CS 6958
LECTURE 13
INTRO TO ANTI-ALIASING, PATHTRACING

March 3, 2014
What is Aliasing?

- Imagine this is our scene

Source: Physically Based Rendering, Pharr & Humphreys
What is Aliasing?

- If we render it with our ray tracers...
What is Aliasing?

- If we render it with our ray tracers…

Source: Physically Based Rendering, Pharr & Humphreys
What is Aliasing?

- Consider our Cornell Box
What is Aliasing?

- Consider our Cornell Box
What is Aliasing?

- Consider our Cornell Box
Anti-Aliasing

- Issue:
  - Our sampling can’t reconstruct original signal
  - Nyquist-Shannon sampling theorem

- Solution:
  - better sampling
  - better reconstruction of original signal

- Lots of literature and theory on sampling and reconstruction techniques
Sampling Techniques

- currently: through center of each pixel
- 1 sample not enough! Use many and “average”

- how do we pick samples?
  - uniform
  - random
  - jittered
Consider a 2x2 image
Pixel centers

(-0.5, -0.5)  
(-0.5, 0.5)  
(-0.5, 0.5)  
(0.5, 0.5)
Pixel centers

foreach pixel ($i$, $j$)

\[
x = 2.0f \times \frac{(j - xres/2.f + 0.5f)/xres;}{y = 2.0f \times \frac{(i - yres/2.f + 0.5f)/yres;}{-0.5, -0.5)}\]

This is our current ray generation method
Sampling a pixel

- pick location other than center
  - or perturb center
- stop when n reached
Generating pixel samples

inv_width = loadf(0, 2)
inv_height = loadf(0, 5)
Generating pixel samples

// value between 0, 1
x_off = (trax_rand() - .5f) * 2.f;
y_off = (trax_rand() - .5f) * 2.f;

// scale offsets to pixel coords
x_off *= inv_width;
y_off *= inv_height;

// offset from center
x += x_off;
y += y_off;

// generate ray
camera.makeRay(ray, x, y);
Different sampling strategies

uniform

Source: Physically Based Rendering, Pharr & Humphreys
Different sampling strategies

uniform

random

Source: Physically Based Rendering, Pharr & Humphreys
Different sampling strategies

uniform

random

jittered

Source: Physically Based Rendering, Pharr & Humphreys
Different sampling strategies

uniform

random

jittered

Difference between random and jittered, is that jittered has only 1 sample within each cell. It’s still randomly perturbed!

Source: Physically Based Rendering, Pharr & Humphreys
Reconstruction filters

- once we have samples, must recombine to get final color

- most popular
  - box
  - triangle
  - Gaussian
foreach pixel

for(number_samples)
    x, y = getNewSampleLocation(pixel);
camera.makeRay(ray, x, y);

    bvh.intersect(hit, ray);
    result += shade(...); // box filter

result /= number_samples; // box filter
image.set(pixel, result);
Global Illumination

- so far, we’ve looked at
  - point lights (no area)
  - direct illumination (no inter-reflections)
  - non-physical Phong shading
  - fake ambient light term

- our world behaves a little differently
- we have all the tools to get it (mostly) right!
Global Illumination
Global Illumination
Let’s try to derive path tracing

- Board time!

- Slides online will include this info
Path tracing algorithm

... 
for(number_samples)
    attenuation = Color(1.f, 1.f, 1.f);
    ray = generateNewRay( ... );
    while(depth < max_depth) {
        HitRecord hit;
        bvh.intersect(hit, ray);
        result += shade(...) * attenuation; // box filter
        attenuation *= mat_color;
        r = hemiRay(...);
        depth++
    }

result /= number_samples;       // box filter

// tone map!
image.set(pixel, result);
End