CSC 433/533 Computer Graphics

Joshua Levine
josh@email.arizona.edu

Lecture 23 Ray Tracing 4

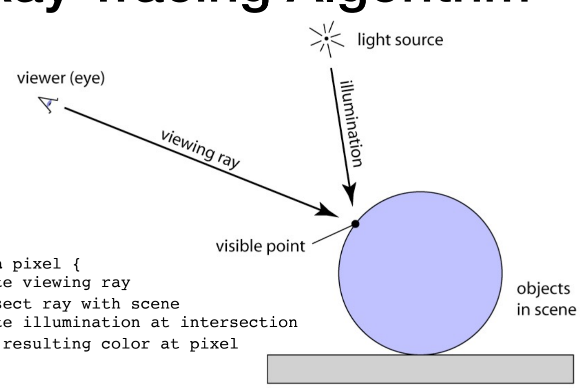
Oct. 7, 2019

Today's Agenda

- Reminders:
  - A03 questions?
  - WARNING: Midterm in two weeks!
- Goals for today:
  - Wrap up discussion of ray tracing

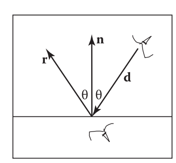
Last Time

Ray Tracing Algorithm

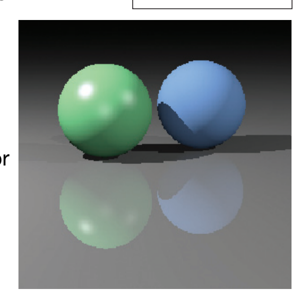



```
for each pixel {
  compute viewing ray
  intersect ray with scene
  compute illumination at intersection
  store resulting color at pixel
}
```

Reflection

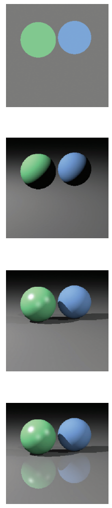


- Ideal specular reflection, or mirror reflection, can be modeled by casting another ray into the scene from the hit point
- Direction $\mathbf{r} = \mathbf{d} - 2(\mathbf{d} \cdot \mathbf{n})\mathbf{n}$
- One can then recursively accumulate some amount of color from whatever object this hits
- color $c = c + k_m$ `*ray_cast()`



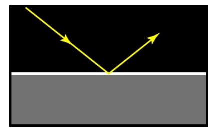
Recursive Ray Tracer

```
Color ray_cast(Ray ray, SurfaceList scene, float near, float far) {
  ...
  //initialize color;  compute hit_surf, hit_position;
  ...

  if (hit_surf is valid) {
    color = hit_surf.kA * Ia;

  }

  return color;
}
```

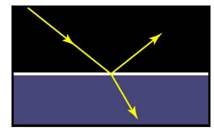


Simple materials

metal



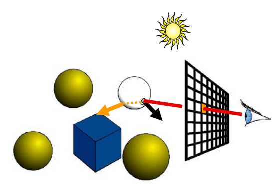
dielectric



Illuminating Dieletrics

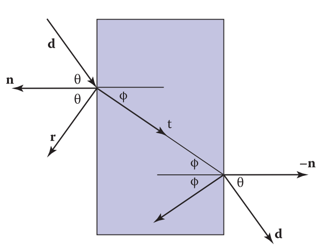
Translucency (Refraction)

- When a ray hits a dielectric surface, some portion of it transmits through the surface, but bends
- Color of the ray can be modulated by a refraction color



Snell's Law

- Governs the angle at which a refracted ray bends
- Computation based on refraction index of original medium, n, versus new index n_t
- $n_t \sin\theta = n \sin\phi$

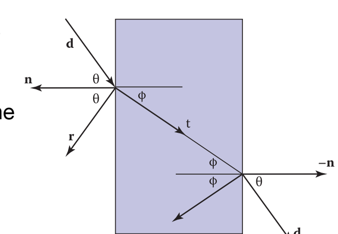


Snell's Law

- Working with cosine's are easier because we can use dot products
- Can derive the vector for the refraction direction **t** as

