $i + 0.5$
- Now $i + \xi_1$: a random spot

Sample point: pixel centre plus the offset

centre $(-0.15,\ 0.5)$



sample $(-0.14,\ 0.49)$

Lecture 13: $u = l + (i + 0.5) \times$ cell width

centre + offset: $(i + 0.5) + (\xi_1 - 0.5) = i + \xi_1$

$u = l + (i + \xi_1) \times$ cell width

$= -0.825 + (13 + 0.7) \times 0.05 = -0.14$

$v = b + (j + \xi_2) \times$ cell height

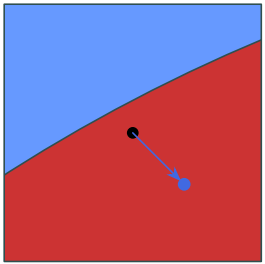
$= -0.625 + (22 + 0.3) \times 0.05 = 0.49$

pixel $(13, 22)$; $\xi = (0.7,\ 0.3)$; $l = -0.825$, $b = -0.625$, cell 0.05

- Lecture 13: centre at $i + 0.5$ • Now $i + \xi_1$: a random spot

Sample point: pixel centre plus the offset

centre $(-0.15,\ 0.5)$



sample $(-0.14,\ 0.49)$

Lecture 13: $u = l + (i + 0.5) \times$ cell width

centre + offset: $(i + 0.5) + (\xi_1 - 0.5) = i + \xi_1$

$u = l + (i + \xi_1) \times$ cell width
$= -0.825 + (13 + 0.7) \times 0.05 = -0.14$

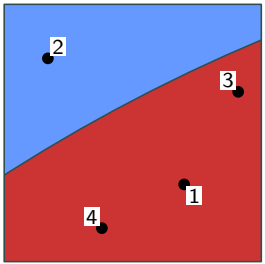
$v = b + (j + \xi_2) \times$ cell height
$= -0.625 + (22 + 0.3) \times 0.05 = 0.49$

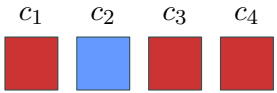
direction $(u,\ v,\ -1) = (-0.14,\ 0.49,\ -1)$

pixel $(13, 22)$; $\xi = (0.7,\ 0.3)$; $l = -0.825$, $b = -0.625$, cell 0.05

- Lecture 13: centre at $i + 0.5$
- Now $i + \xi_1$: a random spot
- Each sample: its own ray

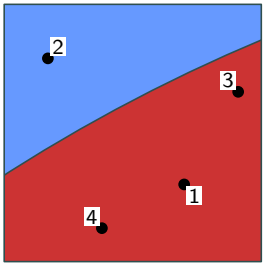
The pixel's colour: the average of N rays

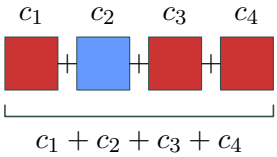




- Each sample ray runs the group loop

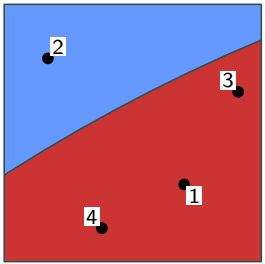
The pixel's colour: the average of N rays

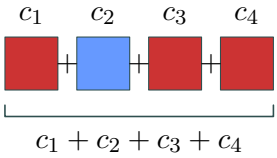




- Each sample ray runs the group loop
- Add up the N colours

The pixel's colour: the average of N rays



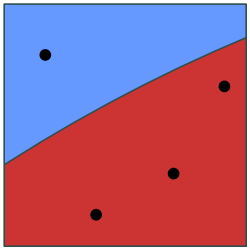


$$\text{colour} = \frac{c_1 + c_2 + \cdots + c_N}{N}$$

- Each sample ray runs the group loop
- Add up the N colours
- Divide once, at the end

