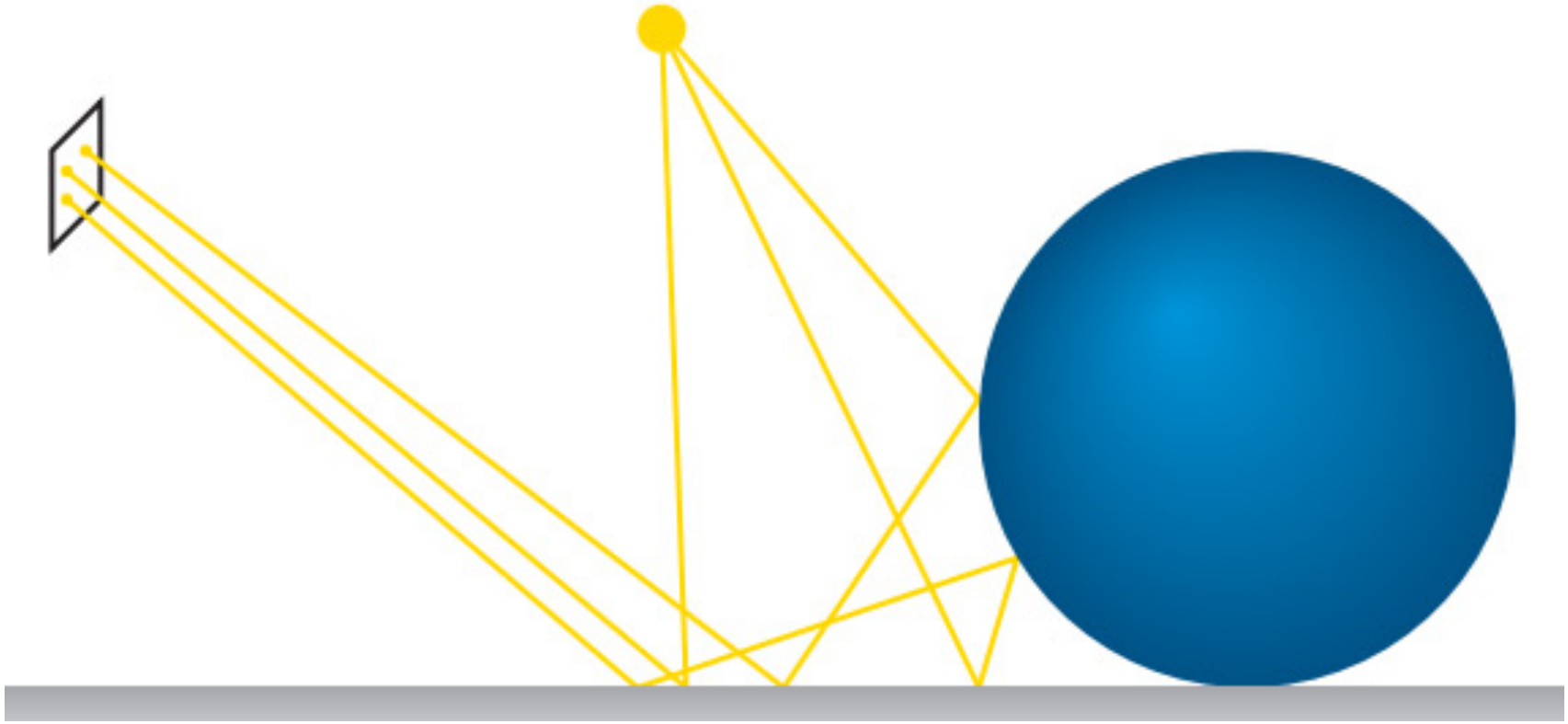
Ray Tracing

CS 148: Summer 2016
Introduction of Graphics and Imaging
Zahid Hossain

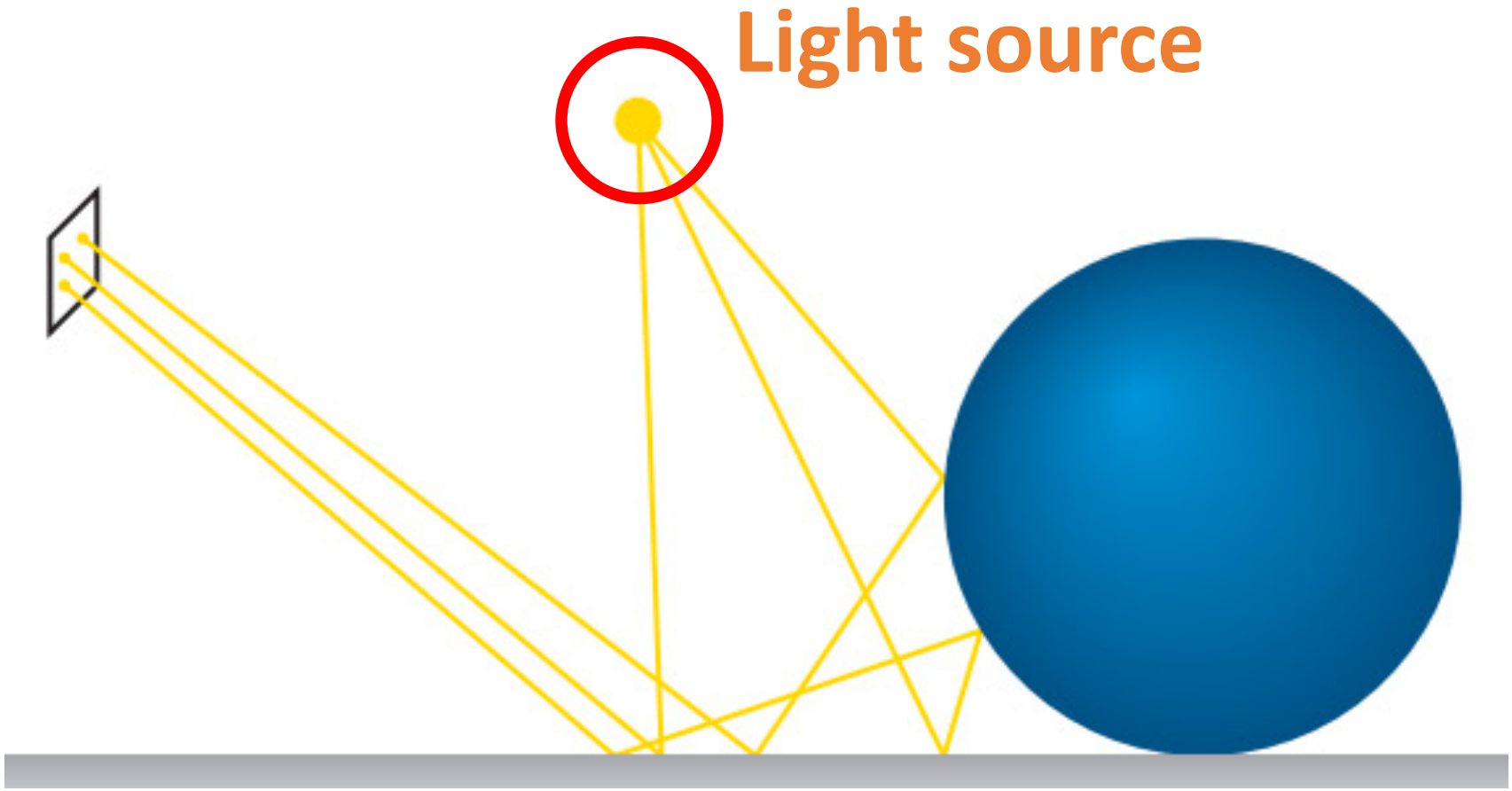
https://en.wikipedia.org/wiki/Ray_tracing_%28graphics%29