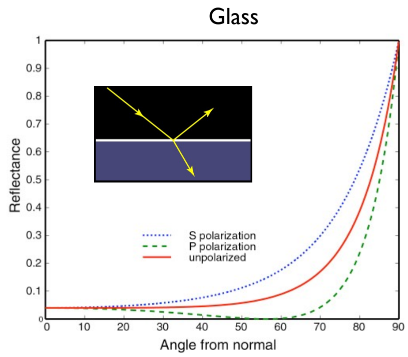
$$\mathbf{t} = \frac{n(\mathbf{d}+\mathbf{n}\cos\theta)}{n_t} - \mathbf{n}\cos\phi$$

$$= \frac{n(\mathbf{d}-\mathbf{n}(\mathbf{d}\cdot\mathbf{n}))}{n_t} - \mathbf{n}\sqrt{1 - \frac{n^2\left(1-(\mathbf{d}\cdot\mathbf{n})^2\right)}{n_t^2}}$$



Specular Reflection from Dielectrics like Glass/Water

- Significant dependence on angle
- About 4% at normal incidence
- Nearly 100% at grazing (rest of the light is transmitted)
- Getting this right has a strong affect on appearance



Fresnel Equation Models the Reflectivity of Dielectrics

- Can be used to predict how much light reflects from a smooth interface (usually, one material is air/empty space)
- R is the fraction that is reflected
- (1-R) is the fraction that is transmitted

$$F_p = \frac{\eta_2 \cos\theta_1 - \eta_1 \cos\theta_2}{\eta_2 \cos\theta_1 + \eta_1 \cos\theta_2}$$
$$F_s = \frac{\eta_1 \cos\theta_1 - \eta_2 \cos\theta_2}{\eta_1 \cos\theta_1 + \eta_2 \cos\theta_2}$$
$$R = \frac{1}{2}\left(F_p^2 + F_s^2\right)$$

Schlick's Approximation to Fresnel Equations

- A quick hack that works pretty well is to approximate R as

$$R(\theta) = R_0 + (1 - R_0)(1 - \cos\theta)^5$$

- Where R_0 is defined by

$$R_0 = \left(\frac{n_t - 1}{n_t + 1}\right)^2$$

Often, assume n is 1.0 for air

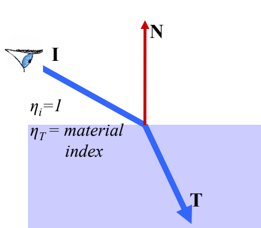
$$F_p = \frac{\eta_2 - \eta_1}{\eta_2 + \eta_1}$$
$$F_s = \frac{\eta_1 - \eta_2}{\eta_1 + \eta_2}$$
$$R_0 = \left(\frac{\eta_2 - \eta_1}{\eta_2 + \eta_1}\right)^2$$

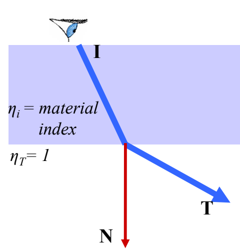
Ray Tracing Dielectrics

- Like a reflective surface, use recursive ray tracing
- But we need two rays:
  - One reflects off the surface (same as mirror ray)
  - The other crosses the surface (computed using Snell's law)
- Both rays do not always exist (e.g. total internal reflection)
- Splitting into two rays creates two recursive paths, sometimes called a **ray tree**
  - Limit recursion with depth

Refraction and Sidedness of Objects

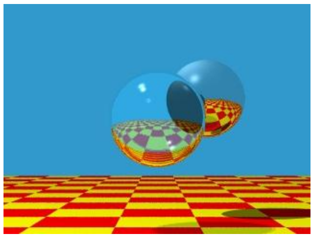
- Make sure you know whether you're entering or leaving the transmissive material. How? Use normals!
- Intersection transmissive objects are a problem (we will ignore for now)