Worked: three of four samples hit sphere 2

$$\text{sum} = 3 \times (0.8,\ 0.2,\ 0.2) + (0.4,\ 0.6,\ 1.0)$$

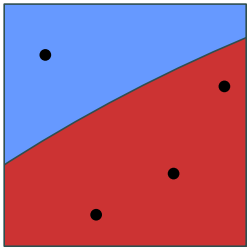


sphere 2: $(0.8,\ 0.2,\ 0.2)$

sky: $(0.4,\ 0.6,\ 1.0)$

- Three sphere colours, one sky colour

Worked: three of four samples hit sphere 2



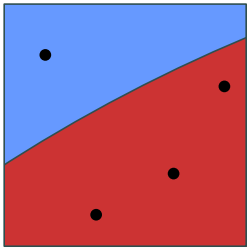
$$\text{sum} = 3 \times (0.8,\ 0.2,\ 0.2) + (0.4,\ 0.6,\ 1.0)$$
$$= (2.4,\ 0.6,\ 0.6) + (0.4,\ 0.6,\ 1.0)$$

sphere 2: $(0.8,\ 0.2,\ 0.2)$

sky: $(0.4,\ 0.6,\ 1.0)$

- Three sphere colours, one sky colour

Worked: three of four samples hit sphere 2



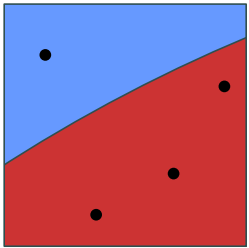
sphere 2: $(0.8,\ 0.2,\ 0.2)$

sky: $(0.4,\ 0.6,\ 1.0)$

$$\begin{aligned}\text{sum} &= 3 \times (0.8,\ 0.2,\ 0.2) + (0.4,\ 0.6,\ 1.0) \\ &= (2.4,\ 0.6,\ 0.6) + (0.4,\ 0.6,\ 1.0) \\ &= (2.8,\ 1.2,\ 1.6)\end{aligned}$$

- Three sphere colours, one sky colour

Worked: three of four samples hit sphere 2



$$\begin{aligned}\text{sum} &= 3 \times (0.8,\ 0.2,\ 0.2) + (0.4,\ 0.6,\ 1.0) \\ &= (2.4,\ 0.6,\ 0.6) + (0.4,\ 0.6,\ 1.0) \\ &= (2.8,\ 1.2,\ 1.6)\end{aligned}$$

$$\text{average} = (2.8,\ 1.2,\ 1.6)/4 = (0.7,\ 0.3,\ 0.4)$$

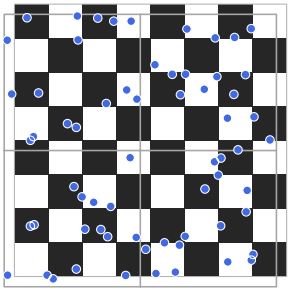
the pixel's colour

sphere 2: $(0.8,\ 0.2,\ 0.2)$

sky: $(0.4,\ 0.6,\ 1.0)$

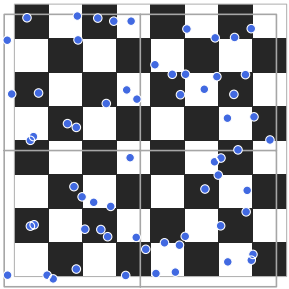
- Three sphere colours, one sky colour
- Average: $(0.7,\ 0.3,\ 0.4)$

Averaged, the far checkerboard turns gray



- 16 random samples per pixel
- About half land on white

Averaged, the far checkerboard turns gray



averaged pixels

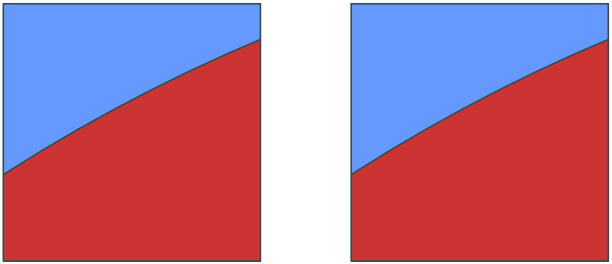
8/16	7/16
7/16	9/16

share of white samples

- 16 random samples per pixel
- About half land on white
- Average: gray, as our eyes see