Paths of Light



http://software.intel.com/file/37491

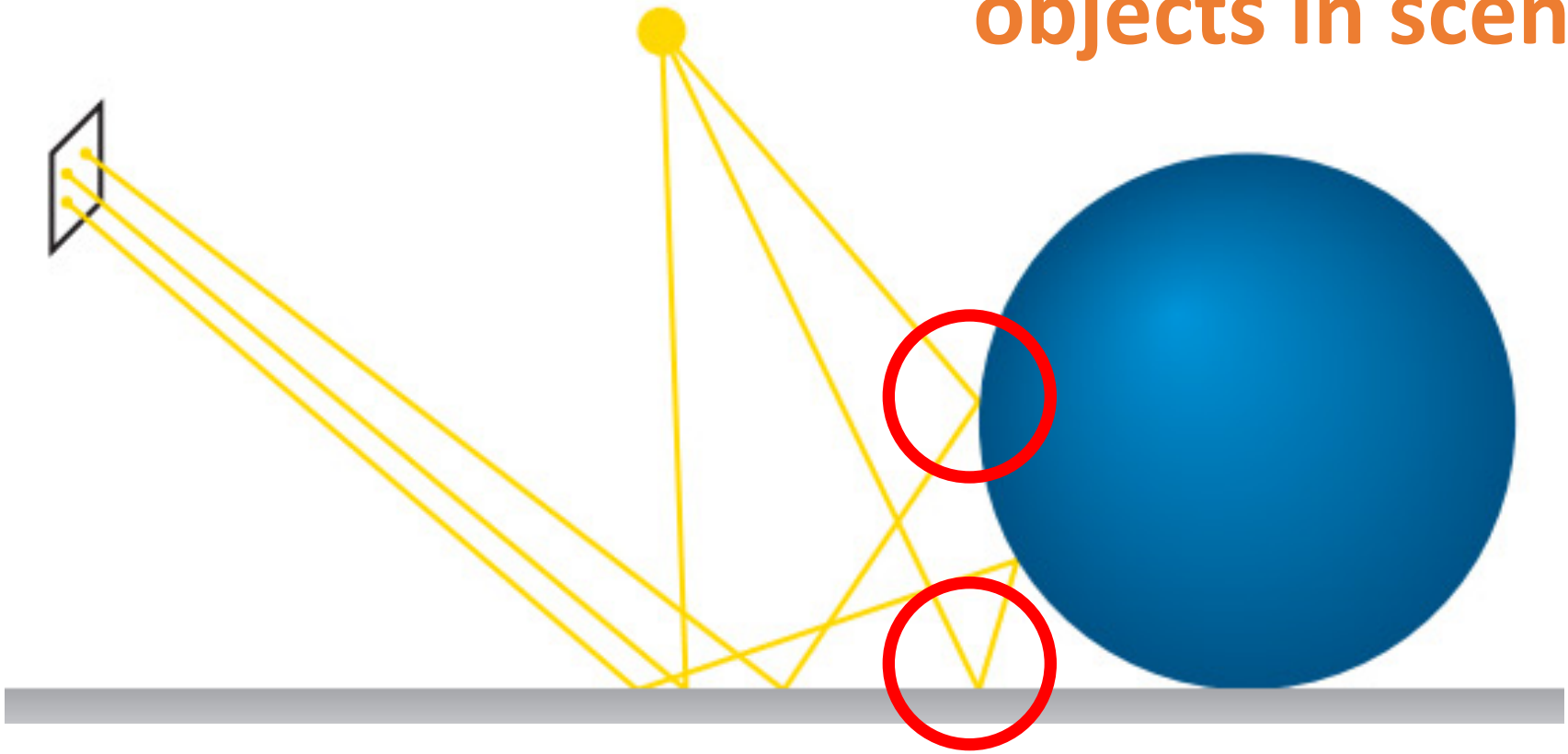
Paths of Light



http://software.intel.com/file/37491

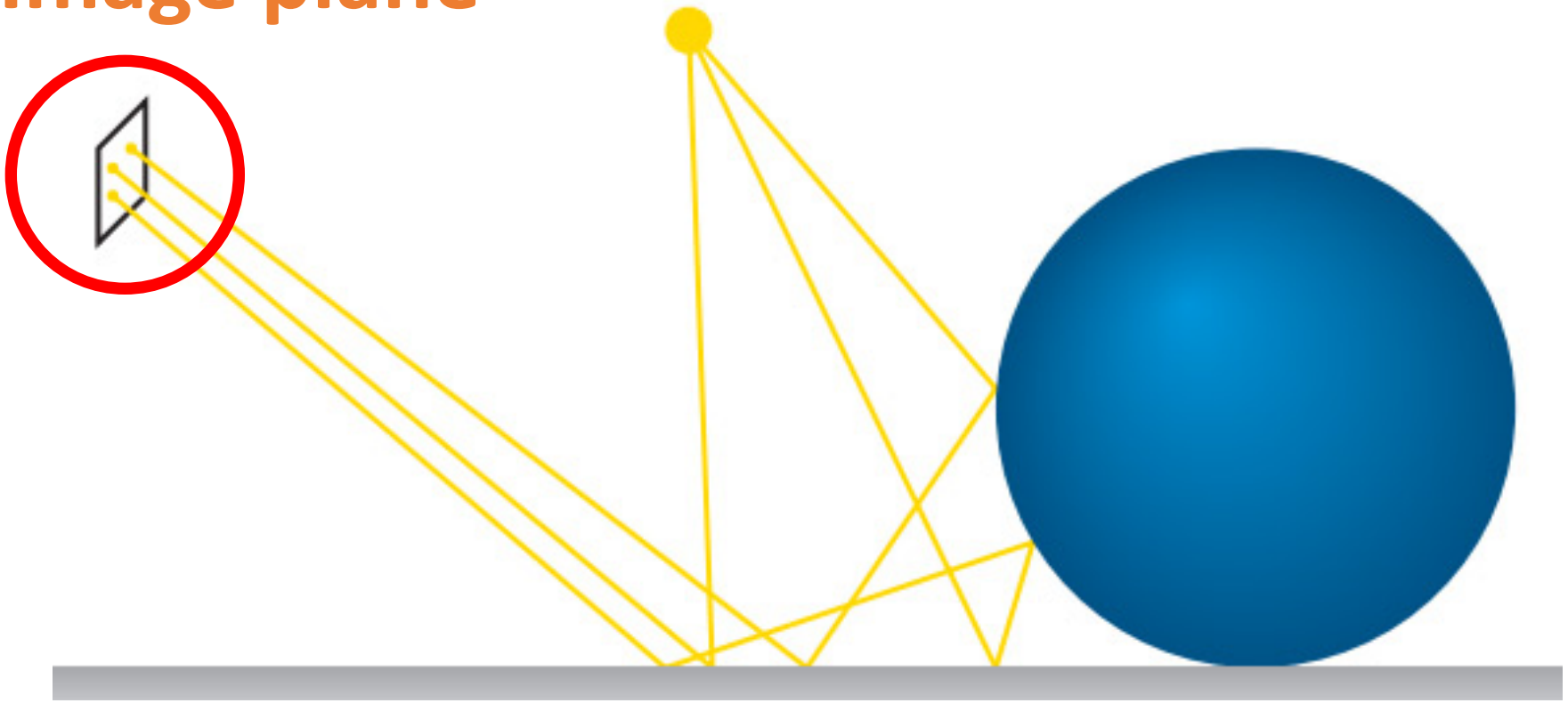
Paths of Light

Light interacts with objects in scene



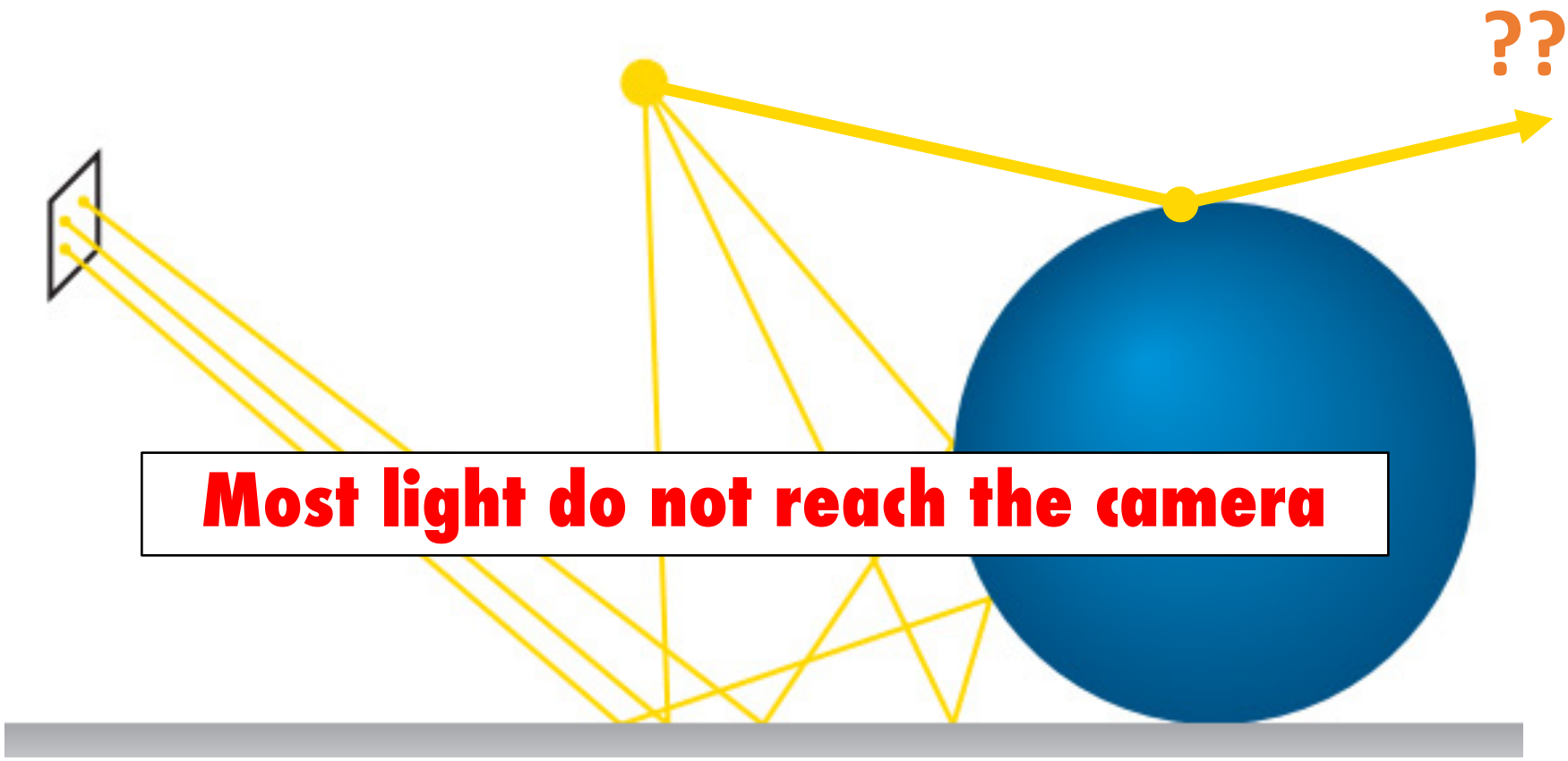
http://software.intel.com/file/37491

Paths of Light



http://software.intel.com/file/37491

Problem



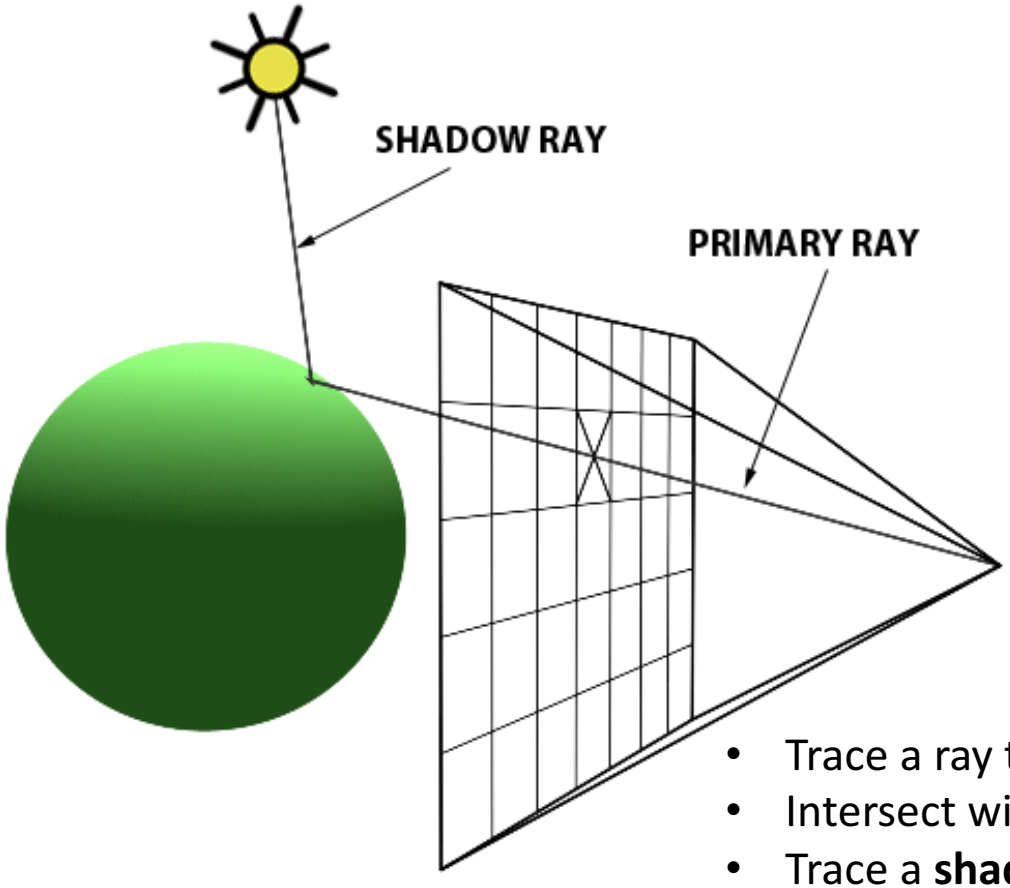
http://software.intel.com/file/37491

Forward Ray Tracing

Trace rays from light to camera

Backward Ray Tracing

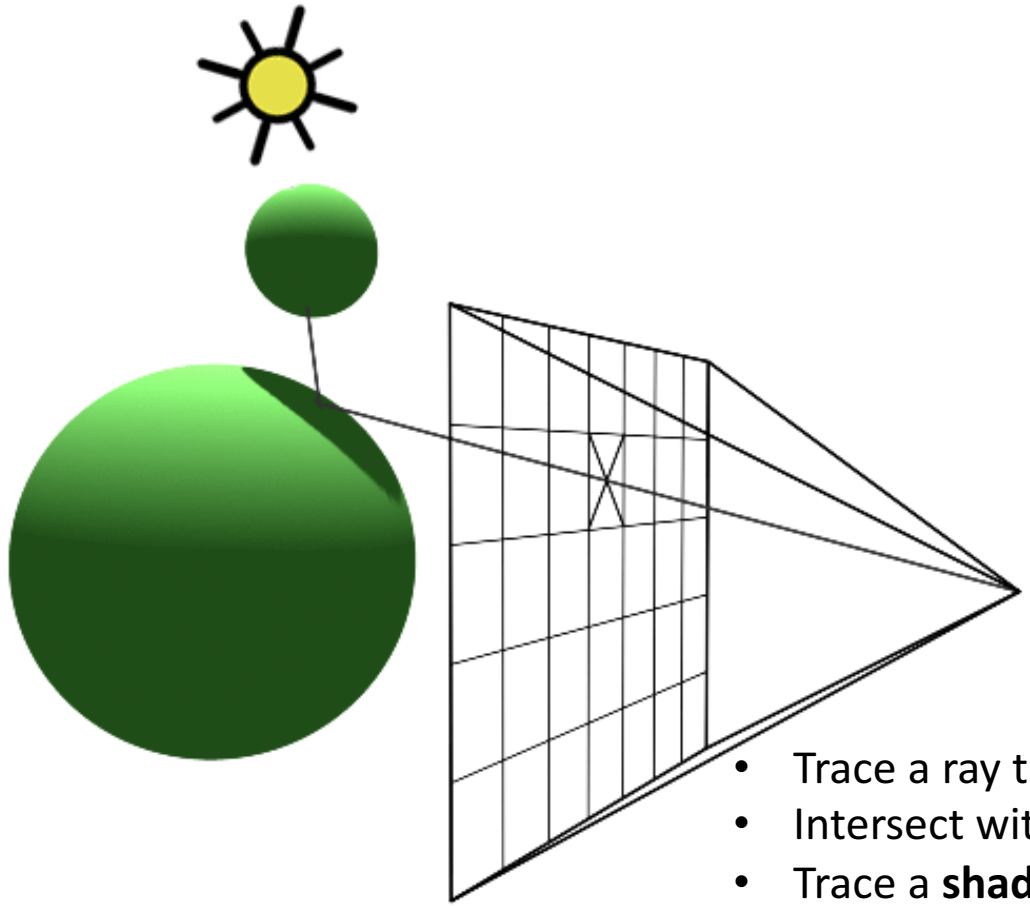
Ray Tracing Algorithm



- Trace a ray through each pixel
- Intersect with geometry
- Trace a **shadow ray** to the light
- Shade the pixel

http://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-ray-tracing/implementing-the-raytracing-algorithm

Ray Tracing Algorithm



- Trace a ray through each pixel
- Intersect with geometry
- Trace a **shadow ray** to the light
- Shade the pixel
  - **Shadow if light not visible**

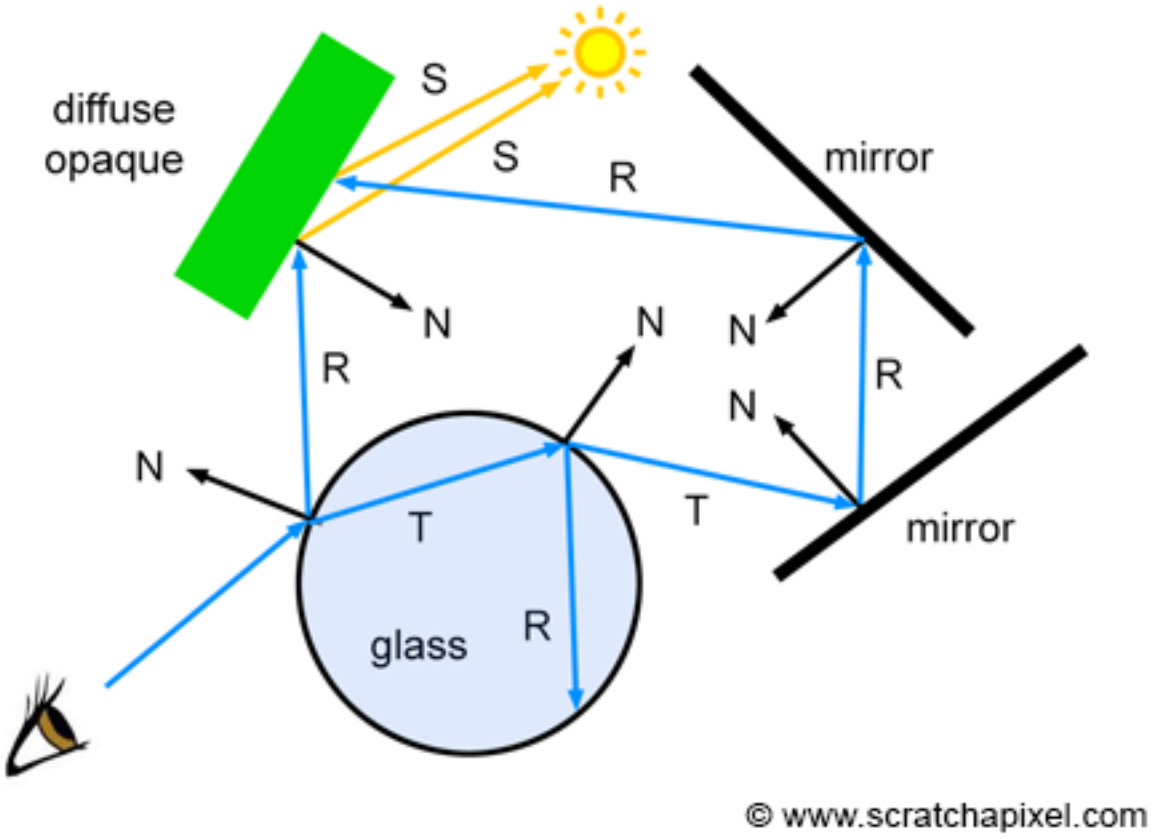
http://www.scratchapixel.com

Simple to implement

- For a simple case:
  - http://www.scratchapixel.com/code.php?id=3&origin=/lessons/3d-basic-rendering/introduction-to-ray-tracing

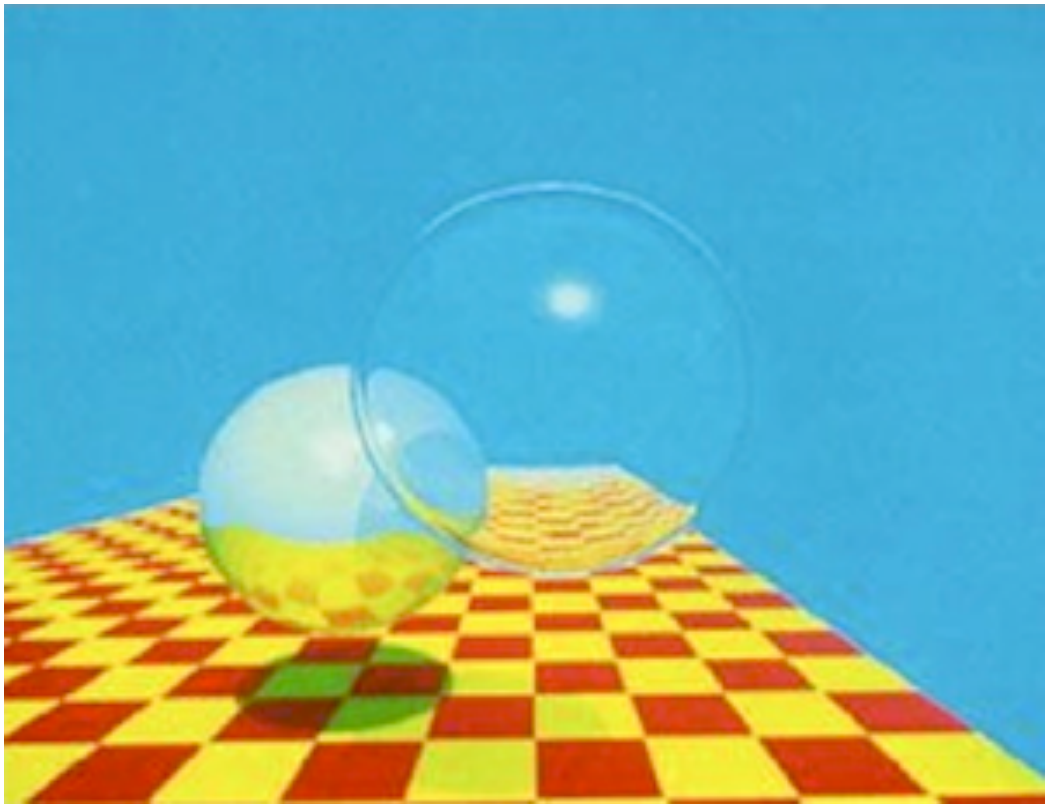
Few lines of code; already impressive

Recursive Ray Tracing



http://www.scratchapixel.com/images/upload/ray-tracing-refresher/rt-whitted-example.png?

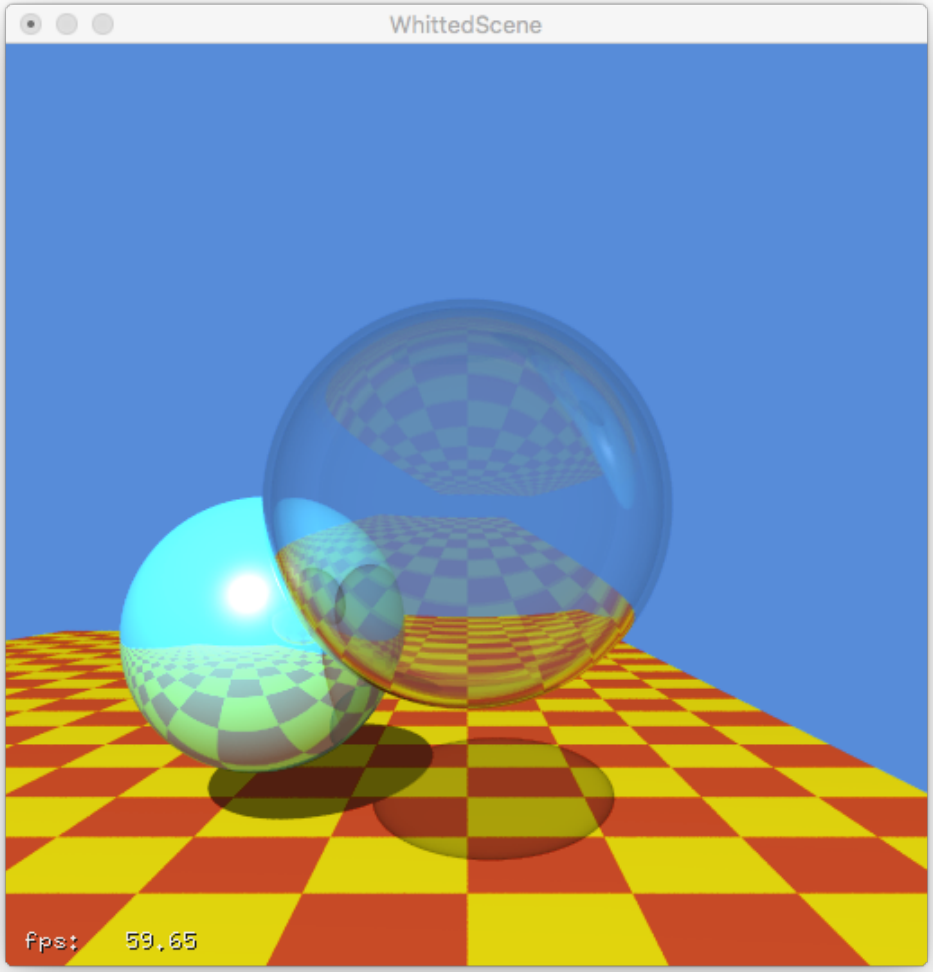
Whitted Ray Tracing



T. Whitted, Communications of ACM, 1980

Render Time: 74 minutes

Today: nVidia OptiX



Render Time: 16 ms ! (60 fps): my Macbook Pro nVidia GTX M750

Today: Realism

http://images.povcomp.com/entries/images/105_main.jpg

Today: Realism

http://images.povcomp.com/entries/images/105_main.jpg

Rasterizer Vs Ray-Tracing

Rasterization

```
for (each object in scene)
    drawObject();
```

Rasterizer Vs Ray-Tracing

Rasterization

```
for (each object in scene)
    drawObject();
```

Ray Tracing

```
for (each pixel in image)
    sendRay();
```

Rasterization Vs Ray Tracing

Rasterization

```
for (each object i
    drawObject();
```

Object-order rendering

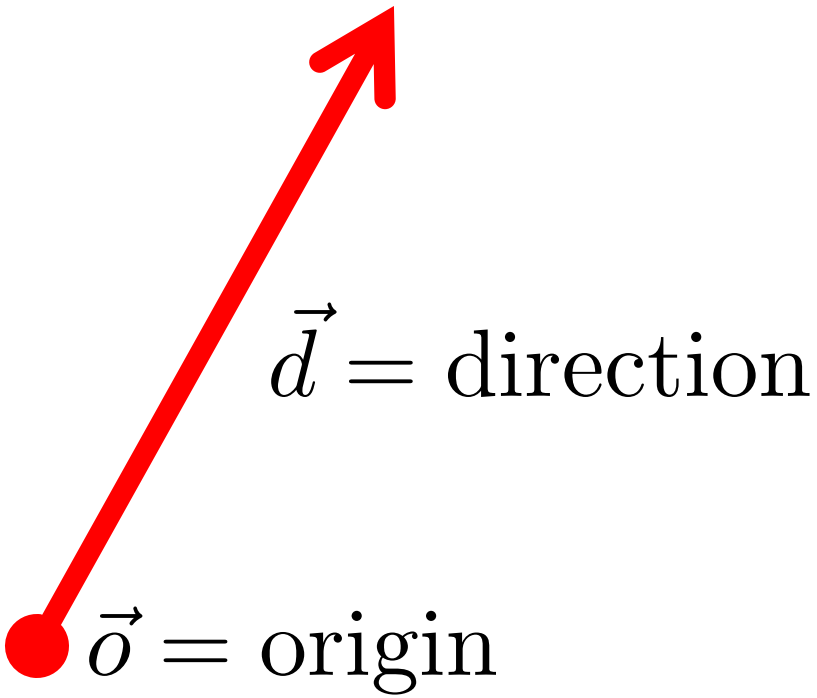
Ray Tracing

```
for (each pixel in
    sendRay();
```