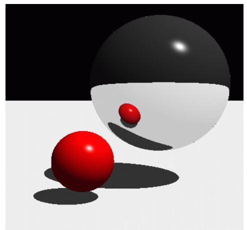




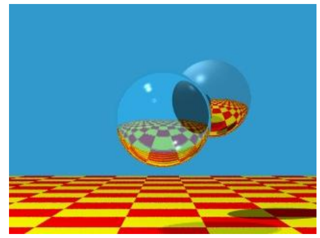
Distribution Ray Tracing

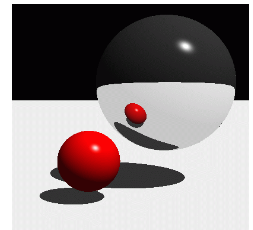
Reality Check: Do These Pictures Look Real?





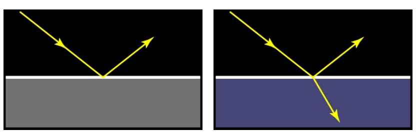
What's Wrong?



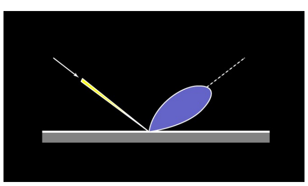


- No surface is a perfect mirror because no surface is perfectly smooth

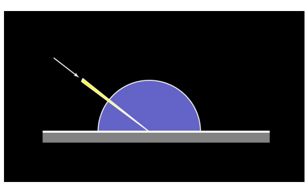
What have we modeled?



ideal specular (mirror)