More samples land closer to the true share

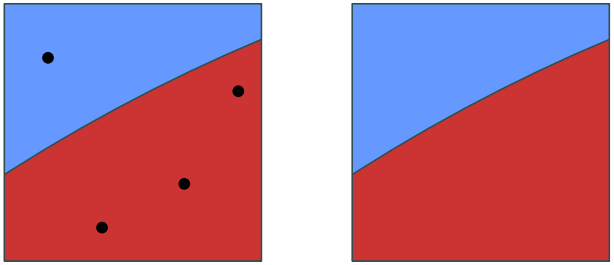
pixel $(13, 22)$: sphere 2 covers 62% of it



- Sphere 2 covers 62% of the pixel

More samples land closer to the true share

pixel $(13, 22)$: sphere 2 covers 62% of it

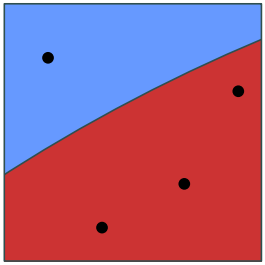


4 samples: 3 on sphere 2

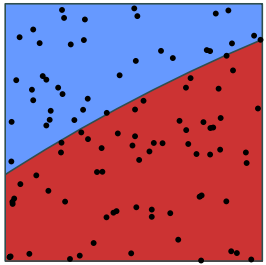
- Sphere 2 covers 62% of the pixel
- 4 samples say 75%

More samples land closer to the true share

pixel $(13, 22)$: sphere 2 covers 62% of it



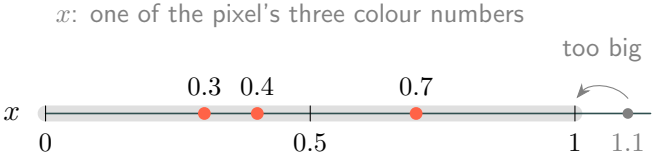
4 samples: 3 on sphere 2



100 samples: 63 on sphere 2

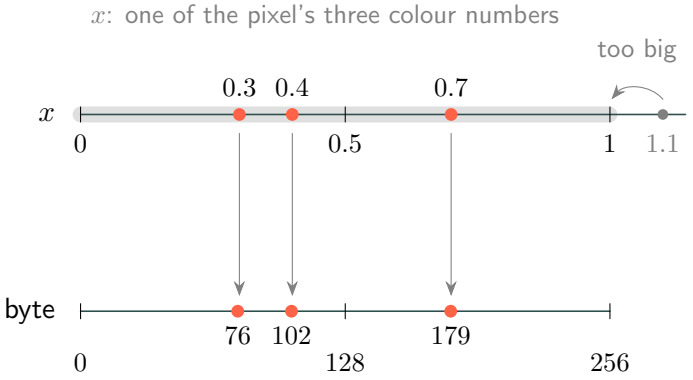
- Sphere 2 covers 62% of the pixel
- 4 samples say 75%
- 100 samples say 63%

Goal: store the pixel's colour as bytes



- Our edge pixel's average: $(0.7,\ 0.3,\ 0.4)$
- Clamp: keep each in $[0,\ 0.999]$

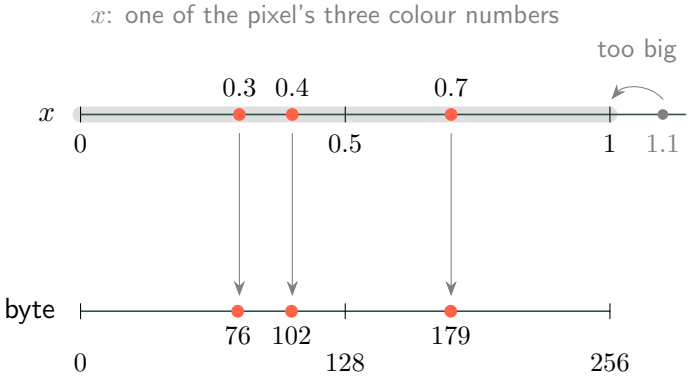
Goal: store the pixel's colour as bytes



- Our edge pixel's average: $(0.7,\ 0.3,\ 0.4)$
- Clamp: keep each in $[0,\ 0.999]$
- Byte: whole part of $256\,x$