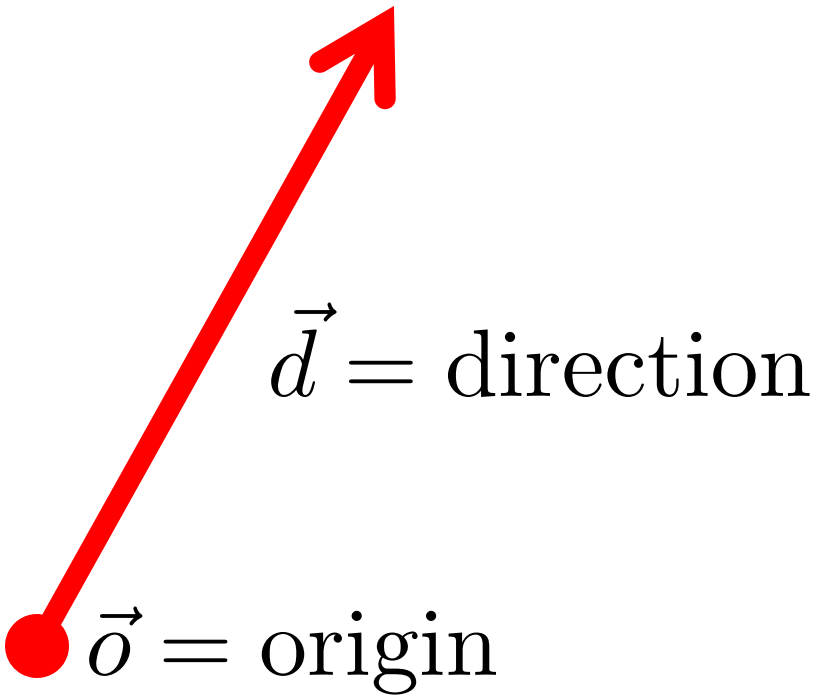
Image-Order Rendering

Math for Ray Tracing

Ray

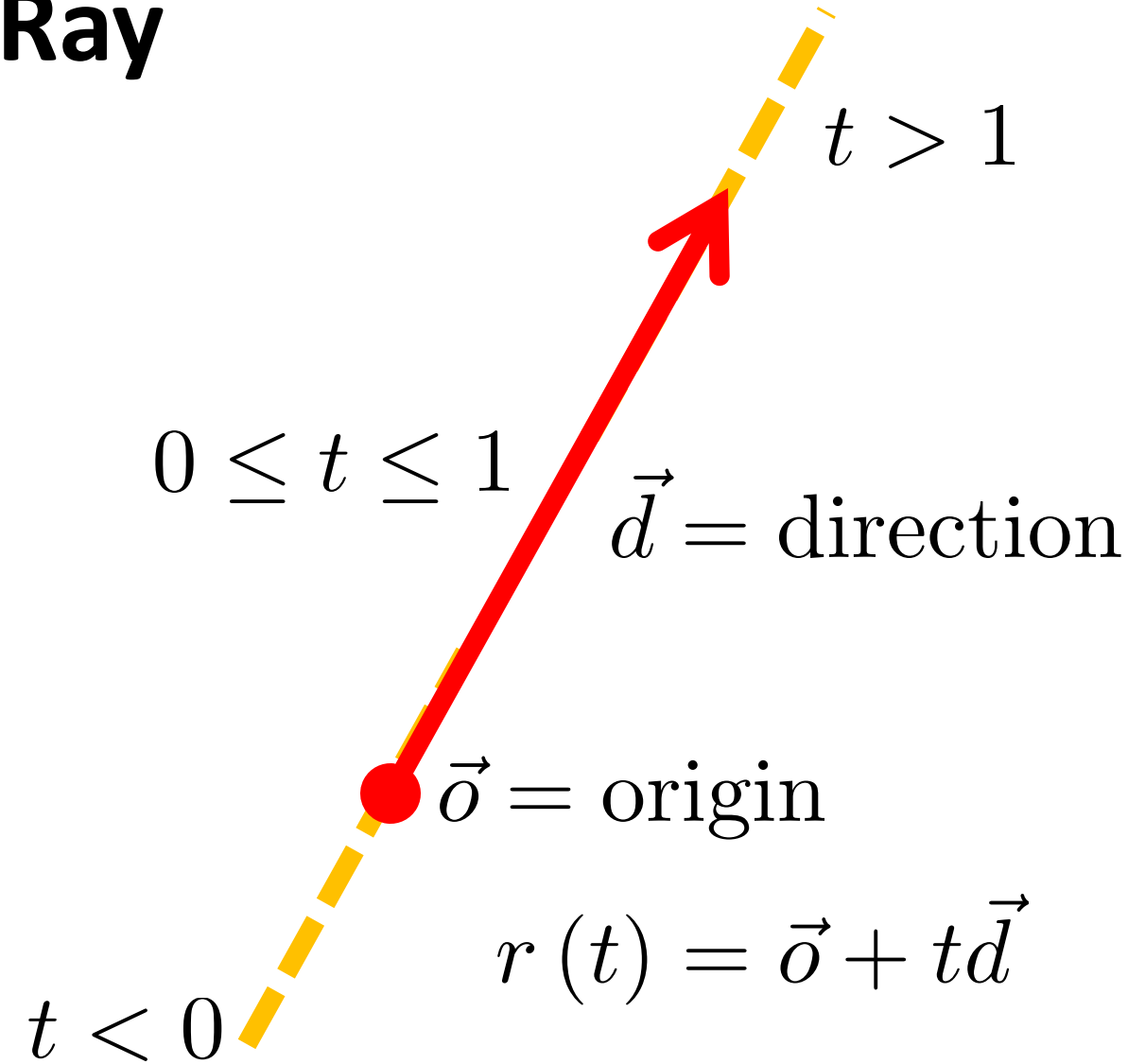


Ray

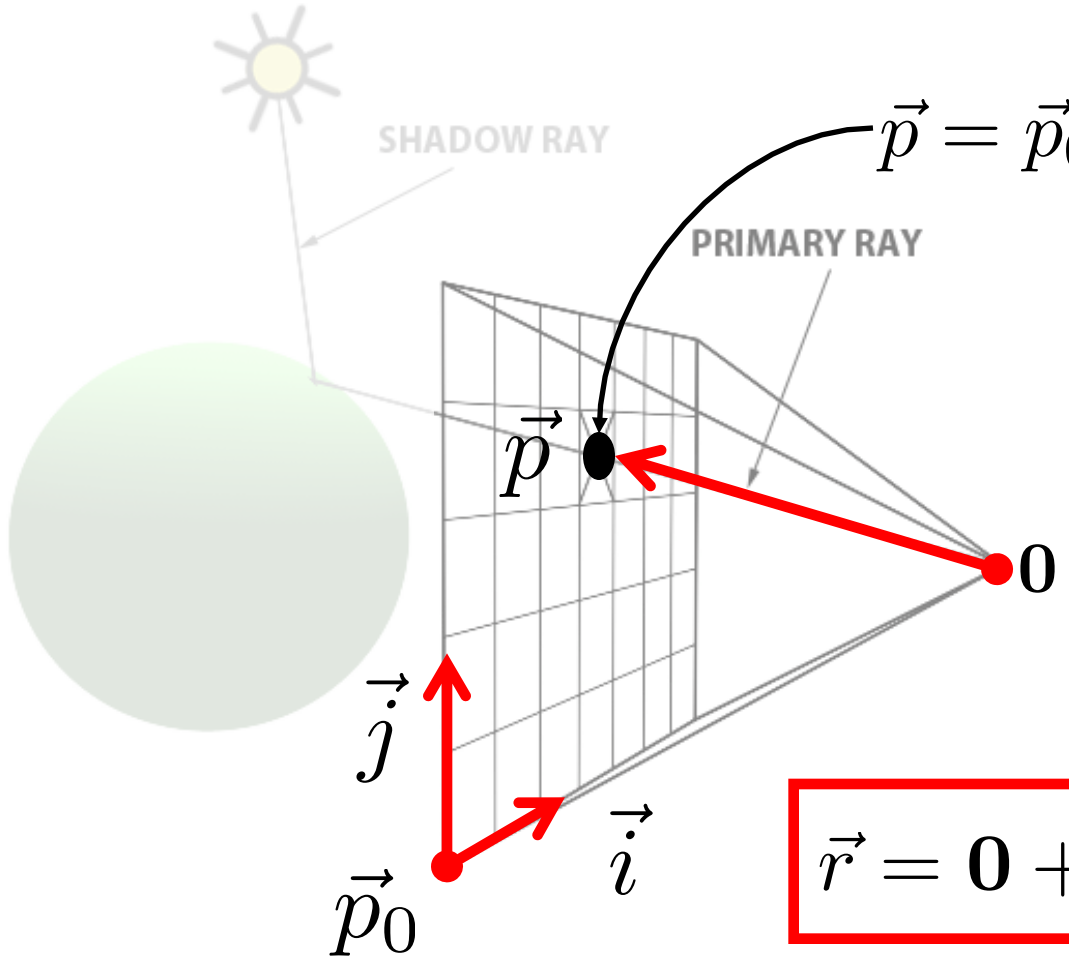


$$r(t) = \vec{o} + t\vec{d}$$

Ray



Generating a Ray



$$\vec{p} = \vec{p}_0 + u\vec{i} + v\vec{j}$$

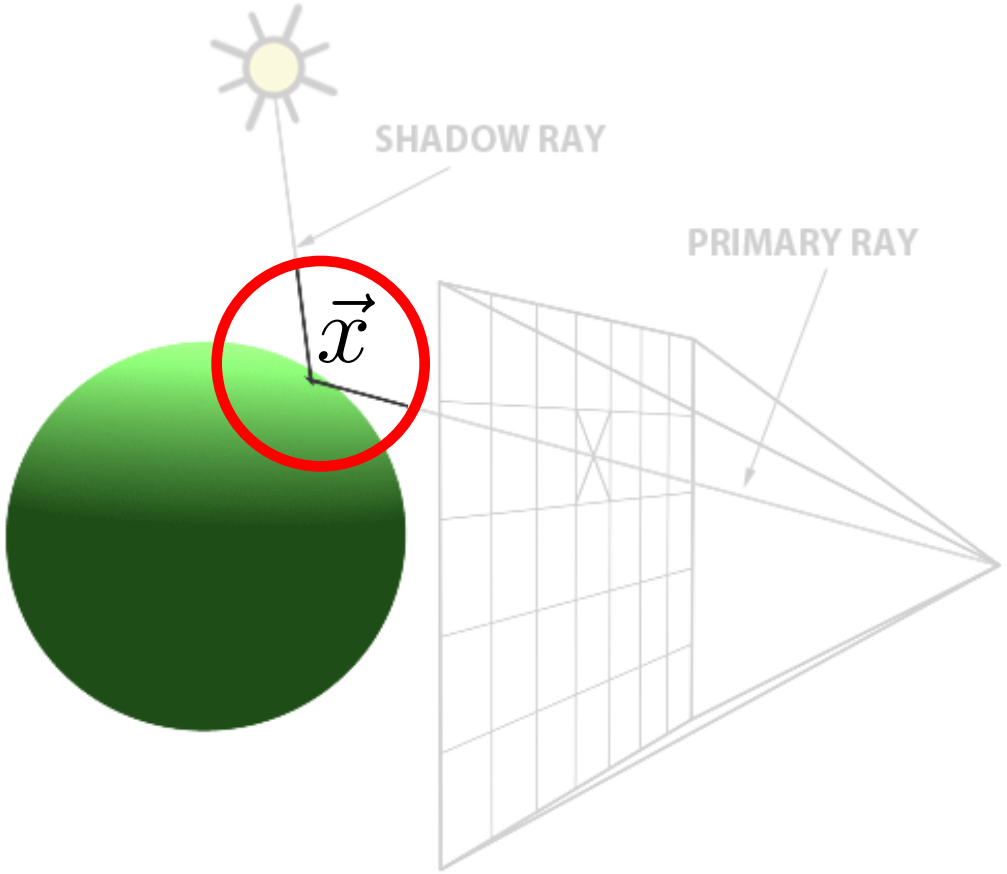
$$u \in [0, \text{width}]$$

$$v \in [0, \text{height}]$$

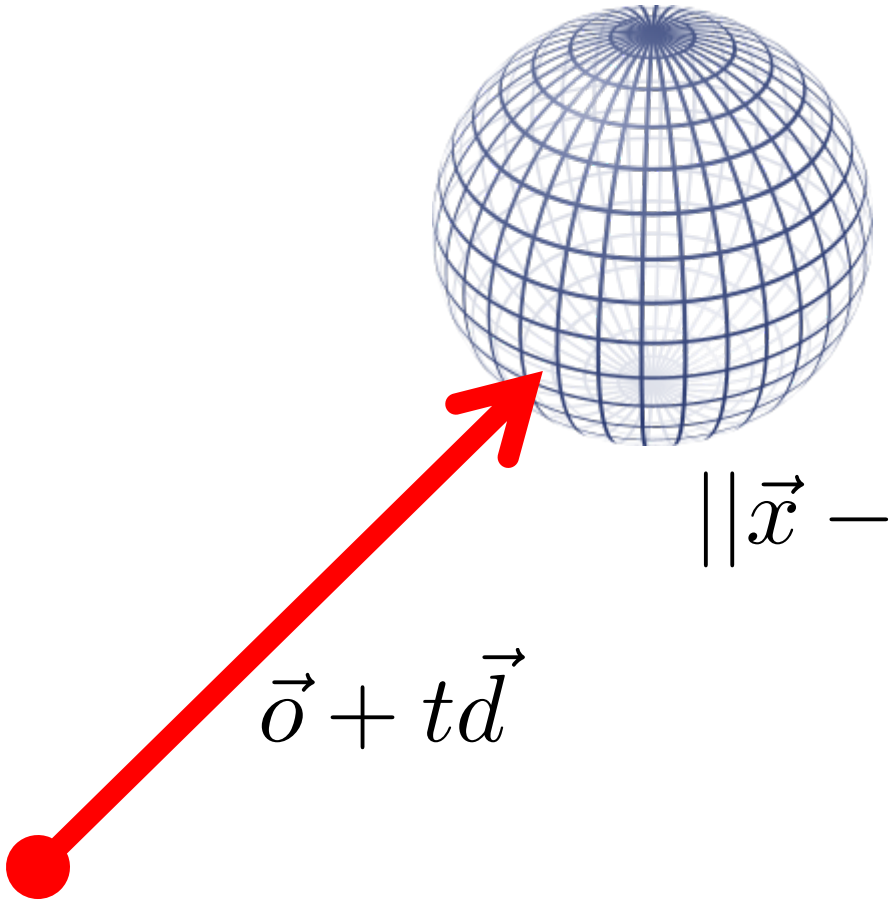
$$\vec{r} = \mathbf{0} + t\vec{p} = t\vec{p}$$

No perspective transformation needed!

Ray Intersection With a Sphere



Ray Intersection With a Sphere



$$||\vec{x} - \vec{c}||^2 = R^2$$

$$\vec{o} + t\vec{d}$$

Ray Intersection with a Sphere

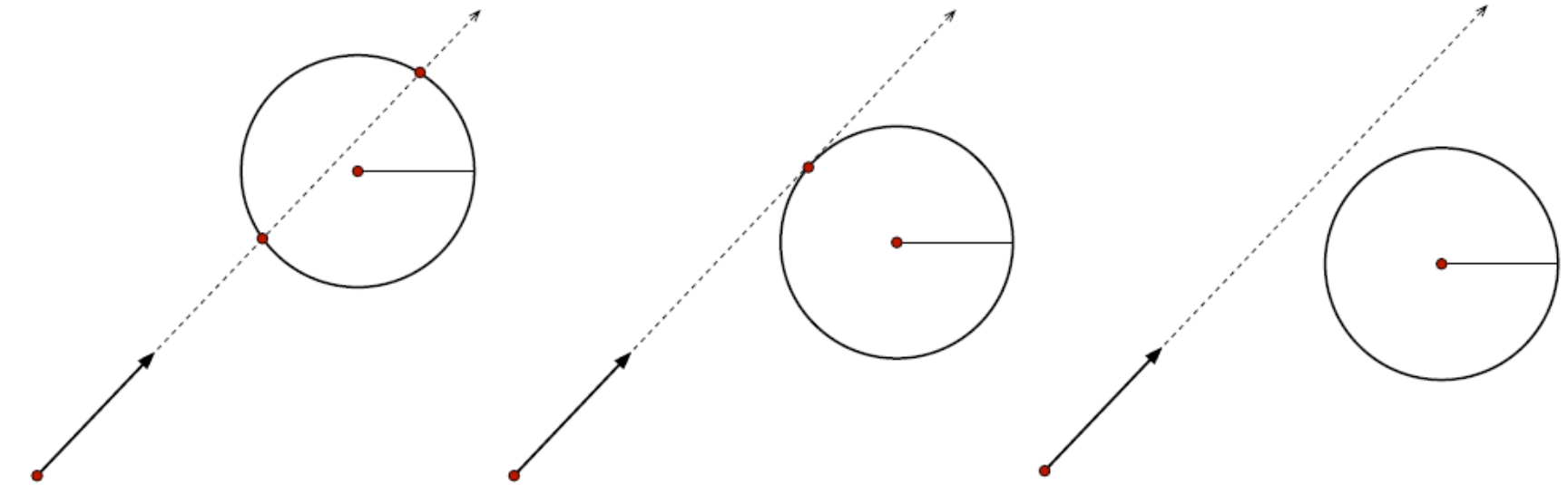
$$\vec{o} + t\vec{d} \qquad\qquad ||\vec{x} - \vec{c}||^2 = R^2$$

$$||\vec{o} + t\vec{d} - \vec{c}||^2 = R^2$$

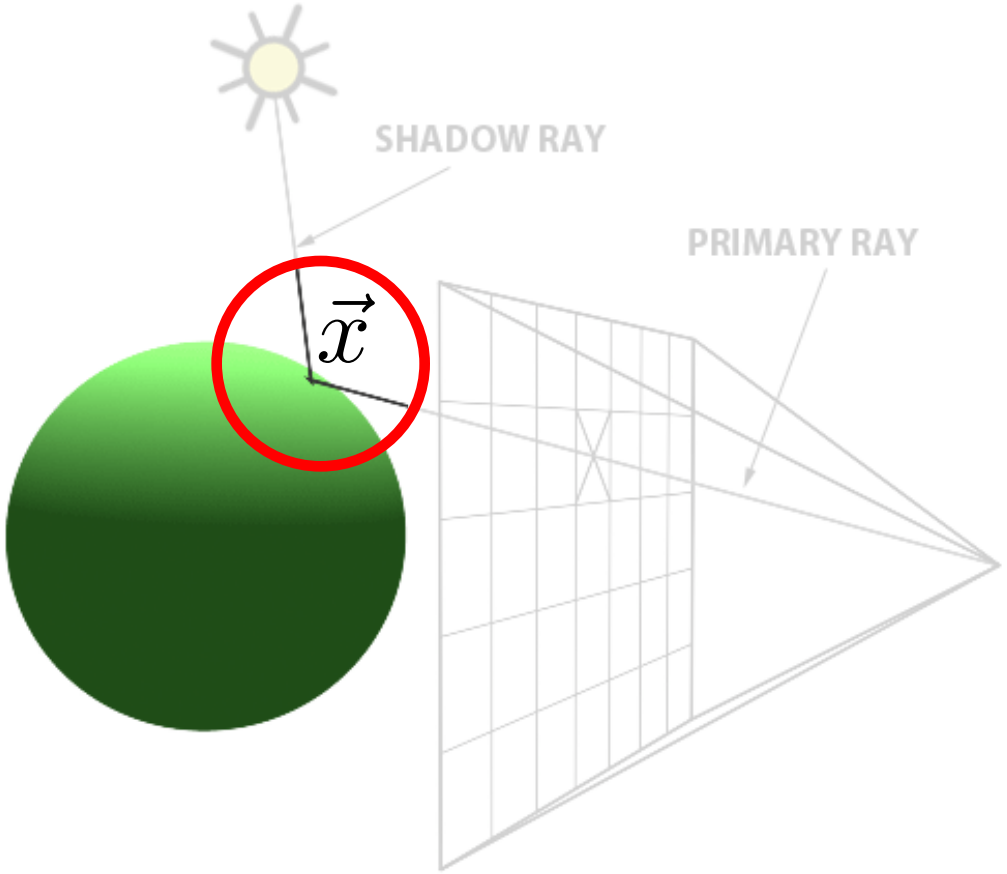
$$\left(\vec{o} + t\vec{d} - \vec{c}\right) \cdot \left(\vec{o} + t\vec{d} - \vec{c}\right) = R^2$$

$$\left(||\vec{d}||^2\right) t^2 + 2\left((\vec{o} - \vec{c}) \cdot \vec{d}\right) t + ||\vec{o} - \vec{c}||^2 = R^2$$