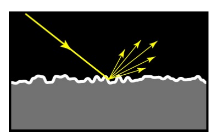


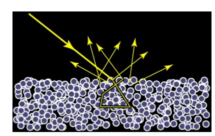
glossy specular



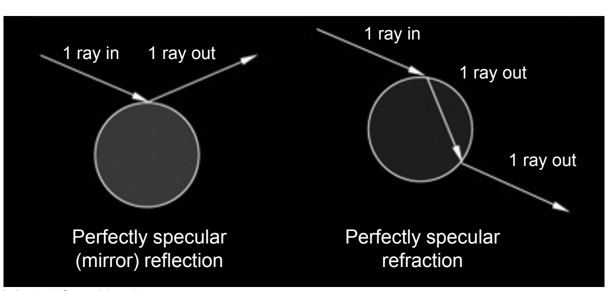
Lambertian

Most Surfaces have Microgeometry





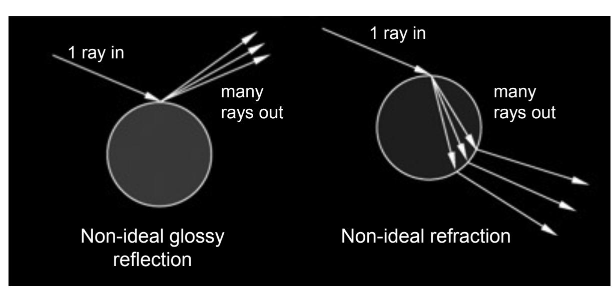
Ideal Reflection/Refraction



Adapted from blender.org

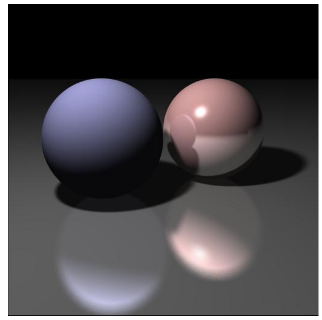
Non-Ideal Reflection/Refraction

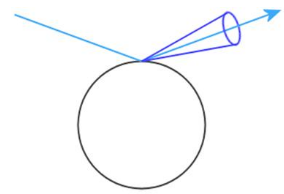
- Can approximate the microgeometry



Adapted from blender.org

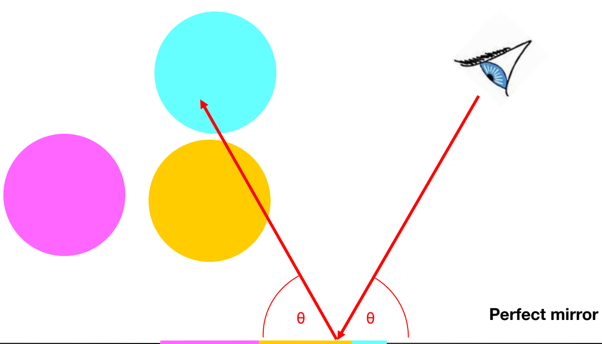
Approach: Distribution Glossy Reflection by Randomly Sampling Rays





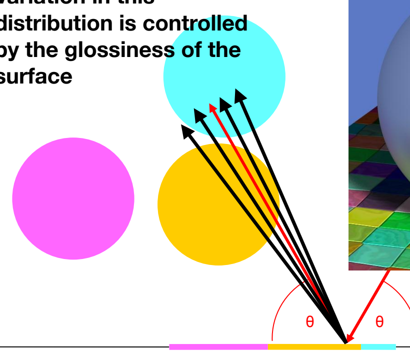
https://graphics.stanford.edu/wikis/cs148-11-fall/RaytracingResults
http://www.baylee-online.net/Projects/Raytracing/Algorithms/Glossy-Reflection-Transmission

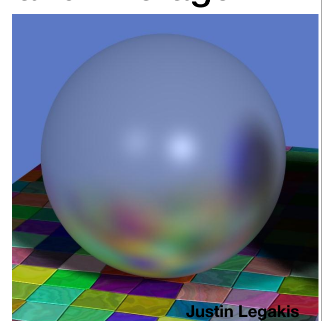
Ideal Reflection: One Ray Per Bounce



Glossy Reflection: Compute Many Rays per Bounce and Average

Variation in this distribution is controlled by the glossiness of the surface





Other Uses of Distribution Ray Tracing

Computer Graphics Volume 18, Number 3 July 1984

Distributed Ray Tracing

Robert L. Cook
Thomas Porter
Loren Carpenter

Computer Division
Lucasfilm Ltd.

Abstract

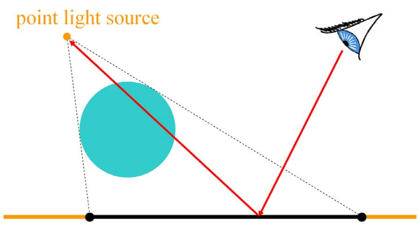
Ray tracing is one of the most elegant techniques in computer graphics. Many phenomena that are difficult or impossible with other techniques are simple with ray tracing, including shadows, reflections, and refracted light. Ray directions, however, have been determined precisely, and this has limited the capabilities of ray tracing. By distributing the directions of the rays according to the analytic function they sample, ray tracing can incorporate fuzzy phenomena. This provides correct and easy solutions to some previously unsolved or partially solved problems, including motion blur, depth of field, penumbras, translucency, and fuzzy reflections. Motion blur and depth of field calculations can be integrated with the visible surface calculations, avoiding the problems found in previous methods.

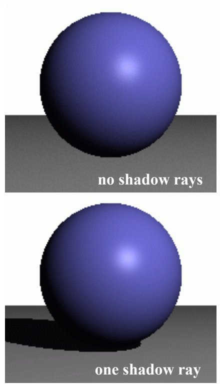
Ray traced images are sharp because ray directions are determined precisely from geometry. Fuzzy phenomenon would seem to require large numbers of additional samples per ray. By distributing the rays rather than adding more of them, however, fuzzy phenomena can be rendered with no additional rays beyond those required for spatially oversampled ray tracing. This approach provides correct and easy solutions to some previously unsolved problems.