$256 \times 0.7 = 179.2 \to 179$

Goal: store the pixel's colour as bytes

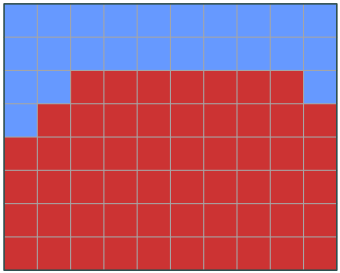


- Our edge pixel's average: $(0.7,\ 0.3,\ 0.4)$
- Clamp: keep each in $[0,\ 0.999]$
- Byte: whole part of $256\,x$

$256 \times 0.7 = 179.2 \to 179$

$(0.7,\ 0.3,\ 0.4) \to (179,\ 76,\ 102)$

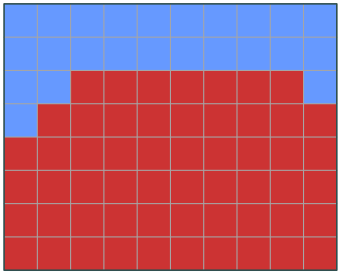
Before and after: 1 sample, 100 samples



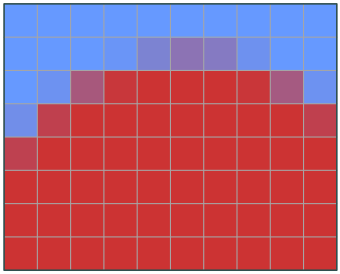
1 sample per pixel

- Same scene, same pixels

Before and after: 1 sample, 100 samples



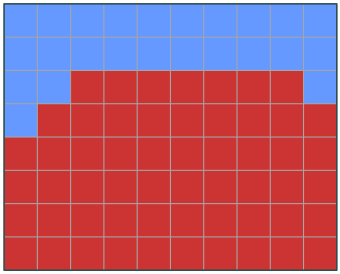
1 sample per pixel



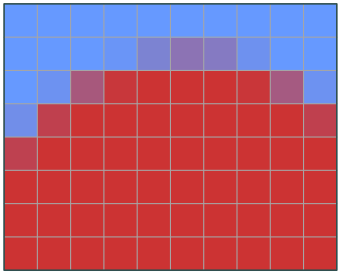
100 samples per pixel

- Same scene, same pixels
- Edges now blend sphere and sky

Before and after: 1 sample, 100 samples



1 sample per pixel



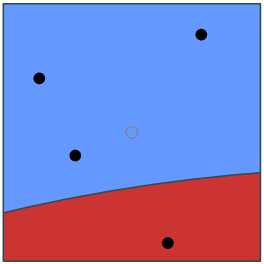
100 samples per pixel

100 times the rays

- Same scene, same pixels
- Edges now blend sphere and sky
- Cost: 100 rays per pixel

Quick check 2: average four samples

pixel $(15, 23)$



sphere 2: $(0.8,\ 0.2,\ 0.2)$
sky: $(0.4,\ 0.6,\ 1.0)$

Its four samples: 1 on sphere 2, 3 on the sky.

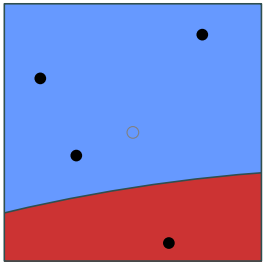
1. The pixel's colour, averaged?

2. With one ray, through the centre (circle)?

- Add four colours, then divide by 4
- The centre alone: all or nothing

Quick check 2: average four samples

pixel $(15, 23)$



sphere 2: $(0.8,\ 0.2,\ 0.2)$
sky: $(0.4,\ 0.6,\ 1.0)$

Its four samples: 1 on sphere 2, 3 on the sky.