Ray Intersection with a Sphere

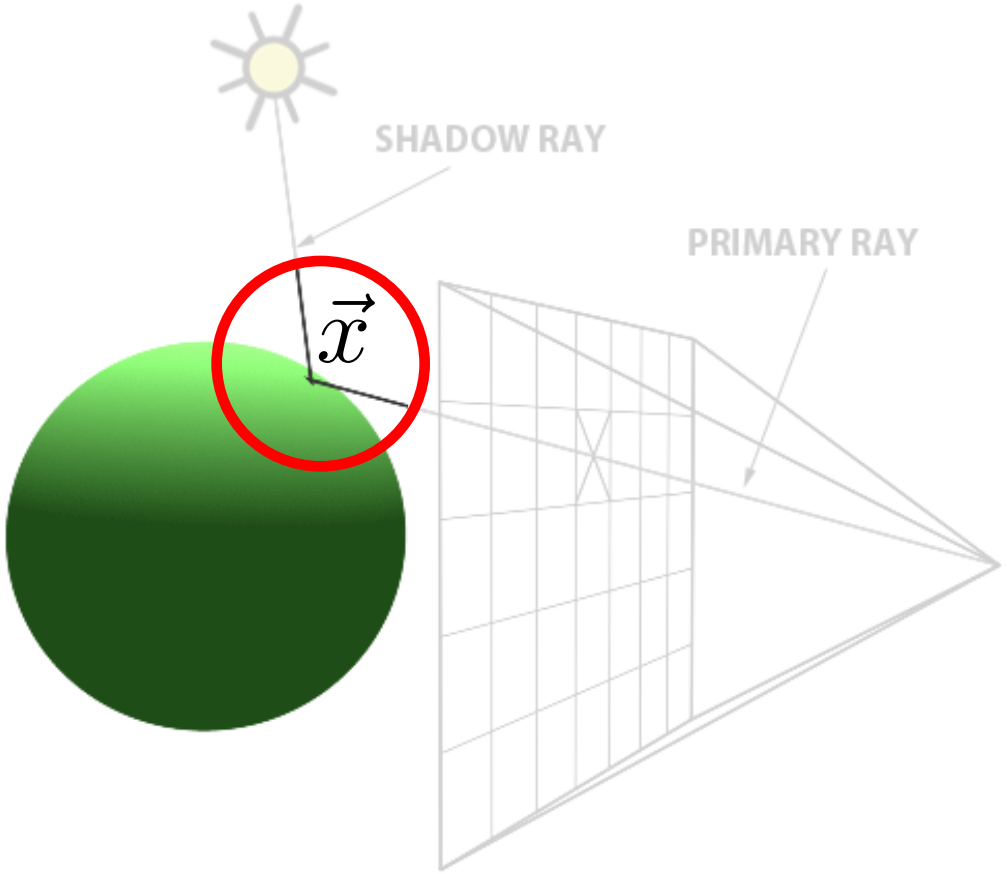


Ray Intersection With a Sphere



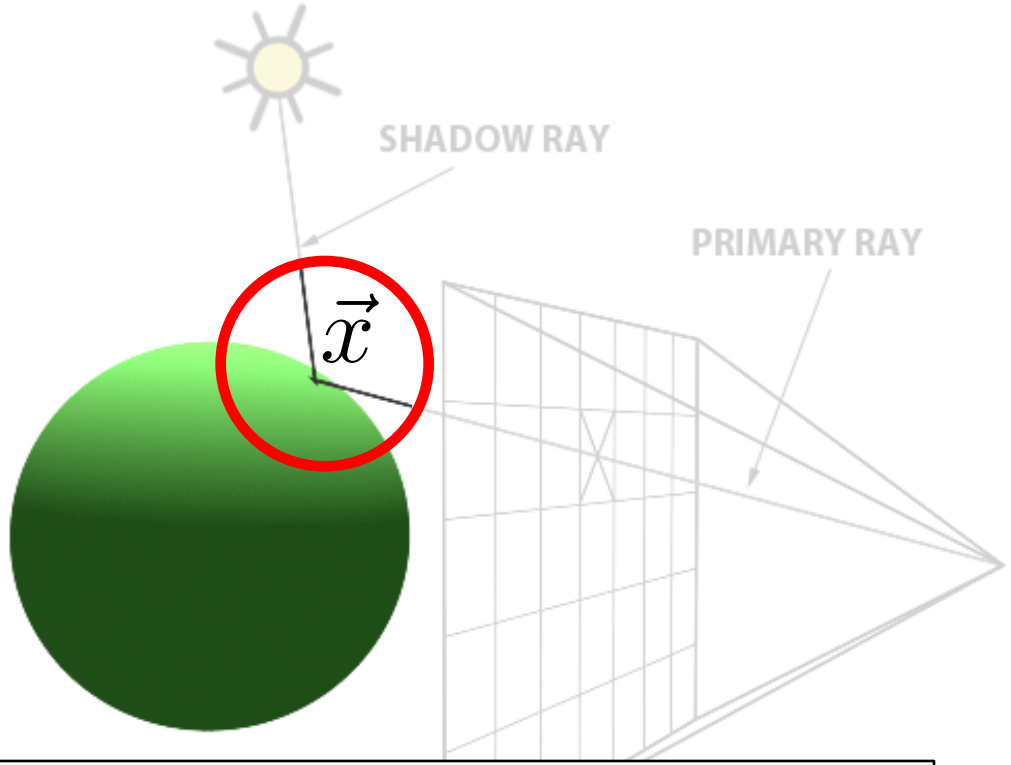
$$\vec{x} = \vec{o} + t^* \vec{d}$$

Ray Intersection With a Sphere



$$\vec{x} = \vec{o} + t^* \vec{d}$$

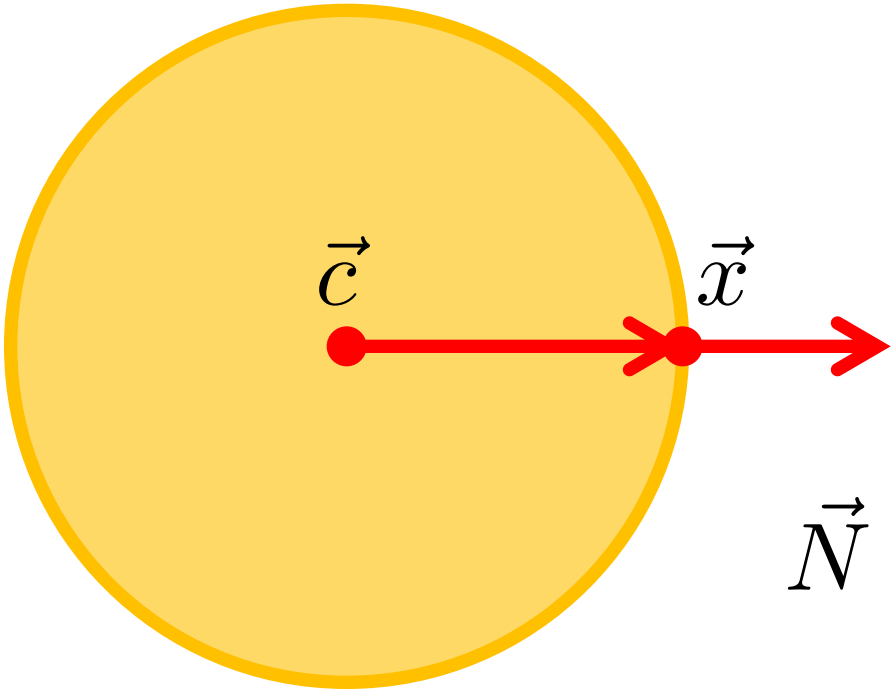
Ray Intersection With a Sphere



If t* < 0 then what ?

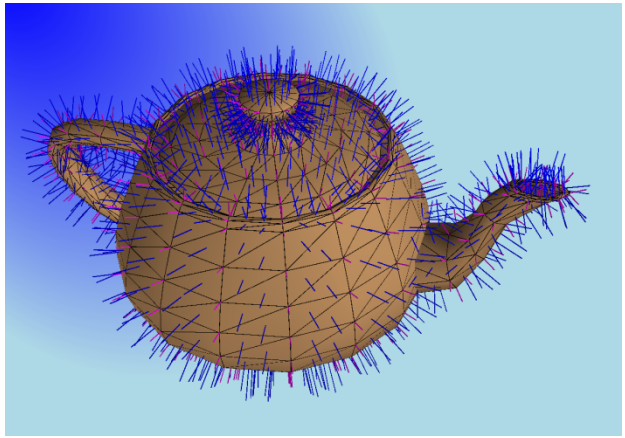
$$\vec{x} = \vec{o} + t^* \vec{d}$$

Sphere Surface Normal

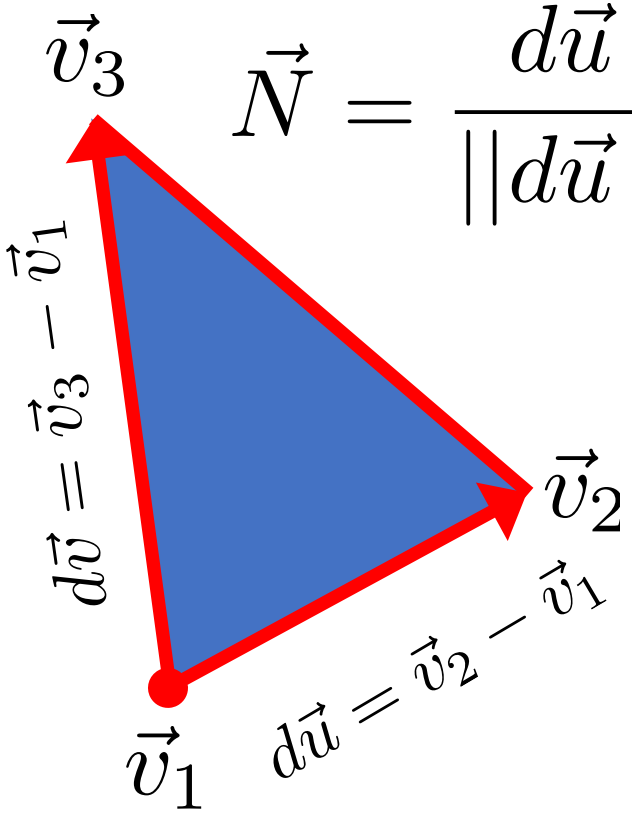


$$\vec{N} = \frac{\vec{x} - \vec{c}}{||\vec{x} - \vec{c}||}$$

Surface Normal

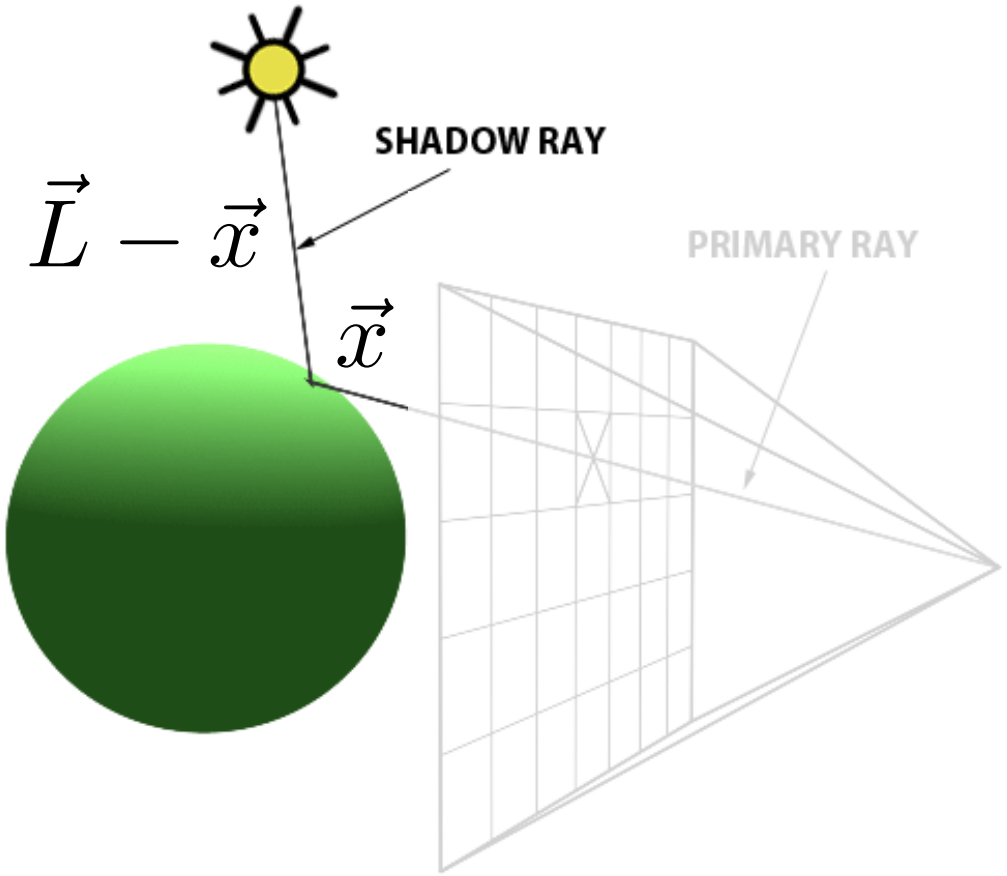


$$\vec{N} = \frac{d\vec{u} \times d\vec{v}}{||d\vec{u} \times d\vec{v}||}$$

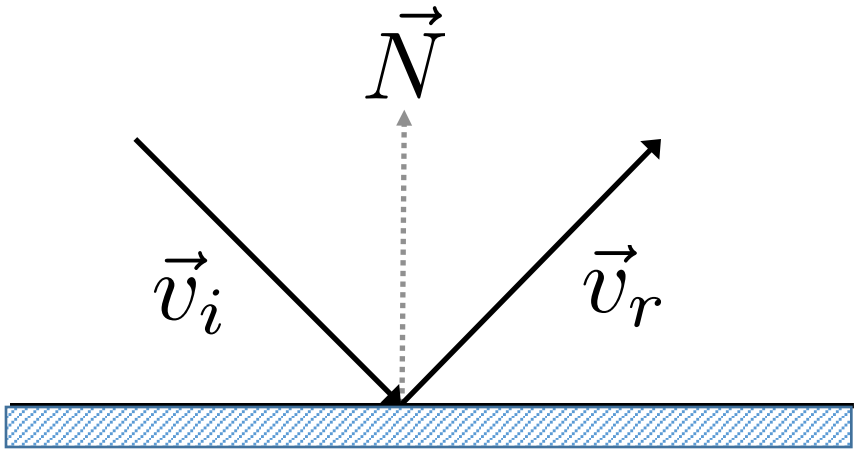


http://designjk.files.wordpress.com/2011/12/teapot-decorated.png

Shadow Ray / Light Ray



Reflection Vector (Recap)



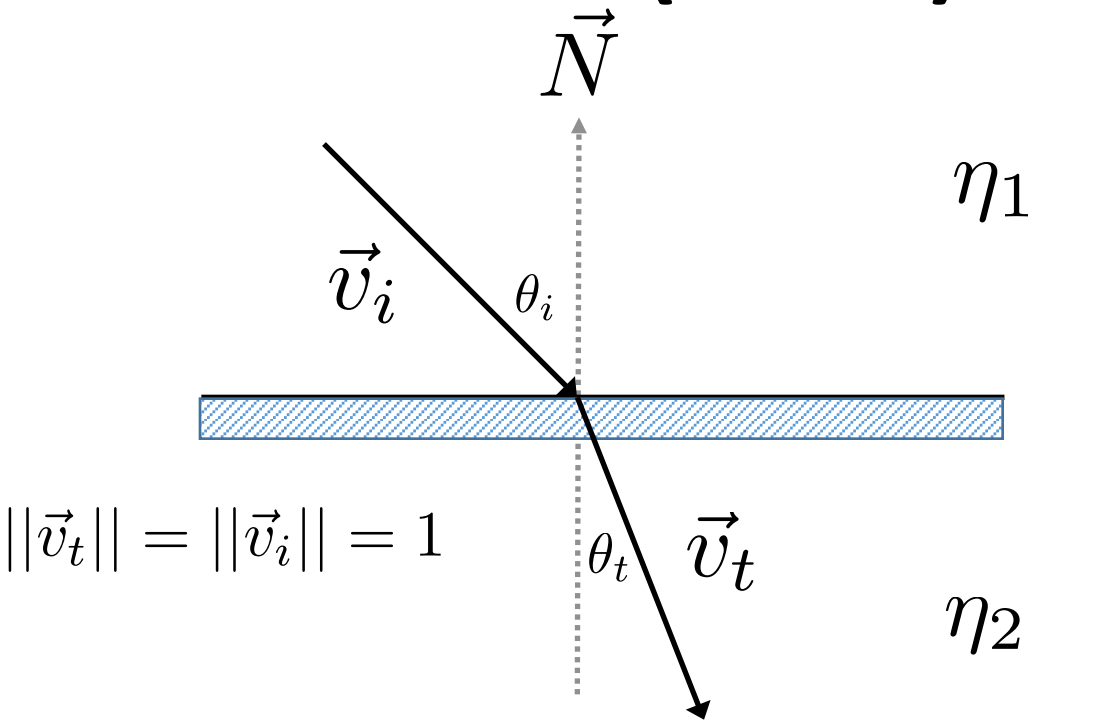
$$||\vec{v}_r|| = ||\vec{v}_i|| = 1$$

$$\vec{v}_r = \vec{v}_i - 2\left(\vec{v}_i \cdot \vec{N}\right)\vec{N}$$

Note the directions !

http://graphics.stanford.edu/courses/cs148-10-summer/docs/2006--degreve--reflection_refraction.pdf

Refraction Vector (Extra)

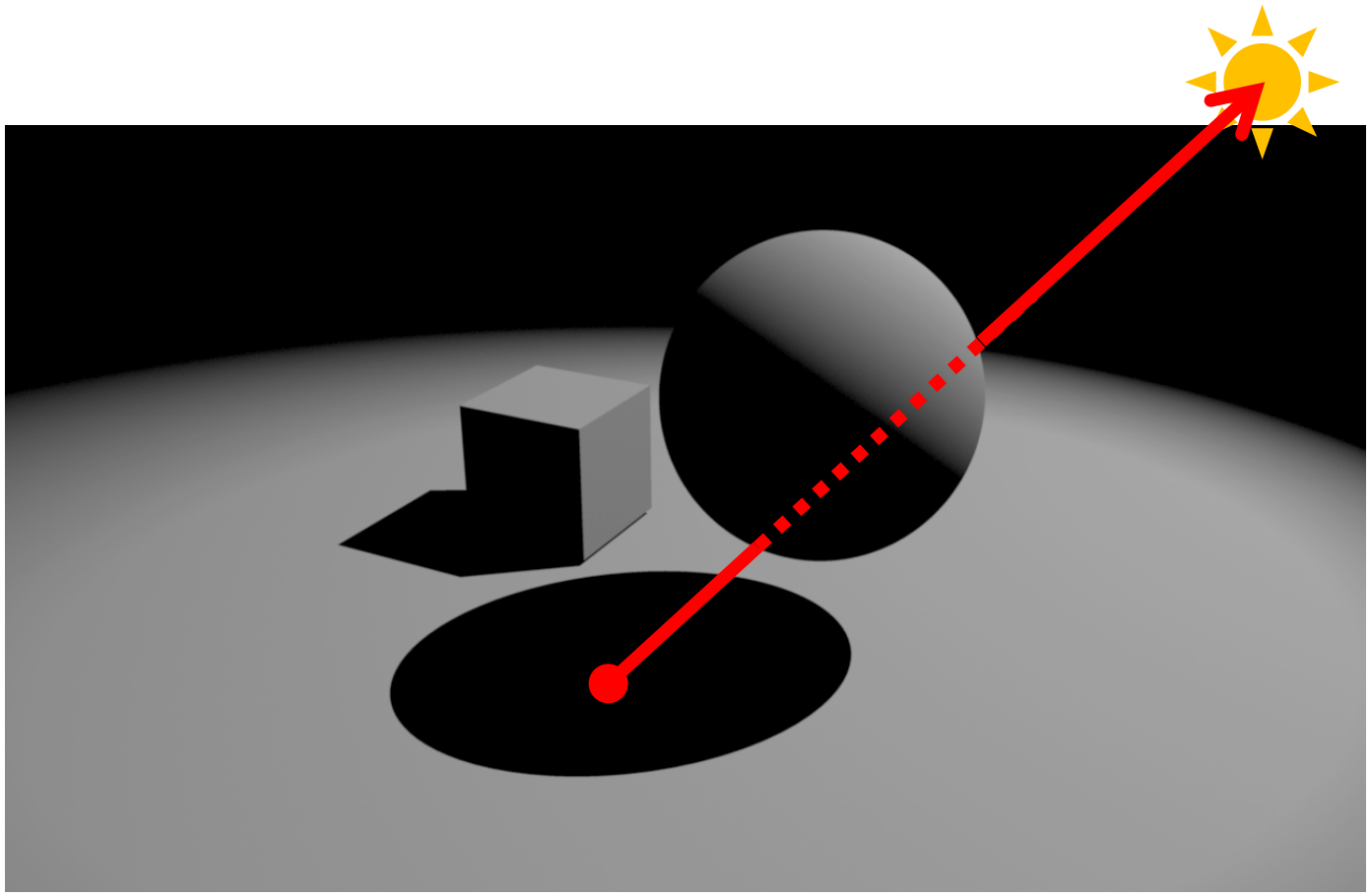


$$\vec{v}_t = \frac{\eta_1}{\eta_2}\vec{v}_i + \left(\frac{\eta_1}{\eta_2}\cos\theta_i - \sqrt{1-\sin^2\theta_t}\right)\vec{N}$$

$$\sin^2\theta_t = \left(\frac{\eta_1}{\eta_2}\right)^2\left(1-\cos^2\theta_i\right), \quad \cos\theta_i = -\vec{v}_i\cdot\vec{N}$$

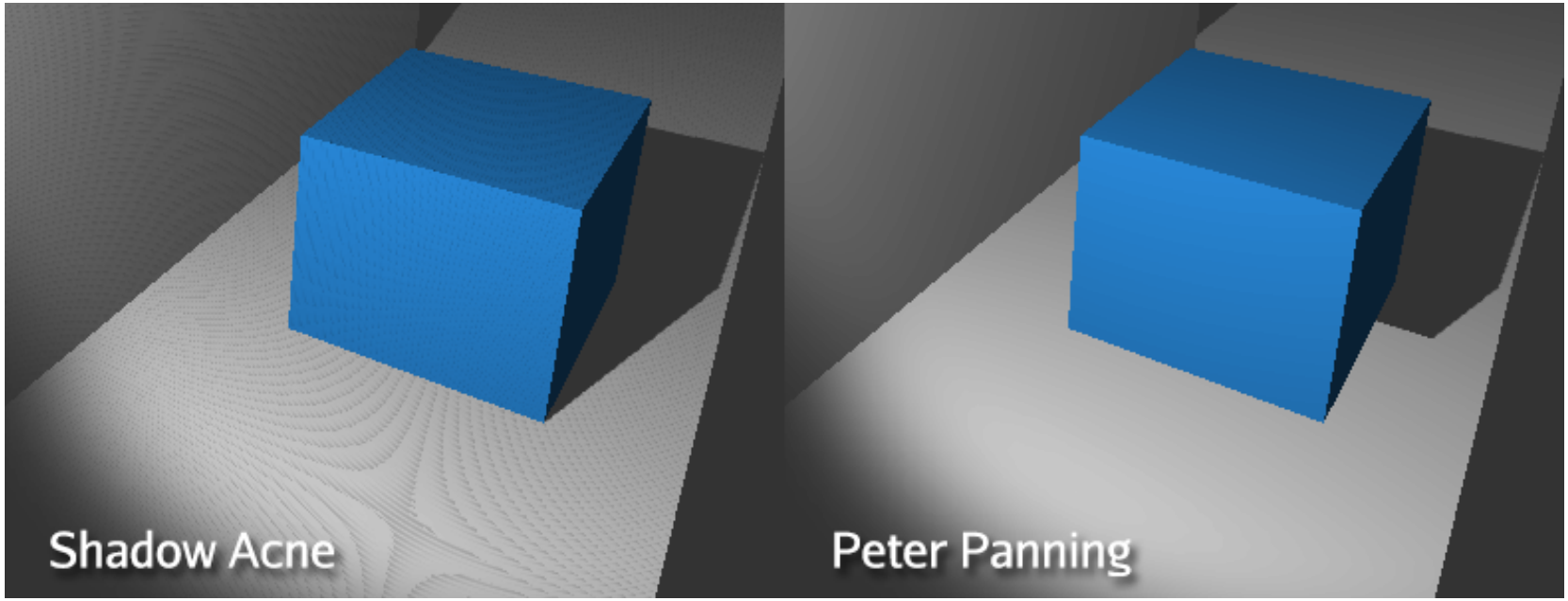
http://graphics.stanford.edu/courses/cs148-10-summer/docs/2006--degreve--reflection_refraction.pdf

Shadows



http://2.bp.blogspot.com/-2CpxoSdtNE8/T5nMClgobpI/AAAAAAAAAJ4/XgaZ168HpKg/s1600/P0008_I004.png

Shadow Acne / Peter Panning



Real life issues !