This approach has not been possible before because of aliasing. Ray tracing is a form of point sampling and, as such, has been subject to aliasing artifacts. This aliasing is not inherent, however, and ray tracing can be filtered as effectively as any analytic method[4]. The filtering does incur the expense of additional rays, but it is not

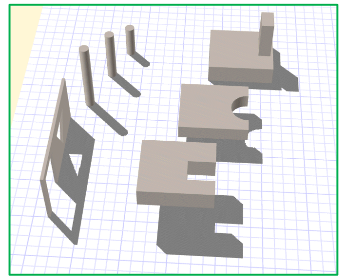
Problem: Hard Shadows

- One shadow ray per intersection per point light source

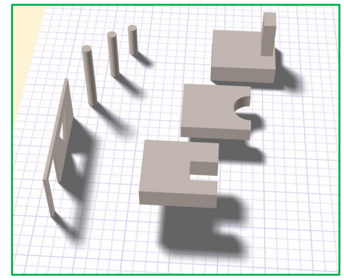




Soft Shadows



Hard shadows

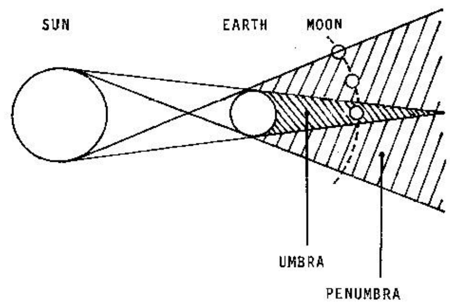


Soft shadows

http://erich.realtimerendering.com/shadow_comparison.html

Soft Shadows

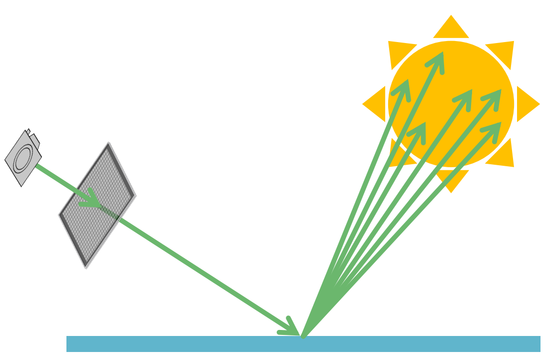
What Causes Soft Shadows



http://user.online.be/felixverbelen/lunecl.jpg

Lights aren't all point sources

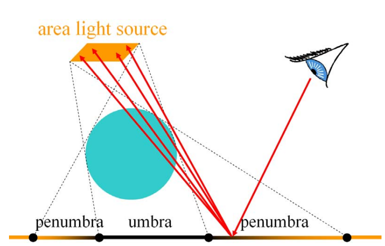
Distribution Soft Shadows

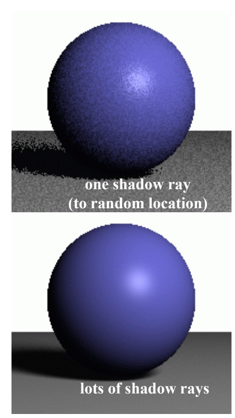


Randomly sample light rays

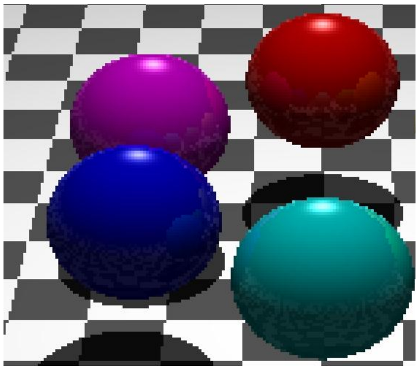
Computing Soft Shadows

- Model light sources as spanning an area
- Sample random positions on area light source and average rays





