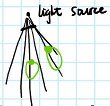
Ray Tracing

2021年8月3日 20:09

§0 Shadows

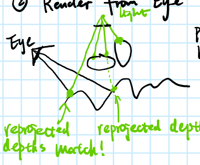
— how to draw shadows using rasterization
→ Shadow mapping
Key idea:
the points NOT in shadow must be seen by the light and by the camera.

① Render from Light.



记录看到点的深度

② Render from Eye & Project to light



project visible points in eye view
back to light source.
if reprojected depth is the same, the point is not in Shadow
Otherwise is.

点光源会导致点光源，只能做 hard shadows

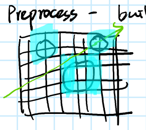
- Hard shadows
- Quality depends on shadow map resolution.
- Involve comparison of floating point depth.

Soft shadow: 光源有一定大小

§1 Whitted Style Ray Tracing

1. Grid

Preprocess - build grid



rays step through grid
For each grid cell
test intersection with all objects stored in that cell
Remember a fact: intersection with boxes is easier to compute

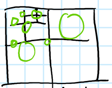
How to decide grid resolution.
too many cells → also inefficient

Heuristic
$\# \text{cells} = C * \# \text{objs}$
$C \approx 27$ in 3D

grids work well on large collection of objects
work bad when objects are loosely distributed in a large area.

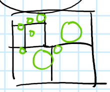
2. Spatial Partitions
improve efficiency on Grids.
examples:

Oct-Tree

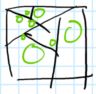


rarely-used when dimension is high

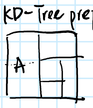
KD-Tree



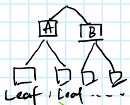
BSP-Tree



KD-Tree preprocessing



Data structure:
Internal nodes:
split axis: x, y, or z
split position.
children nodes

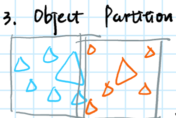


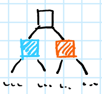
Leaf nodes:
list of objects
Objects aren't stored in internal nodes

类似二分查找，沿着 KD 不断判定求交

KD-Tree's problem: hard to decide whether triangles have intersection with boxes
some objects may appear in multiple boxes, low efficiency

3. Object Partition & Bounding Volume Hireachy (BVH)





at last: objects can only appear in one box
(Bounding boxes can interfere with each other)

How to subdivide a node:
Heuristic
① choose longest axis in node
② Split node at location of median object

§ Radiometry

Measurement for illumination.
Accurately measure spatial properties of light
→ Radiant flux, intensity, irradiance, radiance

Measurement for Illumination.
Accurately measure spatial properties of light
→ Radiant flux, intensity, irradiance, radiance

1. Radiant Energy and Flux (Power)

Energy: Q

Flux: $\Phi = \frac{dQ}{dt}$ (lm, lumen)

2. Radiant Intensity = power per unit solid angle.

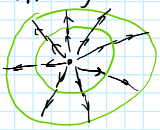
$I(\omega) = \frac{d\Phi}{d\omega}$ ($\frac{lm}{sr}$ = cd = candela)

3. Irradiance

power per unit area.

$E(x) = \frac{d\Phi(x)}{dA}$ ($\frac{lm}{m^2}$ = lux)

$E = \frac{\Phi}{A}\cos\theta$.

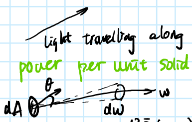


intensity won't decrease
irradiance fall off as dist. ↑

4. Radiance

distribution of light
radiance is associated with a ray
rendering is all about computing radiance

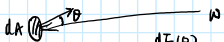
light travelling along a ray
power per unit solid angle per projected area



$L(p,\omega) = \frac{d^2\Phi(p,\omega)}{d\omega\, dA\cos\theta}$

$[\frac{cd}{m^2} = \frac{lm}{sr\cdot m^2} = nit]$

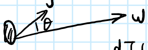
5. Incident Radiance



Irradiance per unit solid angle arriving at the surface.

$L(p,\omega) = \frac{dE(p)}{d\omega\cos\theta}$

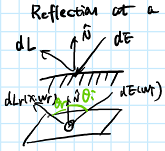
Exiting Radiance



$L(p,\omega) = \frac{dI(p,\omega)}{dA\cos\theta}$

6. BRDF

Reflection at a Point



Irradiance incoming:
$dE(\omega_i) = L(\omega_i)\cos\theta_i\, d\omega_i$ (definition)
radiance exiting:
$dL_r(\omega_r)$

BRDF:

texture ← $f_r(\omega_i \to \omega_r) = \frac{dL_r(\omega_r)}{dE_i(\omega_i)} = \frac{dL_r(\omega_r)}{L_i(\omega_i)\cos\theta\, d\omega_i}$ $[\frac{1}{sr}]$

⇒ Reflection Equation.