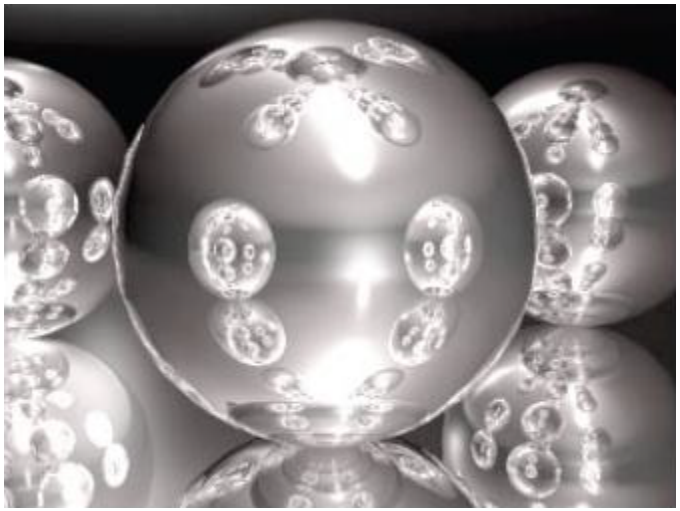
Object-Oriented Ray Tracing

```
abstract class surface
    virtual hit-record hit(ray, t0, t1)
    virtual box bounding-box()

struct hit-record
    vector3d position, normal
    surface hit-surface
    …

abstract class material
    color evaluate(in-ray, normal, pos)
    …
```

Recursive Ray Tracing

```
color ray-trace(ray r)
   hit-record record = intersect(r);
   …

   if (record.hit-surface.is-reflective)
      ray reflected-ray =
       reflect(r,record.normal);
      color reflected-color =
       ray-trace(reflected-ray);
      compute-spectral-component(
       reflected-color);
      …
```

Recursion Depth

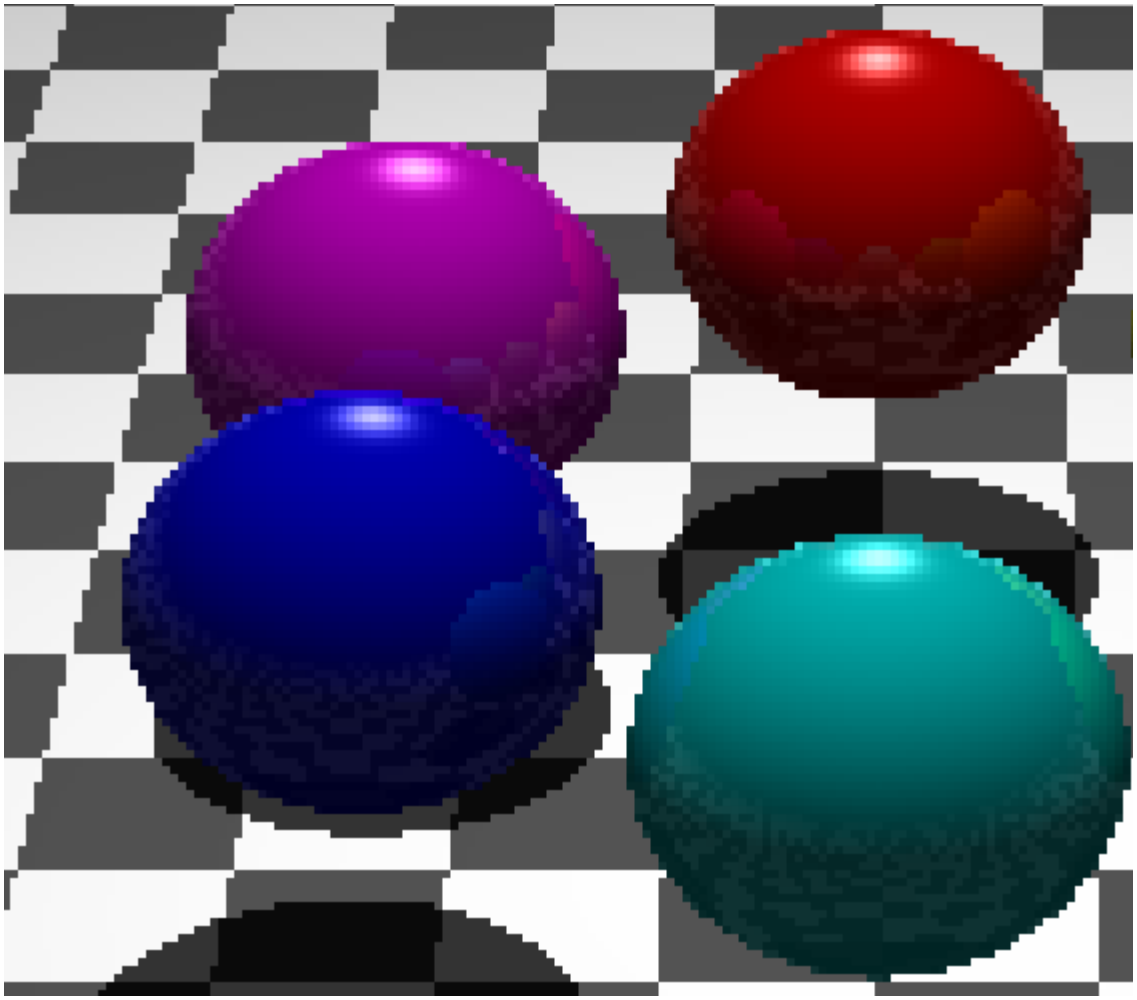


Lots of bounces

Fewer bounces

Avoid infinite recursion: stop after nth depth (typically n=5)

Aliasing: Still Problematic!

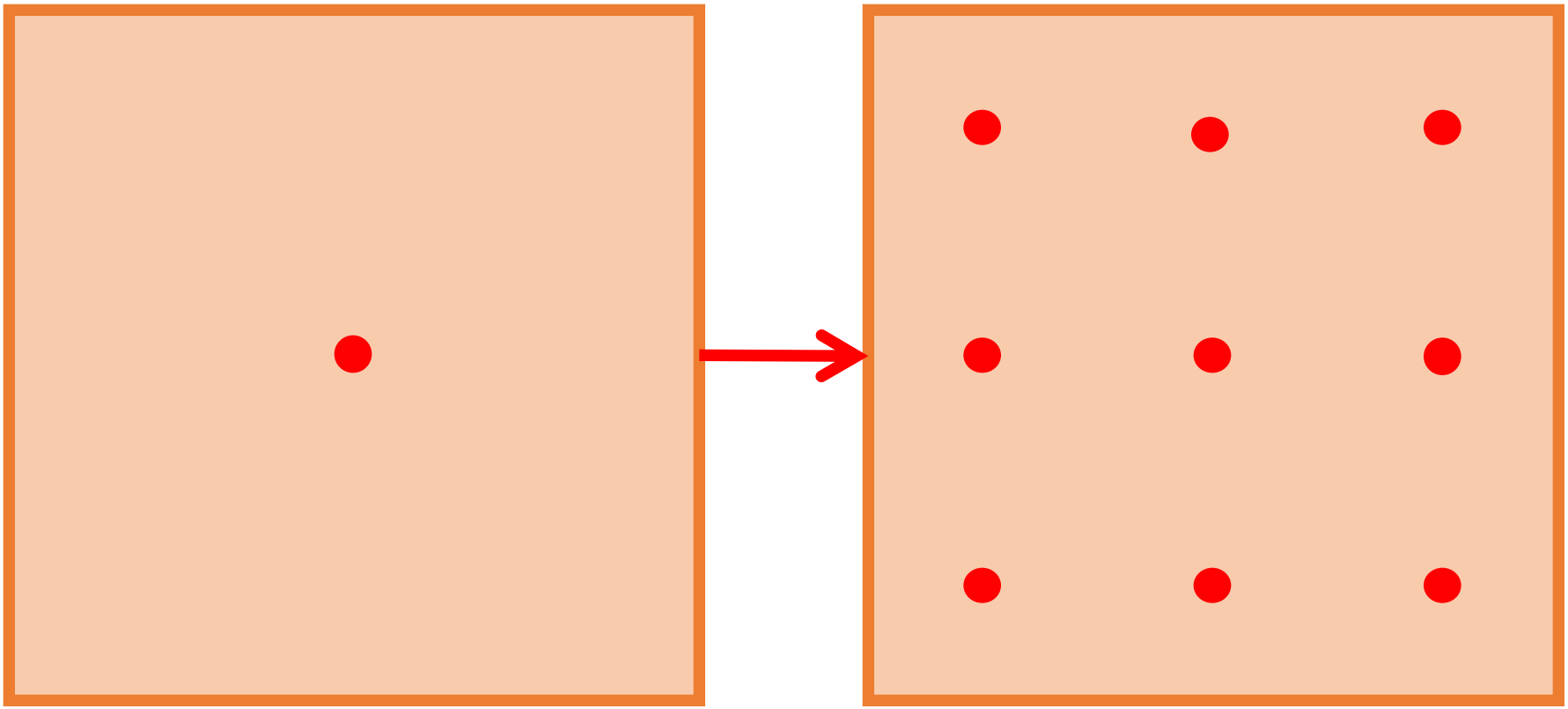


http://www.hackification.com/2008/08/31/experiments-in-ray-tracing-part-8-anti-aliasing/

Distribution Ray Tracing

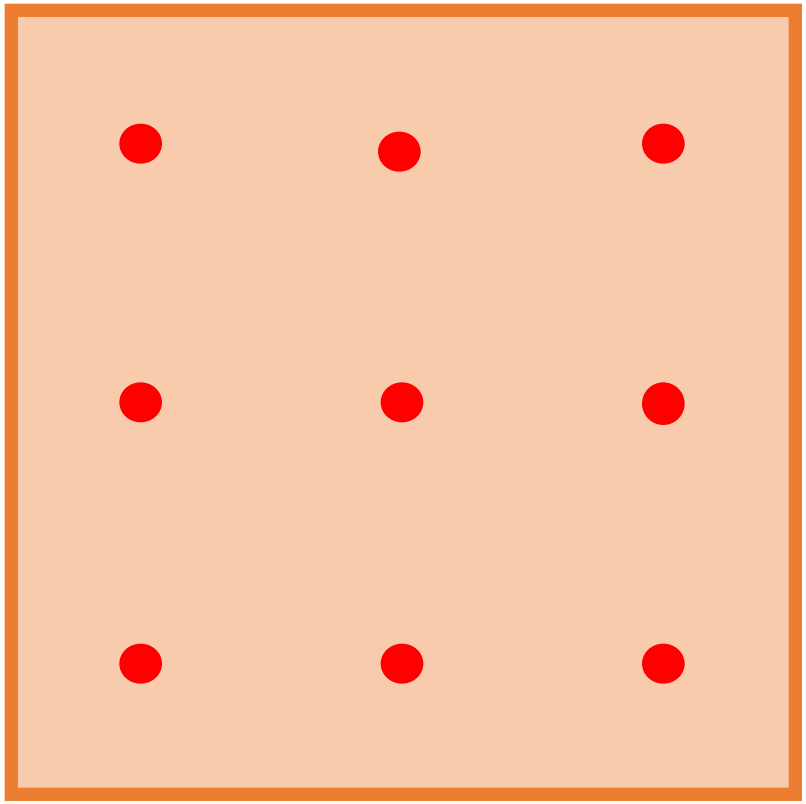
Scatter rays to make ray tracing less crisp.

Distribution Antialiasing



Multiple rays per pixel

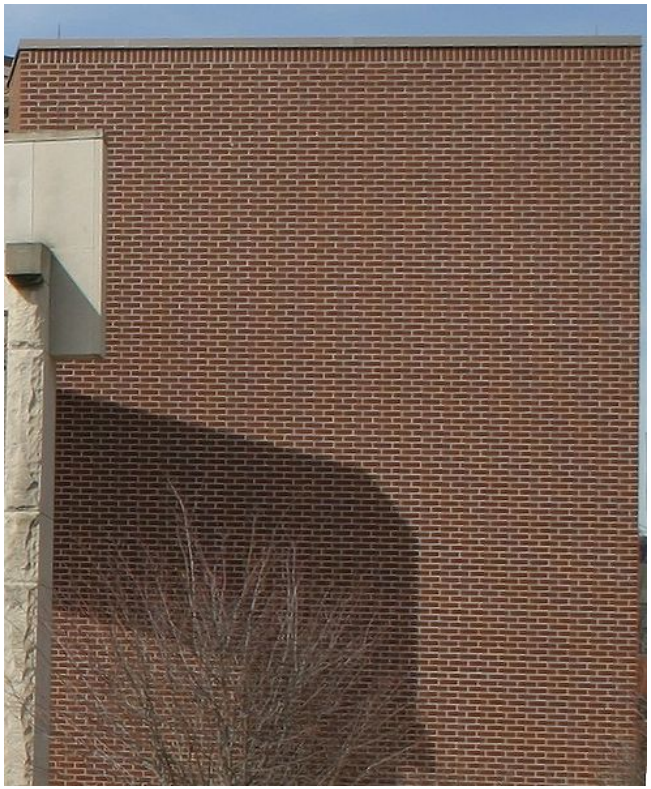
Distribution Antialiasing

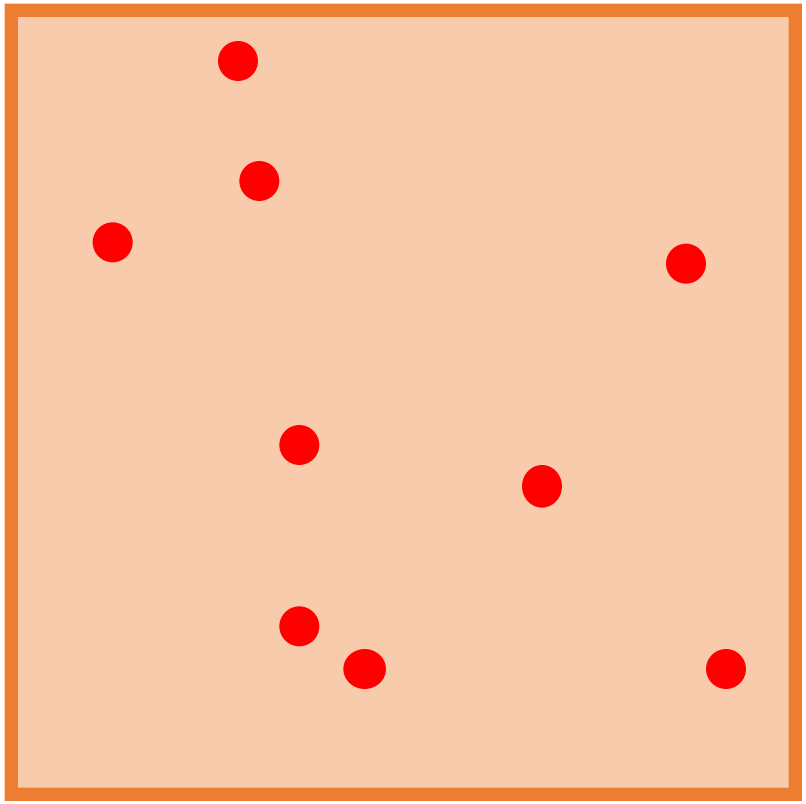


Patterned Noise

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

Random Sampling

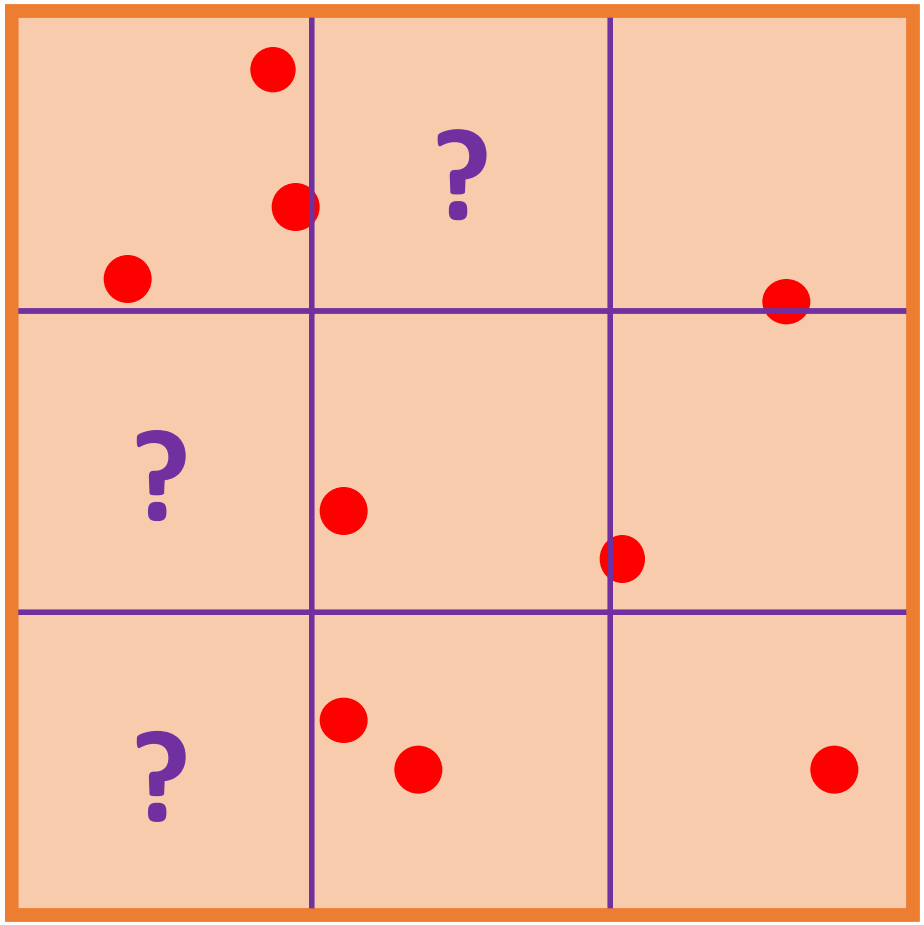




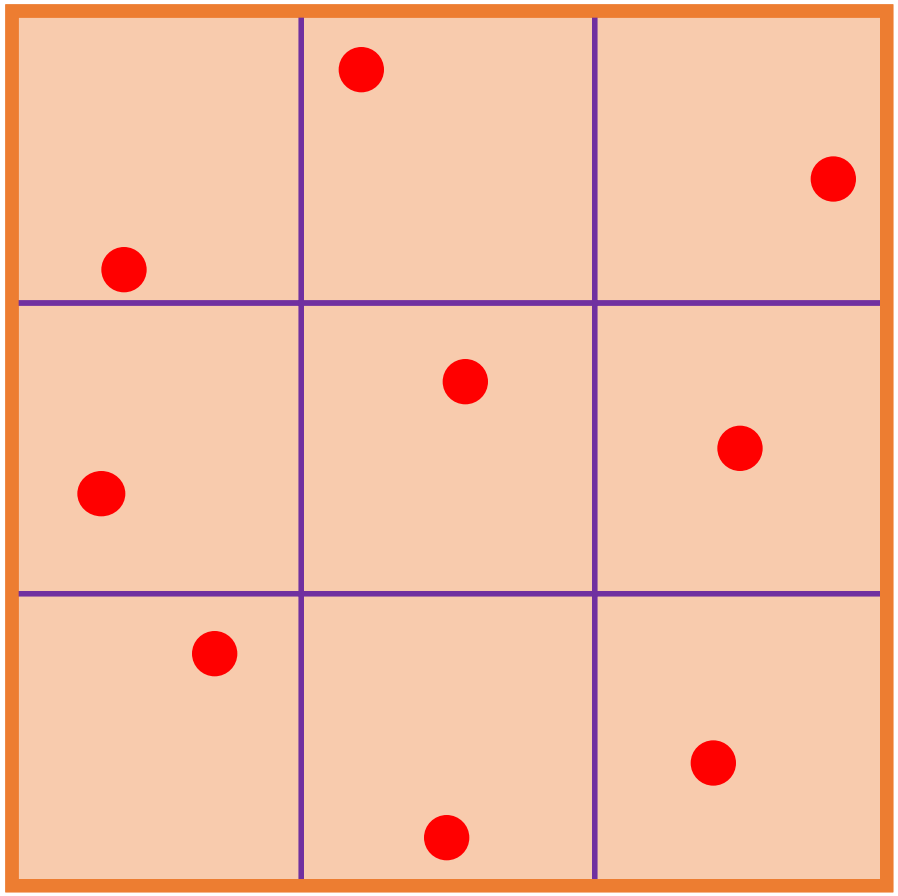
We are less sensitive to granular noise

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

Not Enough Samples!