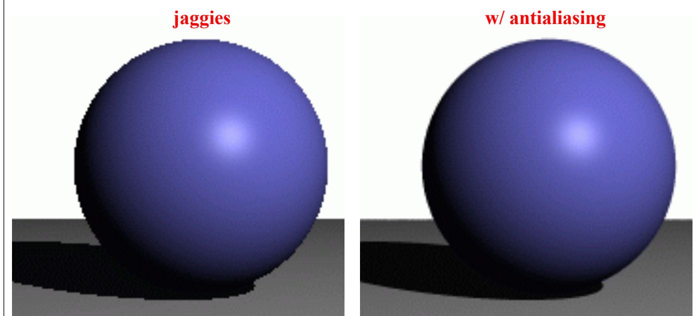
Problem: Aliasing



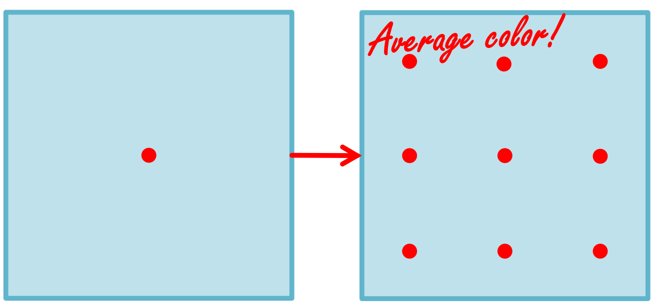
http://www.hackification.com/2008/08/31/experiments-in-ray-tracing-part-8-anti-aliasing/

Antialiasing w/ Supersampling

- Cast multiple rays per pixel, average result

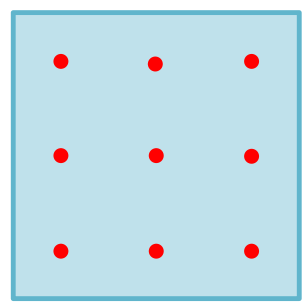


Distribution Antialiasing



Multiple rays per pixel

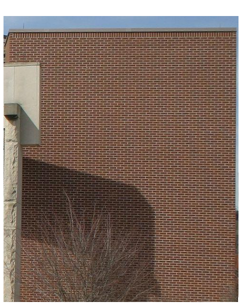
Distribution Antialiasing w/ Regular Sampling

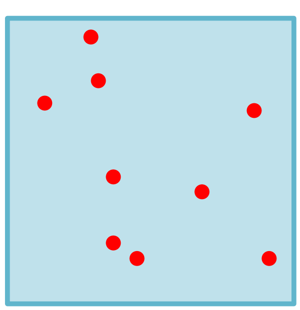


http://upload.wikimedia.org/wikipedia/commons/f/fb/Moire_pattern_of_bricks_small.jpg