1. The pixel's colour, averaged?

$$(0.8,\ 0.2,\ 0.2) + 3 \times (0.4,\ 0.6,\ 1.0) = (2.0,\ 2.0,\ 3.2)$$

$$(2.0,\ 2.0,\ 3.2)/4 = (0.5,\ 0.5,\ 0.8)$$

2. With one ray, through the centre (circle)?

the centre sees the sky: $(0.4,\ 0.6,\ 1.0)$

- Add four colours, then divide by 4
- The centre alone: all or nothing

Recap: a pixel is the average of its samples

$$\text{colour} = \frac{c_1 + c_2 + \cdots + c_N}{N}$$

sample k goes through $(i + \xi_1,\ j + \xi_2),\ \ 0 \leq \xi < 1$

Summary

1 A hit is reported as a record

$s,\ t,\ \mathbf{n}$; a miss has $t = \infty$

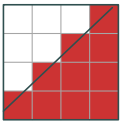
outside when $\mathbf{d} \cdot \mathbf{n}_{\text{out}} < 0$

2 A group is a surface too

if rec.$t < \infty$: keep rec, $t_1 = \text{rec}.t$

any order: the nearest hit comes back

3 One ray per pixel: a point sample



stair steps along edges

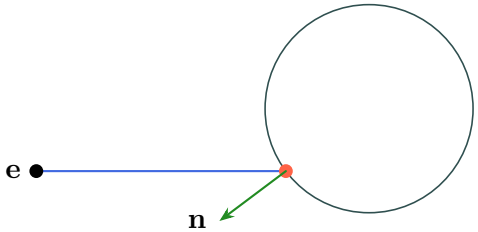
fine detail can come out wrong

4 Antialiasing: average N samples

sample at $(i + \xi_1,\ j + \xi_2),\ \ 0 \le \xi < 1$

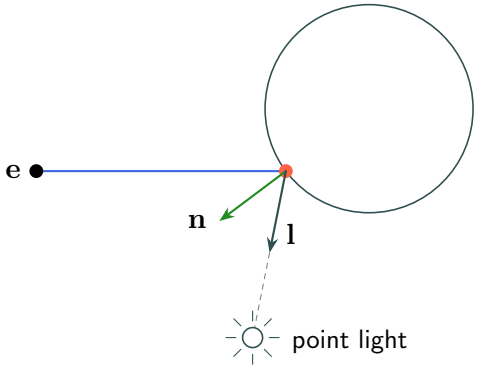
$\text{colour} = (c_1 + c_2 + \cdots + c_N)/N$

Next lecture: light the nearest hit



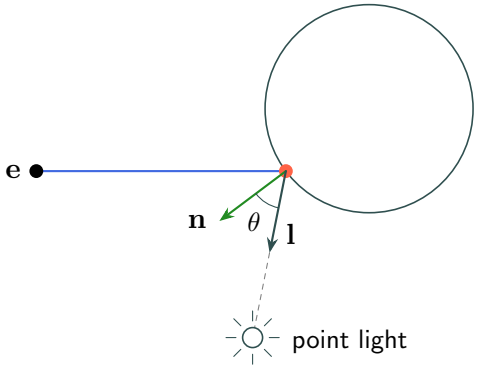
- Light the nearest hit

Next lecture: light the nearest hit



- Light the nearest hit
- A point light, in direction $\mathbf{l}$

Next lecture: light the nearest hit



- Light the nearest hit
- A point light, in direction $\mathbf{l}$
- Brightness from $\mathbf{n} \cdot \mathbf{l}$

Exit Ticket

Today's question *(2 minutes)*

A group lists spheres 1, 2, 3. One ray crosses sphere 1 at $t = 6$ and 9, sphere 2 at 2 and 4, sphere 3 at 7 and 8. Start: $t_0 = 0$, $t_1 = \infty$.

(a) Give t_1 after each sphere. Which record is returned?

(b) Of a pixel's 4 samples, 2 see green $(0.2, 0.6, 0.3)$ and 2 see sky $(0.4, 0.6, 1.0)$. The pixel's colour?

CS 453 — Exit Ticket #15

Name: _____

Roll #: _____ Date: _____

Answer:

Hand it to me on your way out.
No name, no attendance.