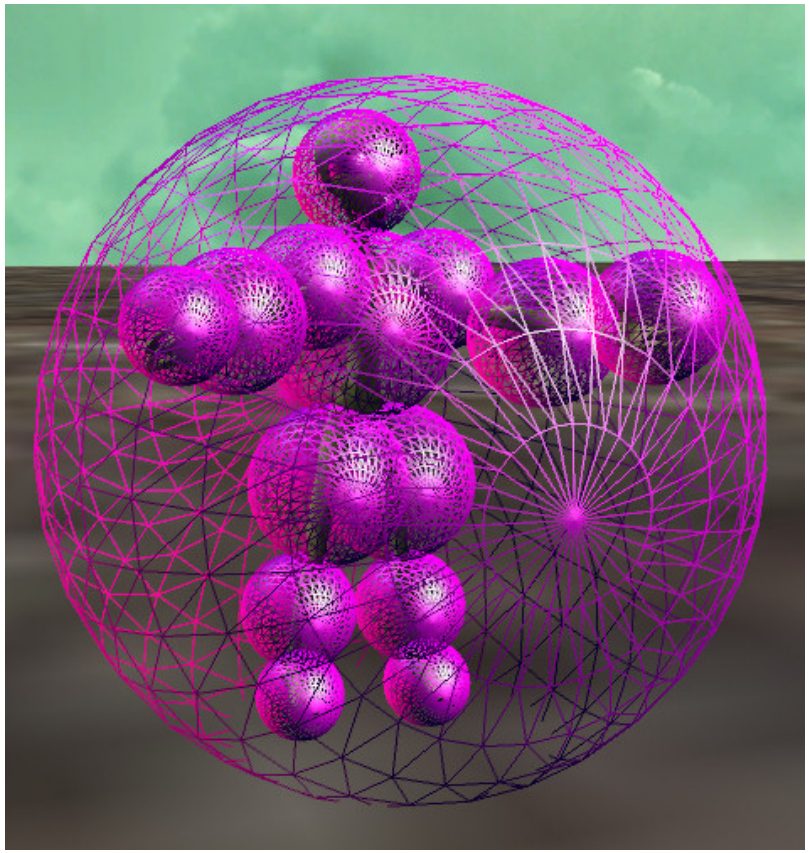


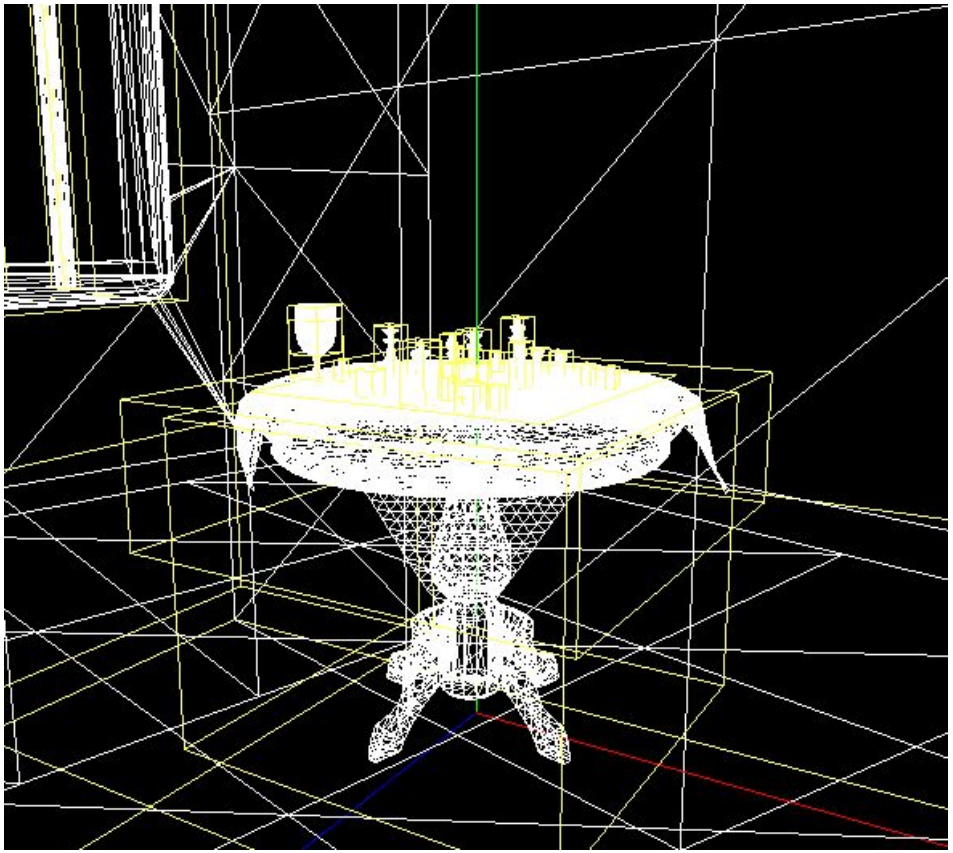
Stratified (Jittered) Sampling



One ray per box

Accelerated Ray Tracing

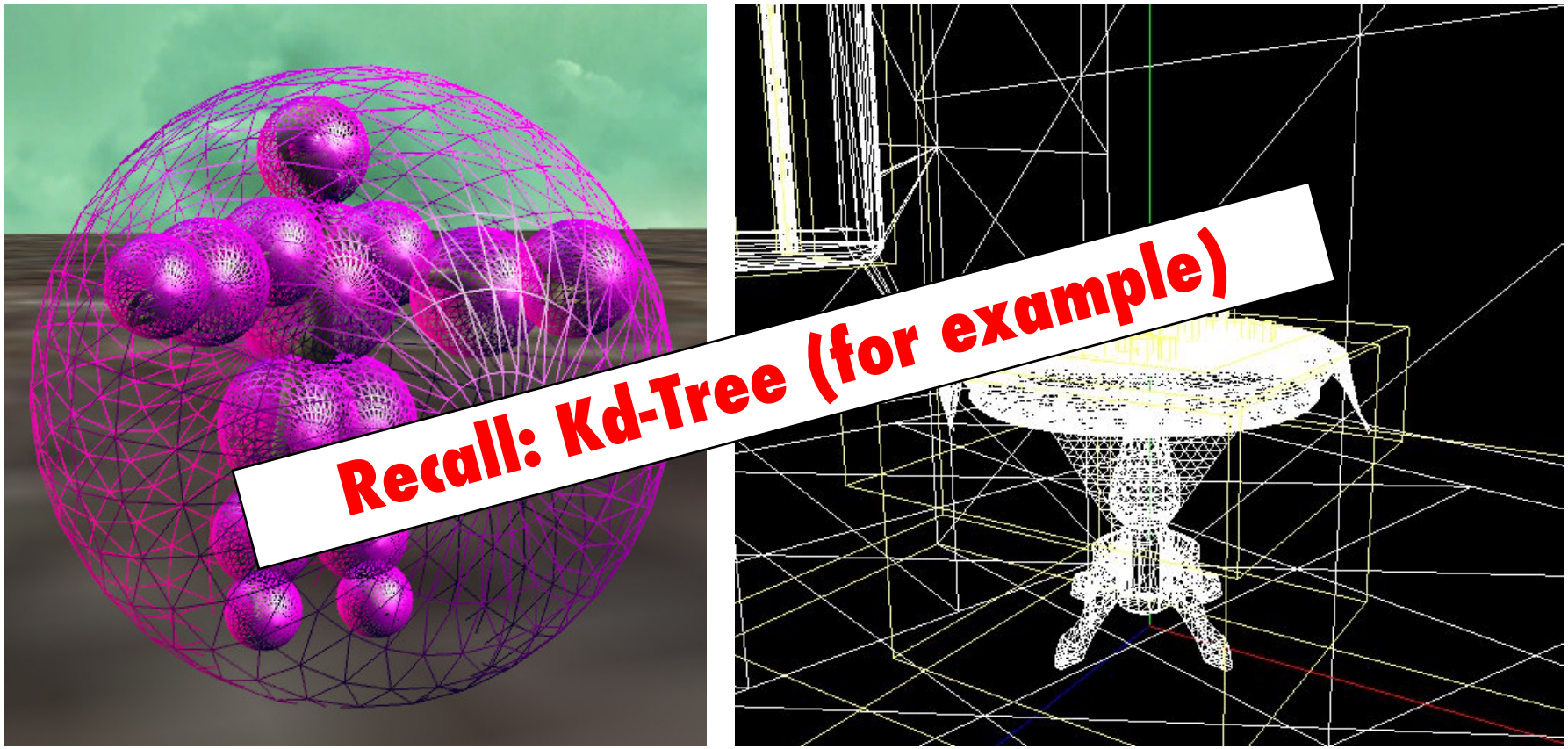




Bounding Volume Hierarchy (BVH)

http://sopra.le-gousteau.de/images/2/2e/Bv_bvh_bspheres.jpg
http://graphics.ucsd.edu/courses/rendering/2004/ssaha/index_files/finalbvh.jfif

Accelerated Ray Tracing



Bounding Volume Hierarchy (BVH)

http://sopra.le-gousteau.de/images/2/2e/Bv_bvh_bspheres.jpg
http://graphics.ucsd.edu/courses/rendering/2004/ssaha/index_files/finalbvh.jfif

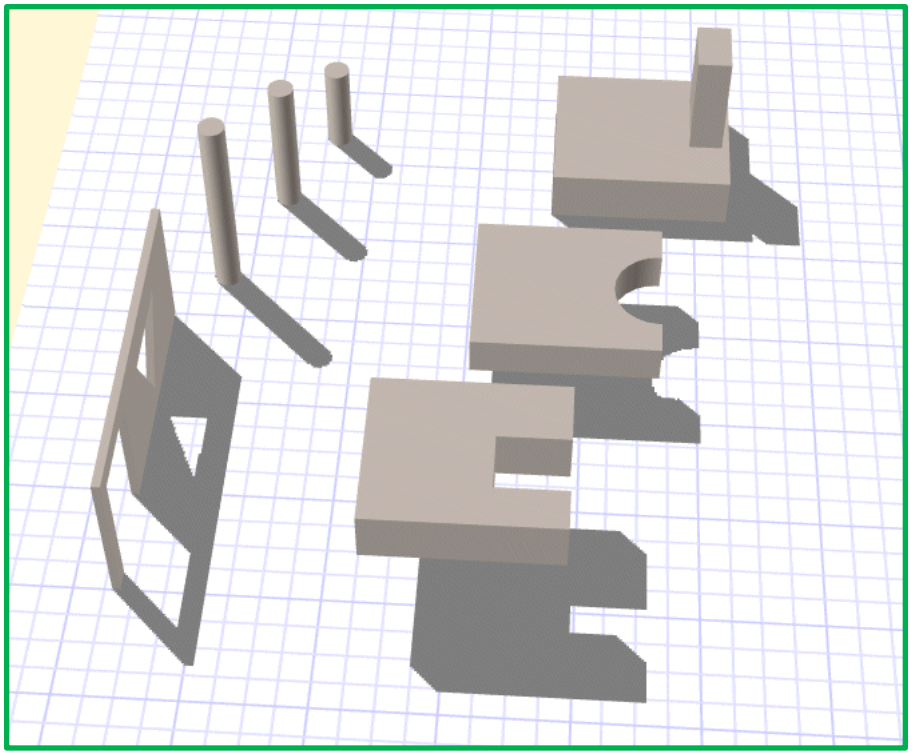
Asymptotic Speed

Value	Variable
p	Pixels on screen
b	Objects in scene
m	Average pixels/object

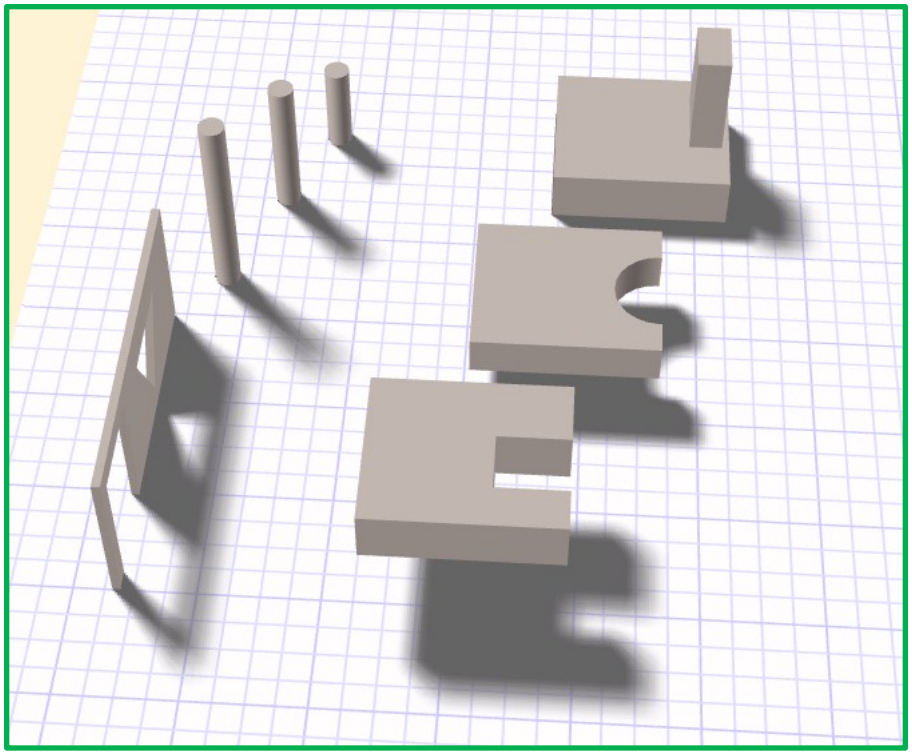
- **Rasterization:**
  $O(bm)$
- **Ray tracing:**
  $O(bp)$
- **Accelerated ray tracing:**
  $O(p \log b)$