Multiple rays per pixel

Distribution Antialiasing w/ Random Sampling

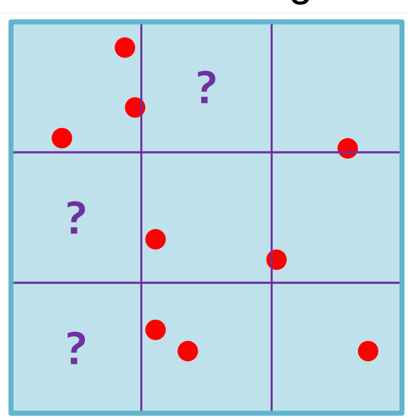




http://en.wikipedia.org/wiki/File:Moire_pattern_of_bricks.jpg

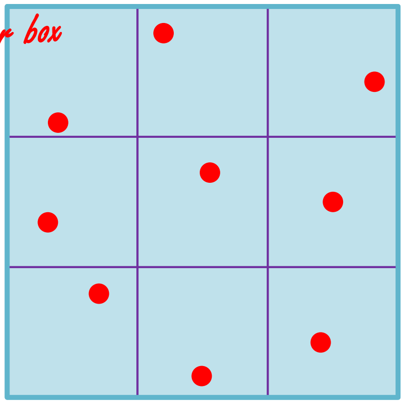
Remove Moiré patterns

Random Sampling Could Miss Regions Without Enough Sampling



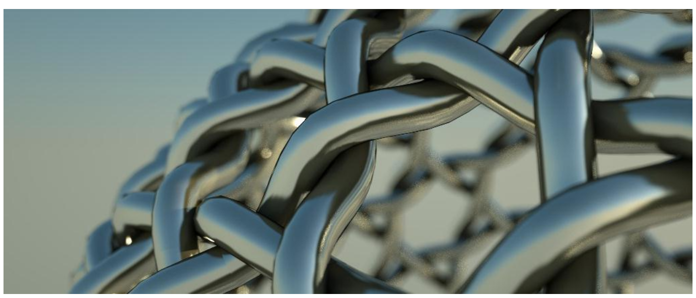
Stratified (Jittered) Sampling

One ray per box



Problem: Focus
Real Lenses Have Depth of Field

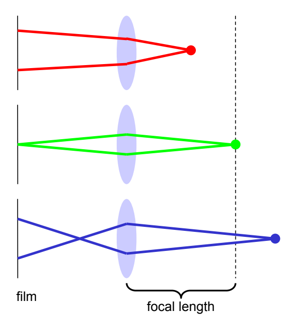
Problem: Focus
Real Lenses Have Depth of Field

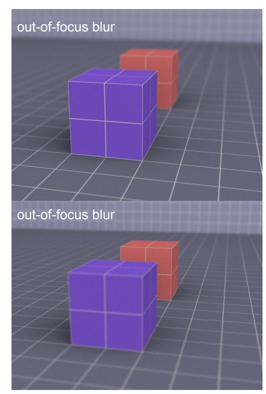


http://liam887.files.wordpress.com/2010/08/weaver.jpg

Depth of Field

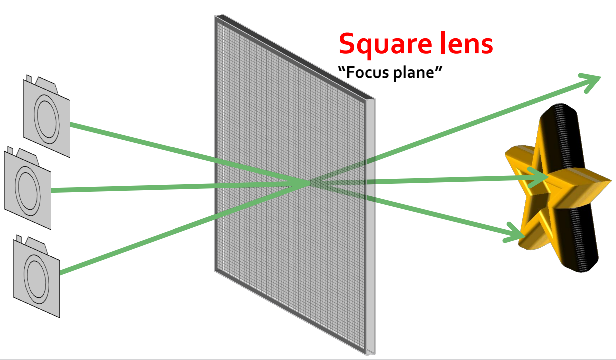
- Multiple rays per pixel, sample lens aperture





Justin Legakis

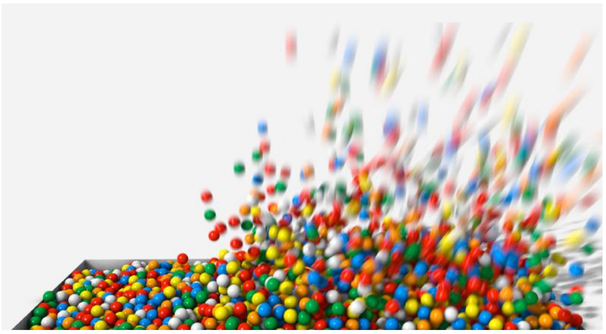
Distribution Depth of Field



Randomly sample eye positions

Problem: Exposure Time
Real Sensors Take Time to Acquire

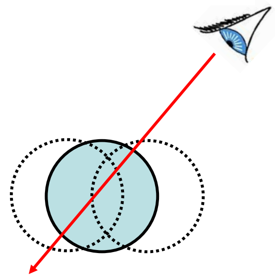
Problem: Exposure Time
Real Sensors Take Time to Acquire



http://www.matkovic.com/anto/3dl-test-balls-01.jpg

Motion Blur

- Sample objects temporally over a time interval



Rob Cook

Lec13 Required Reading

- FOCG, Ch. 23
- Check out recommended reading for some additional references

Reminder: Assignment 03

Assigned: Wednesday, Sept. 25
Written (Past) Due: Monday, Oct. 7, 4:59:59 pm
Program Due: Wednesday, Oct. 9, 4:59:59 pm