$L_r(p,\omega_r) = \int_{H^2} f_r(p,\omega_i \to \omega_r)\, L_i(p,\omega_i)\cos\theta_i\, d\omega_i$

Note that $L_i(p,\omega_i)$ can be some kind of reflected radiance from other points
⇒ Recursive.

7. Rendering Equation.

$L_o(p,\omega_o) = \underbrace{L_e(p,\omega_o)}_{\text{自发光}} + \underbrace{\int_{\Omega^+} L_i(p,\omega_i) f_r(p,\omega_i,\omega_o)(n\cdot\omega_i)\, d\omega_i}_{\text{反射光}}$

8. Linear Operator Equation

$L = E + KL$ 用算符表示

$l(u) = e(u) + \int l(v)\, \boxed{K(u,v)\, dv}$
kernel of equation

$\Rightarrow L = (I + K + K^2 + \cdots)E$

$L = E + KE + K^2E + K^3E + \cdots$
↓ direct Emission
↓ Direct illumination on surfaces
↓ One bounce
↓ Two bounce

9. Probability

$X \sim p(x)$

$Y = f(x)$

$E[Y] = E[f(x)] = \int f(x)\, p(x)\, dx$

§ Monte Carlo and Path Integration.

1. Monte-Carlo Integration.: Solve integration problem

$\int_a^b f(x)\,dx$

Random Variable $X_i \sim p(x)$

estimator: $\boxed{F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}}$

what kind of p(x) is the best?

Random Variable $X_i \sim p(x)$

estimator: $\boxed{F_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}}$

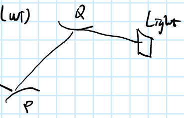
what kind of $p(x)$ is the best?

2. Path tracing

a simple Monte Carlo Solution

```
shade(p, wo)
    Randomly choose N directions wi~pdf
    Lo = 0.0
    For each wi
        Trace a ray r(p, wi)
        If ray r hit the light
            Lo += (1/N) * Li * f_r * cosine / pdf(wi)
        Else if ray r hit an object q
            Lo += (1/N) * shade(q, -wi)
                  * f_r * cosine / pdf(wi)
    Return Lo
```

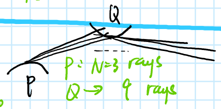
recursive.



Problem 1: too many rays!

\# rays = $N^{\#\text{bounces}}$

↳ sample times in Monte-Carlo



P: N=3 rays

Q → 9 rays

Solution: N=1 how to solve the noisy problem?

Path Tracing

Problem 2: recursive algorithm don't stop!

Solution: Russian Roulette (RR)

with probability P, shoot a ray and return the shading result divided by P: Lo/P

with probability 1-P, don't shoot a ray and got 0.

} expectation is still Lo

```
shade(p, wo):
    Manually pick a P_RR.
    Select ksi in a uniform dist in [0,1]
    If (ksi > P_RR) return 0.0;
```

bounce 次数依靠设定 P_RR 给出.

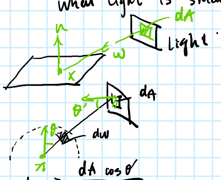
$E(\text{bounce}) = \sum_{i=1}^{\infty} i(1-P\_RR) \cdot (P\_RR)^{i-1} =$

3. Ray Generation:

```
ray_generation(camPos, pixel):
    Uniformly choose N sample positions within the pixel
    pixel_radiance = 0.0
    For each sample:
        shoot a ray r(camPos, cam_to_sample)
        If ray hit the scene at P
            pixel_radiance += 1/N * shade(p, sample_to_cam)
    Return pixel_radiance
```

4. Sampling the Light

when light is small, it's hard to reach the light (for rays)



$dw = \frac{dA \cos\theta'}{\|x'-x\|^2}$

Assume uniformly $\boxed{\text{sampling on the light}}$

$pdf = 1/A$

Thus

$L_o = \int L_i \cdot f_r \cdot \cos \cdot dw$

$\Rightarrow \int F(A)\, dA$

$= \int L_i \cdot f_r \cdot \frac{\cos\theta \cos\theta'}{\|x'-x\|^2} dA$

Thus, Radiance has 2 parts

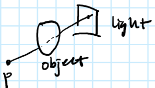
① light source (direct, sample on light, no need for RR)

② other reflections

$L\_dir = L\_i * f\_r * \cos\theta * \cos\theta' / |x'-p|^2 / pdf\_light$

$= \frac{1}{A}$ when sampling uniformly

What if ray is blocked? ⇒ additional algorithm to decide whether light is blocked.



5. What's next?

- Photon mapping
- Metropolis light transport
- VCM / UBP

6. How to sample any function $f(x)$? (Uniformly distribute on the hemisphere) Cube? Polygon?

What's best pdf for Monte Carlo integration

How to generate low discrepency random numbers?

Pixel reconstruction filter

Is radiance of a pixel the color? (gamma correction, curves, color space)