Some Real-life Effects

Soft Shadows



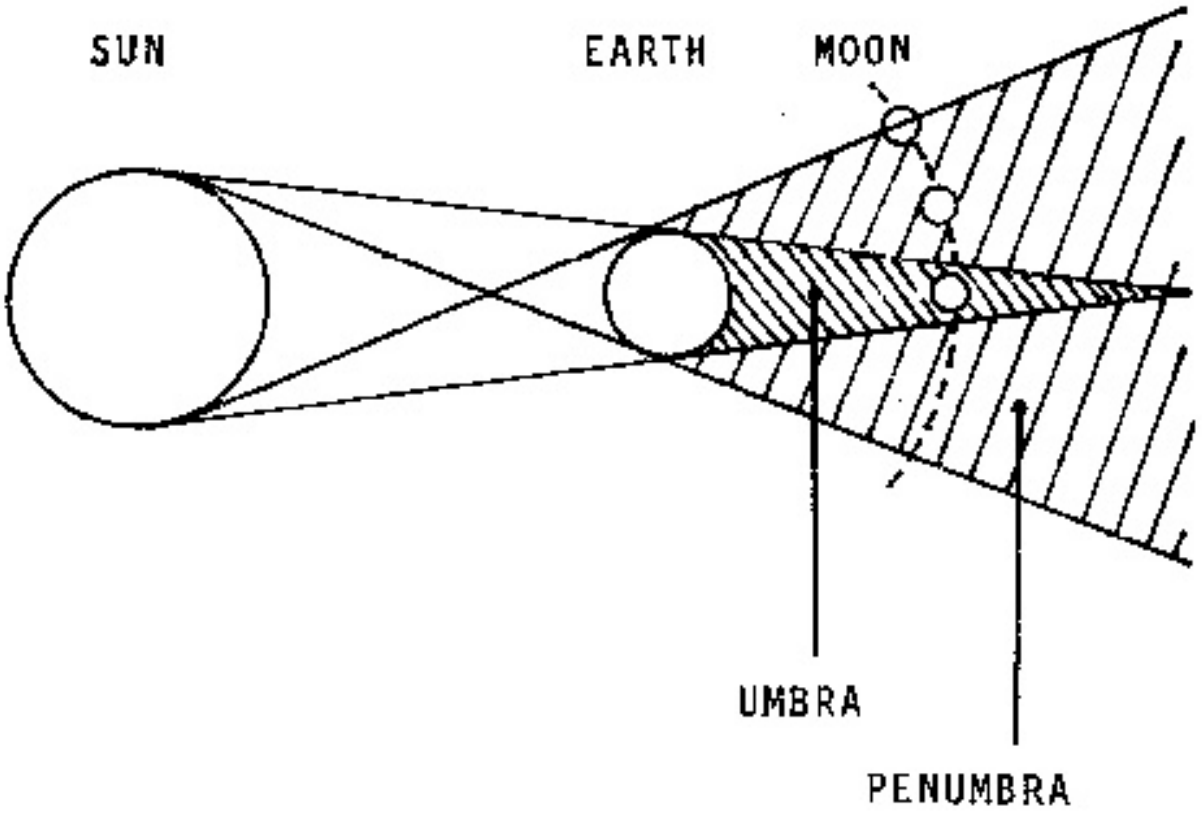
Hard shadows



Soft shadows

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

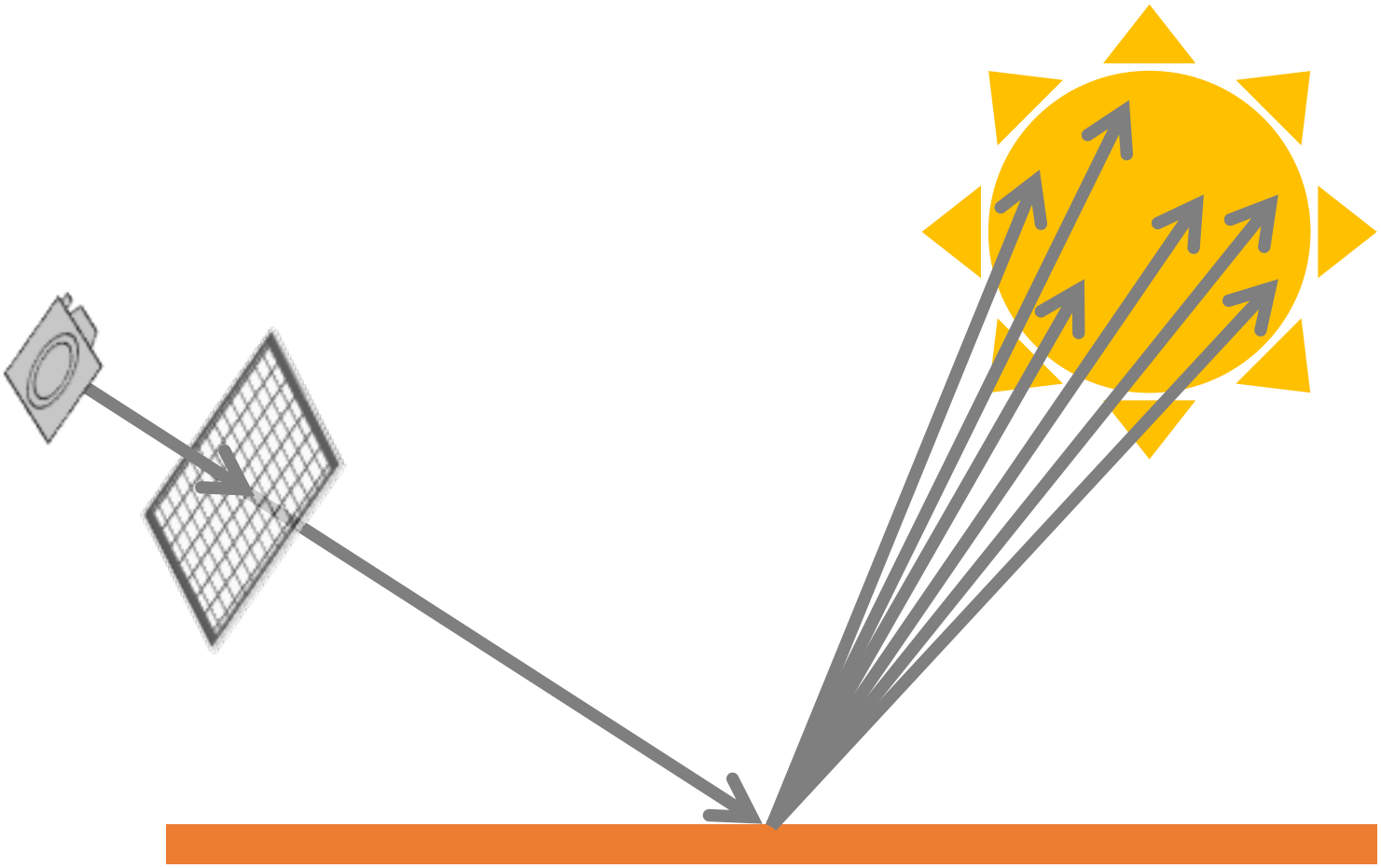
Soft Shadows



Lights aren't all point sources

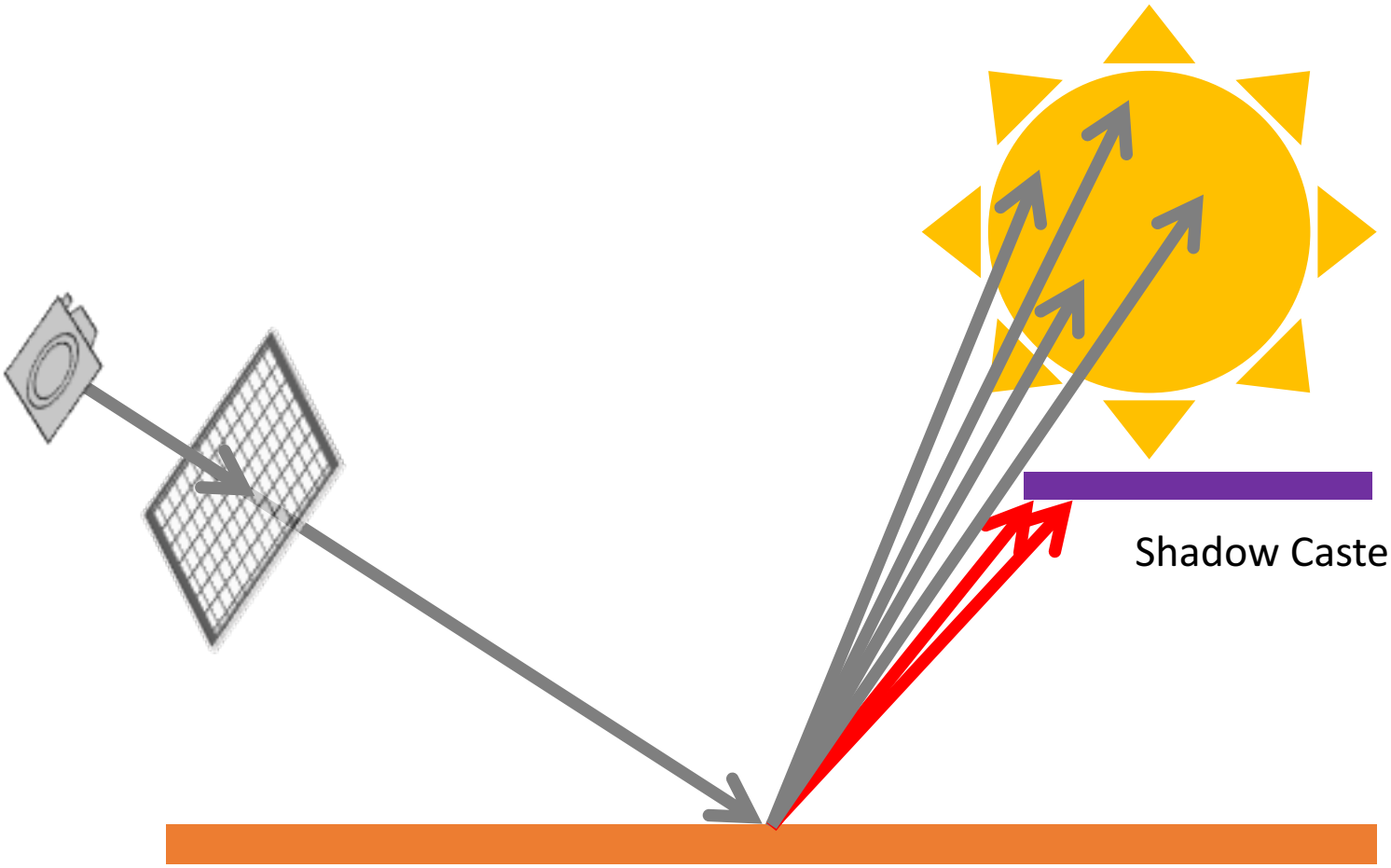
http://user.online.be/felixverbelen/lunecl.jpg

Distribution Soft Shadows



Randomly sample light rays: Area Light

Distribution Soft Shadows

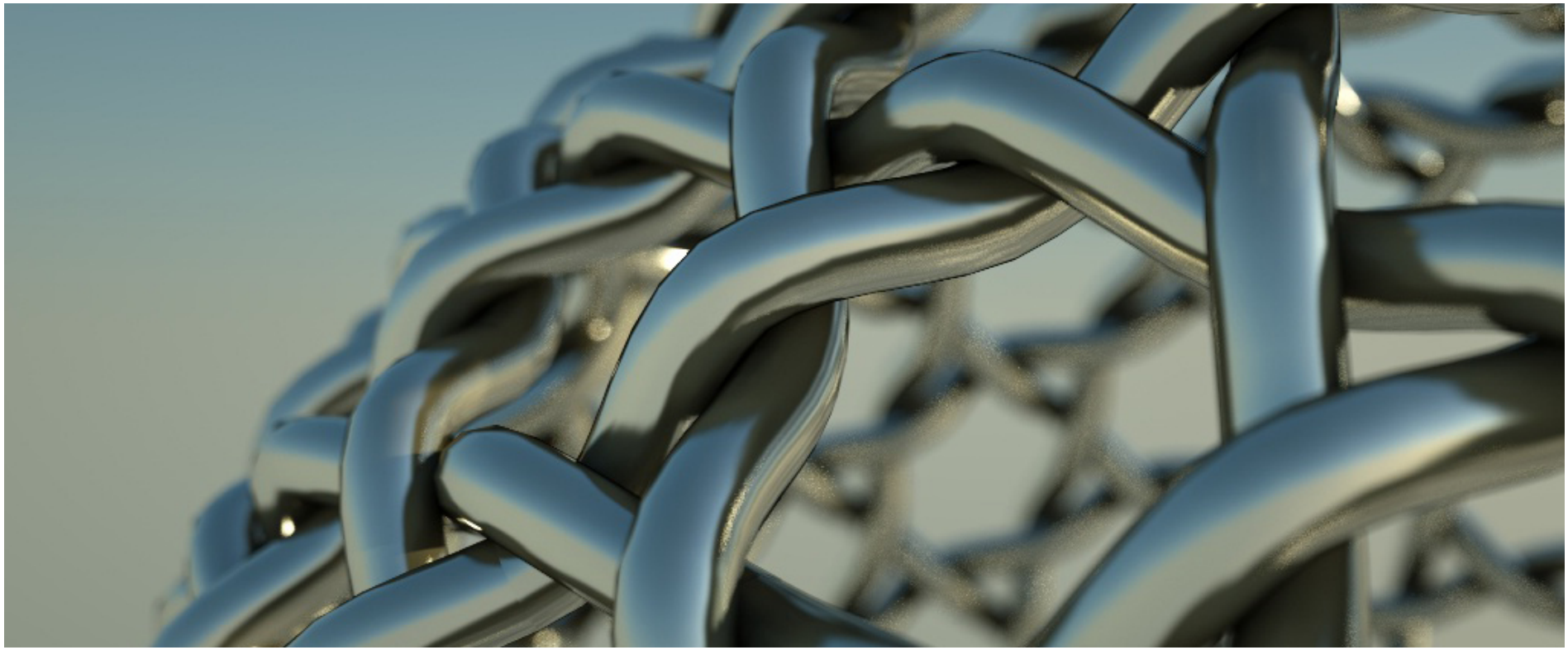


Shadow computed per ray: Average intensity

Distribution Soft Shadows

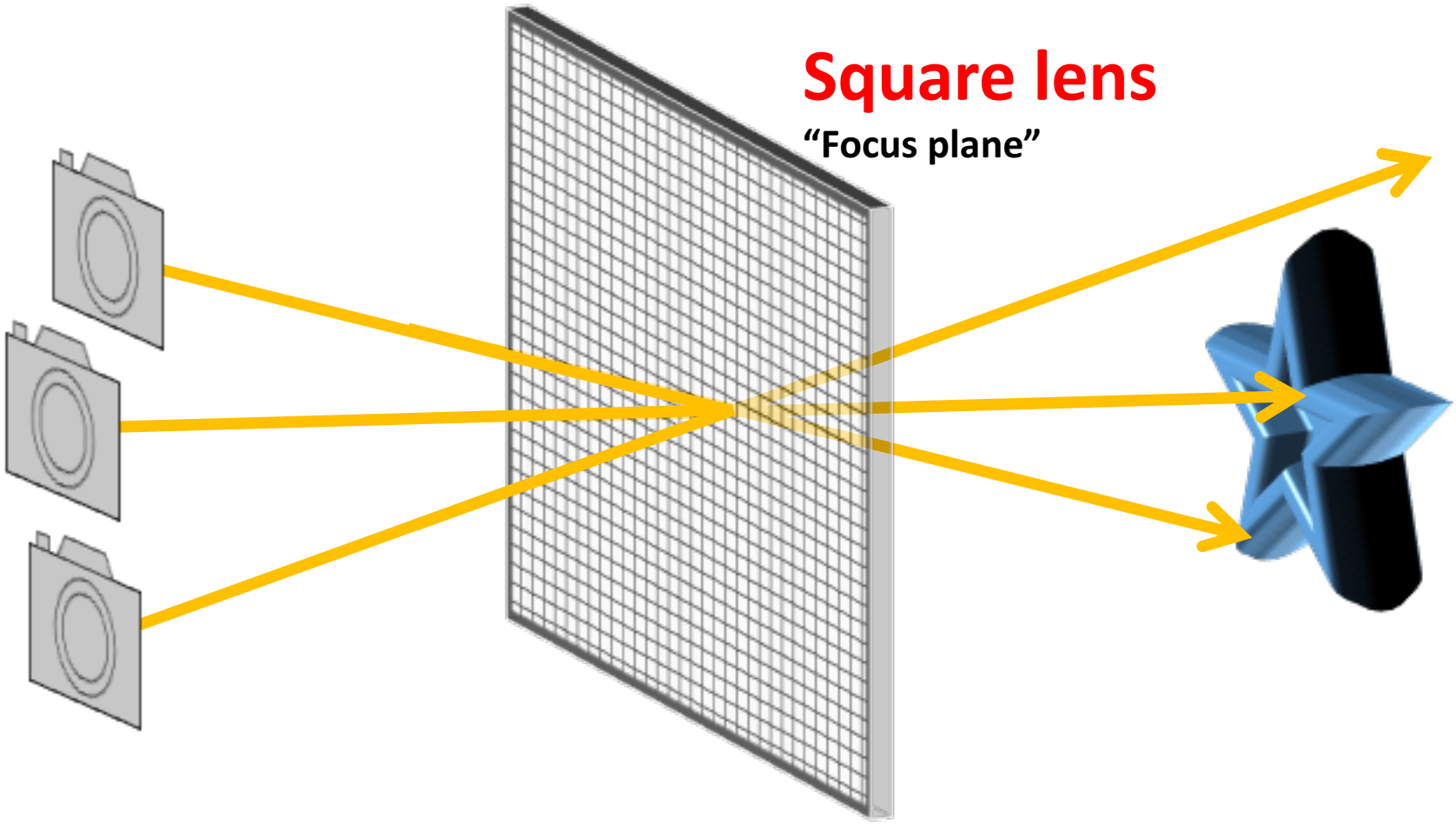
Shadow computed per ray: Average intensity

Depth of Field



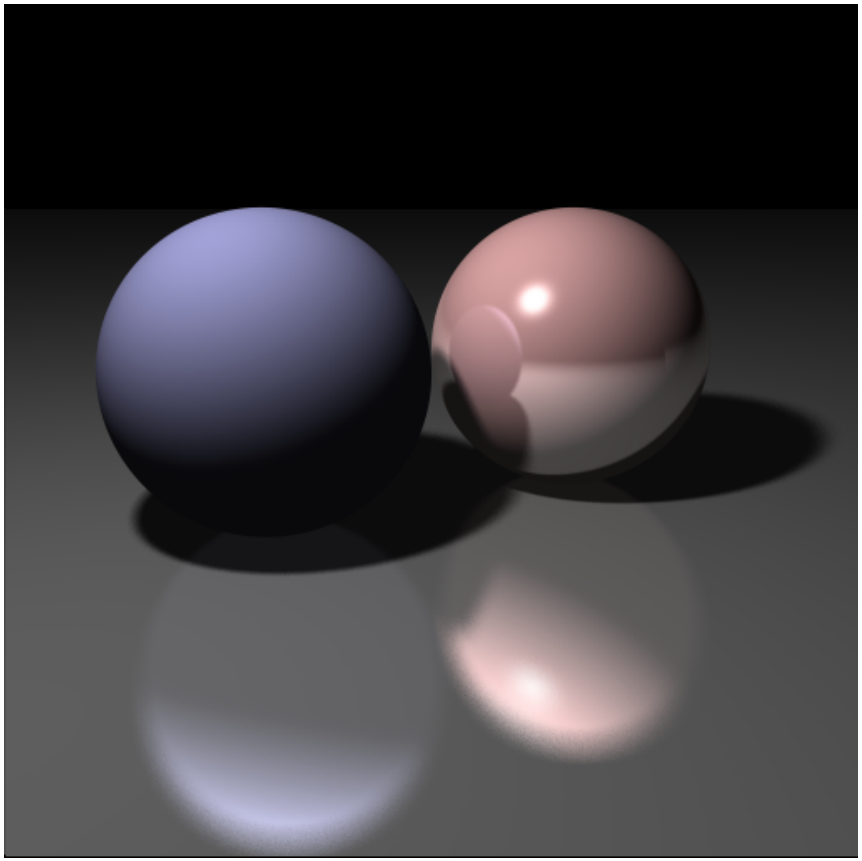
http://liam887.files.wordpress.com/2010/08/weaver.jpg

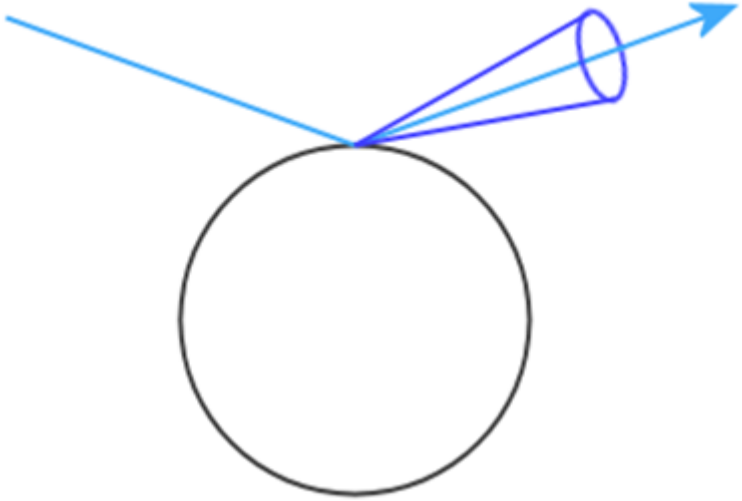
Distribution Depth of Field



Randomly sample eye positions

Distribution Glossy Reflection





https://graphics.stanford.edu/wikis/cs148-11-fall/RaytracingResults

http://www.baylee-online.net/Projects/Raytracing/Algorithms/Glossy-Reflection-Transmission

Randomly sample reflected rays

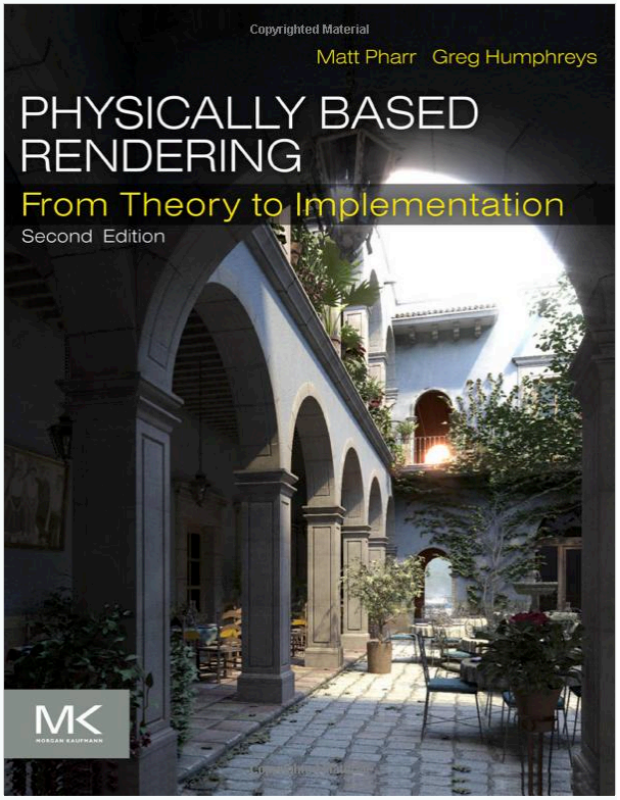
Motion Blur

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

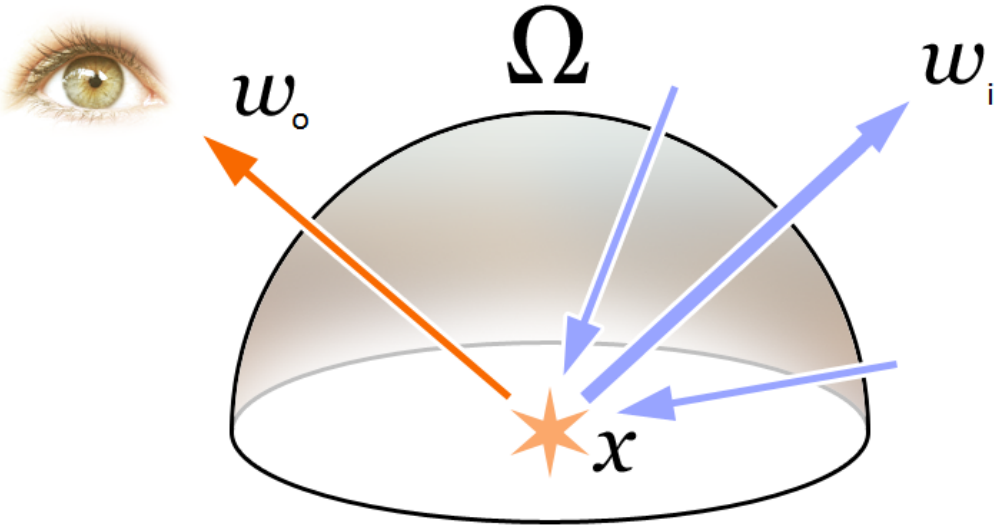
Randomly sample positions

Advanced Ray-Tracing

- Image Synthesis Technique: CS 348b

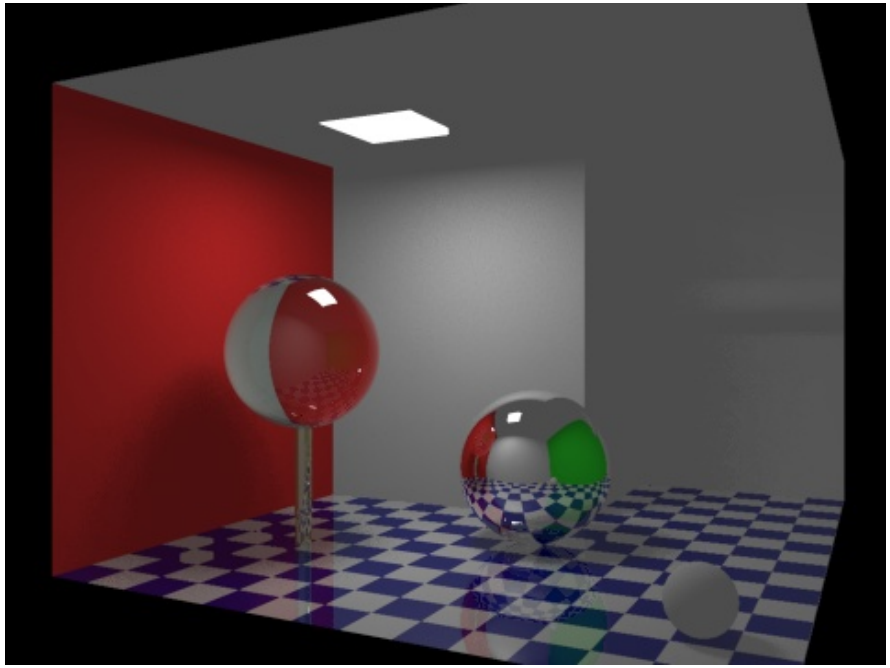


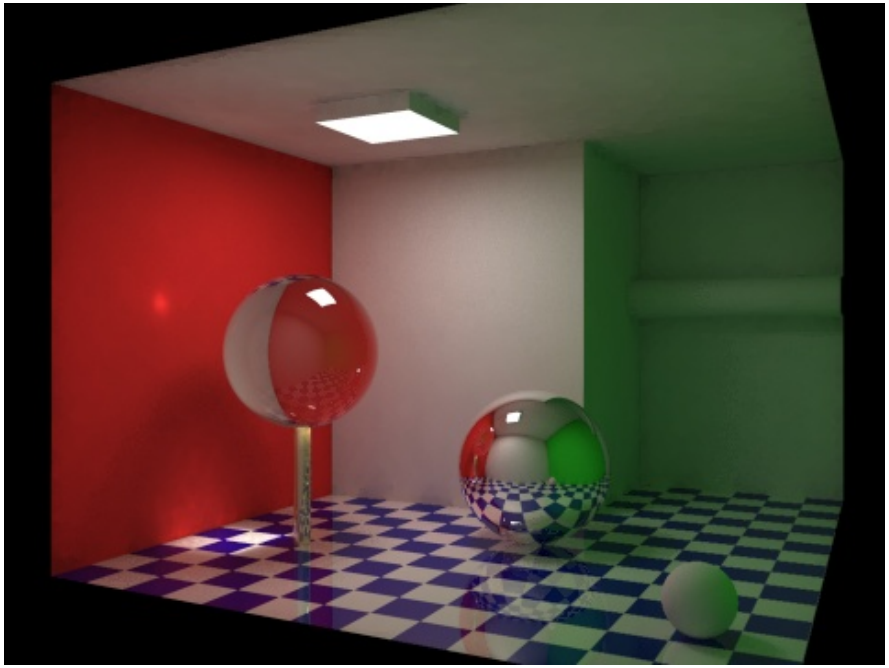
Rendering Equation (Holy Grail)



$$L_o(\vec{\omega}_o) = L_e(\vec{\omega}_o) + \int_\Omega f\left(\vec{\omega}_i, \vec{\omega}_o\right) L_i(\vec{\omega}_i) \underbrace{\left(\vec{n} \cdot \vec{w}_i\right)}_{\cos\theta} d\vec{\omega}_i$$

Global Illumination

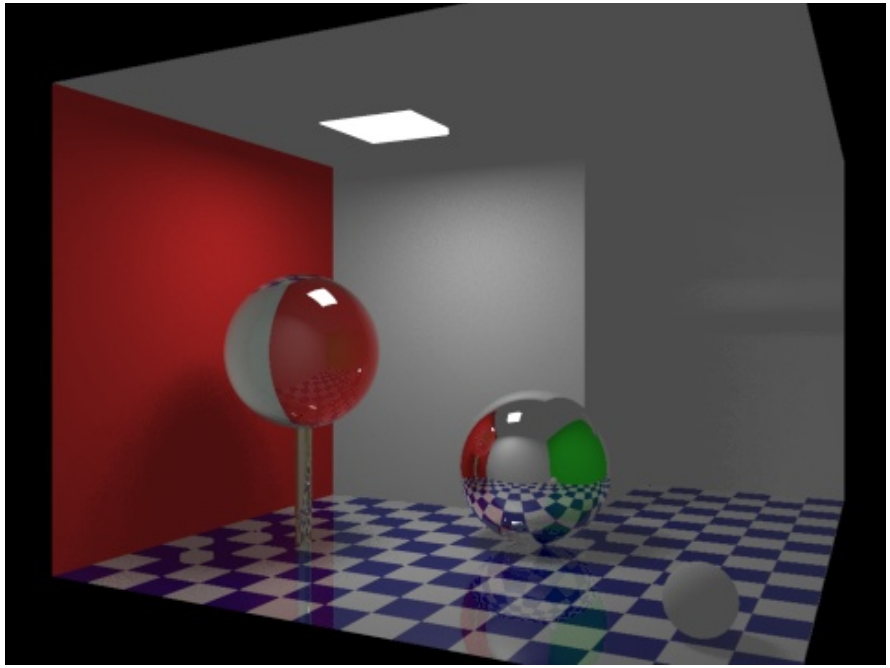


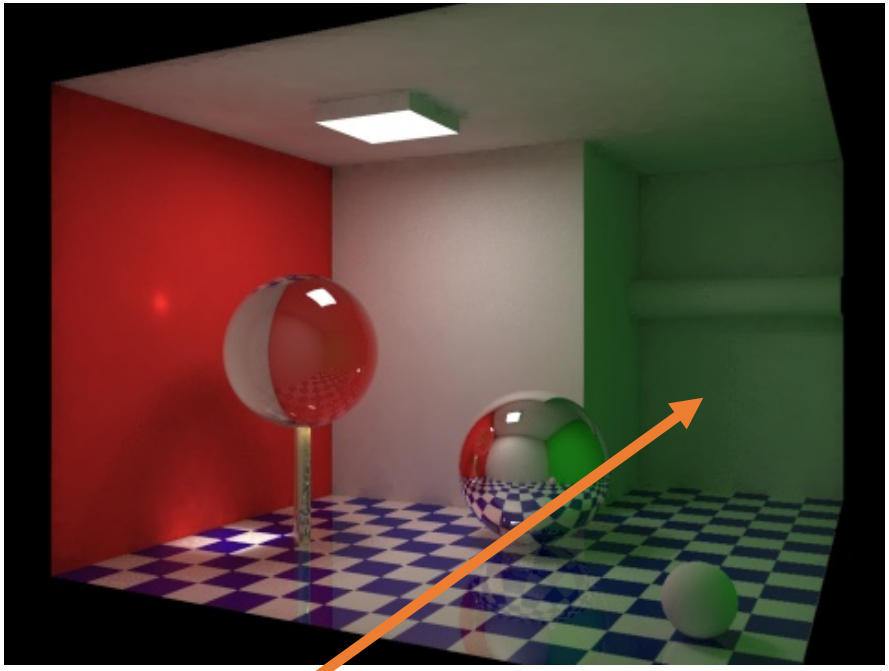


Account for indirect lighting

http://en.wikipedia.org/wiki/Global_illumination

Global Illumination

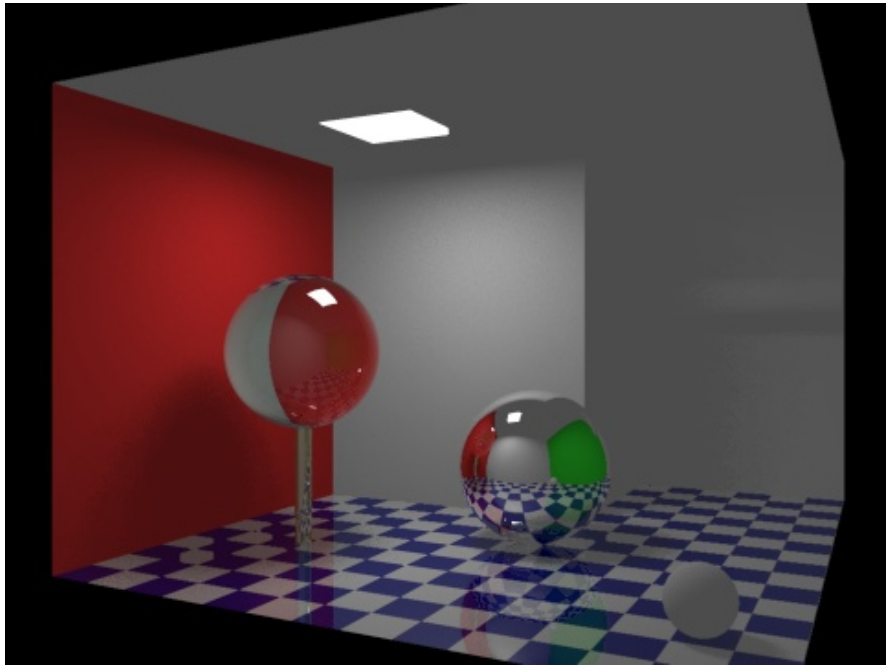


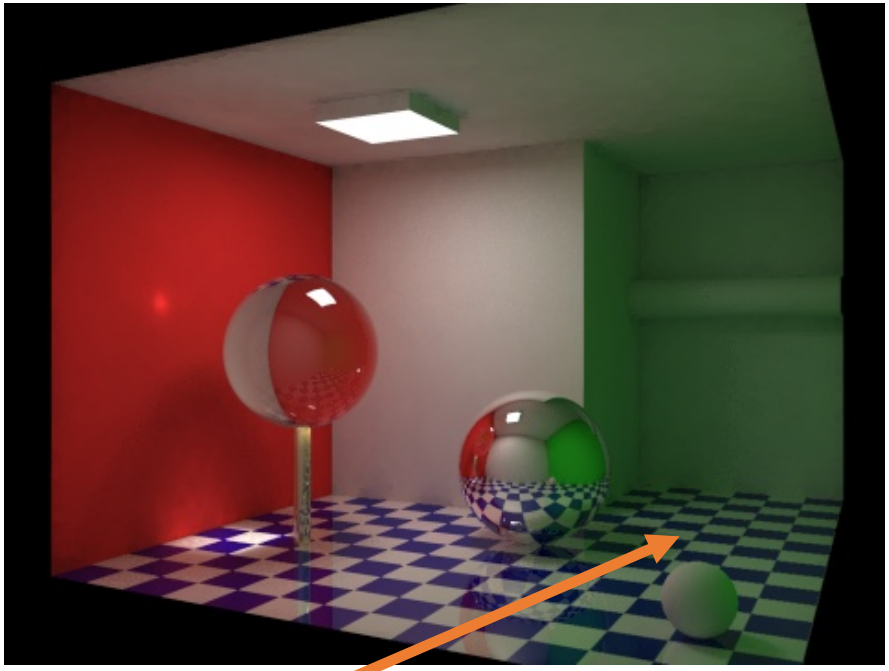


Diffuse inter-reflection

http://en.wikipedia.org/wiki/Global_illumination

Global Illumination

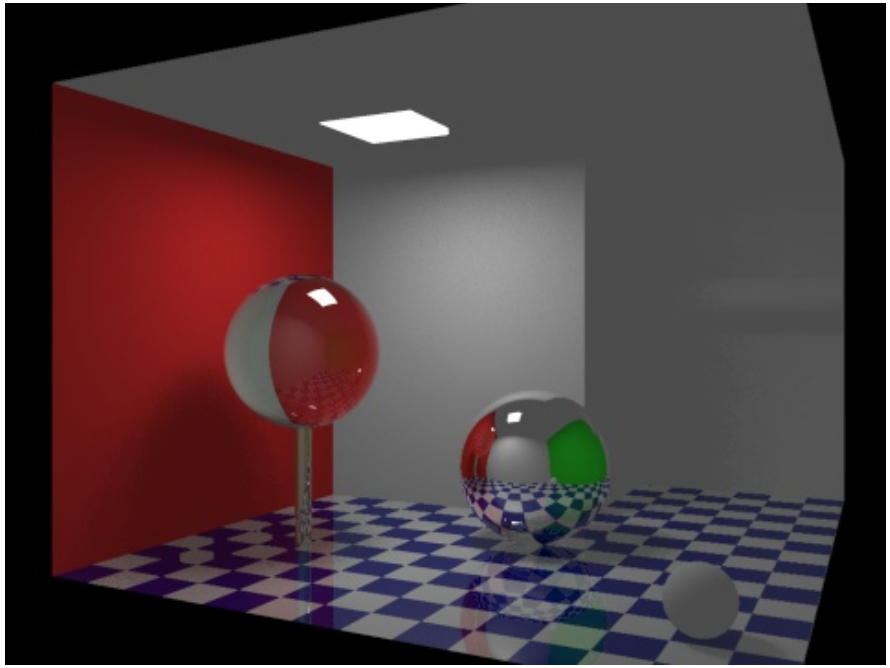


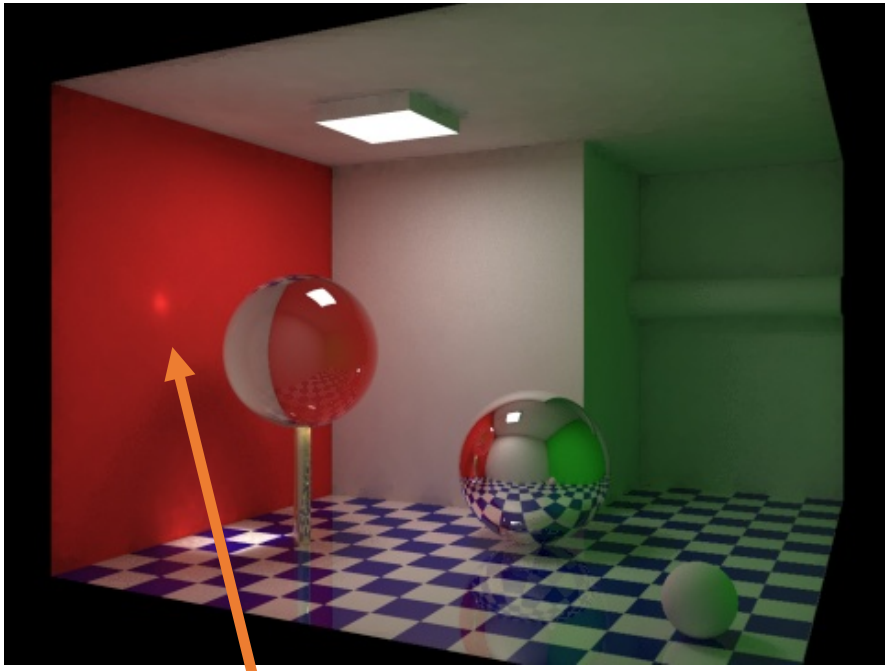


Color Bleed

http://en.wikipedia.org/wiki/Global_illumination

Global Illumination

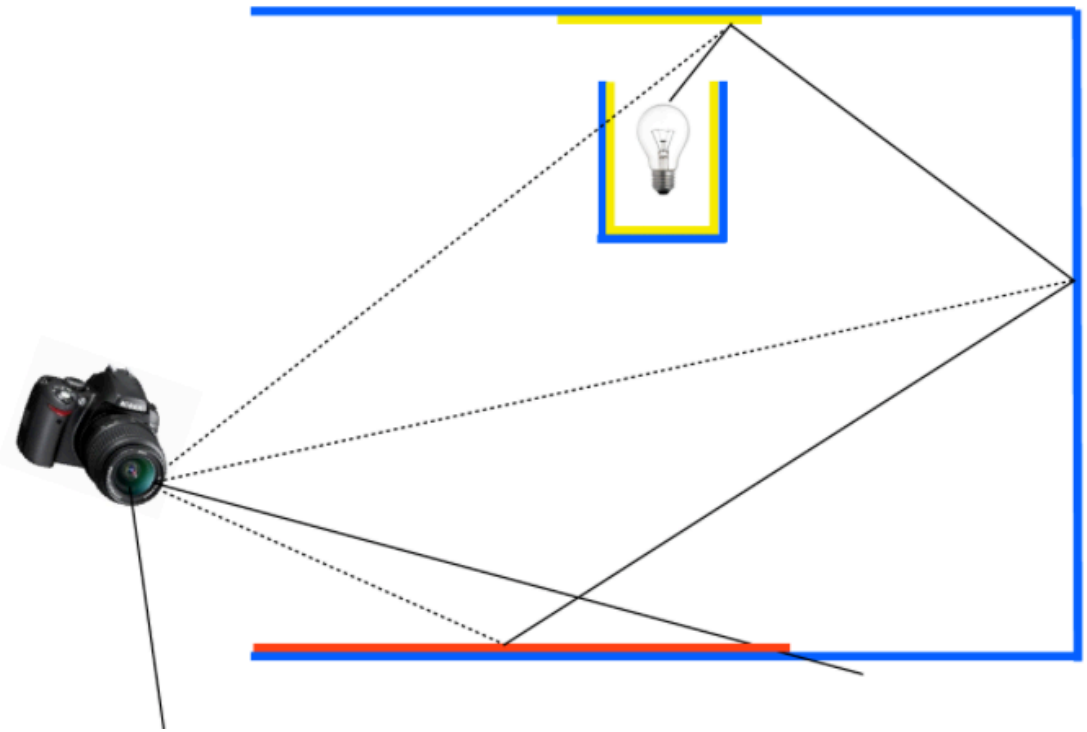




Caustics

http://en.wikipedia.org/wiki/Global_illumination

Difficult Paths



Bi-Directional Path Tracing

http://candela.stanford.edu/cs348b-14/doku.php?id=lectures:lecture17:slide6

Bi-directional Path Tracing

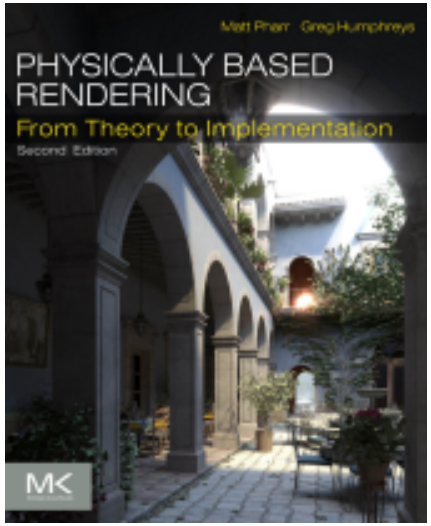
Challenge: Real-Time Ray Tracing

"Embarrassingly parallel," but not SIMD?

[NVIDIA, SIGGRAPH 2008]

Remarks

- Debugging: One solution
  - https://github.com/zdevito/vdb
- PBRT : Physically Based Rendering Toolkit
  - http://www.pbrt.org



CS 348b

Ray Tracing

CS 148: Summer 2016
Introduction of Graphics and Imaging
Zahid Hossain

https://en.wikipedia.org/wiki/Ray_tracing_%28graphics%29