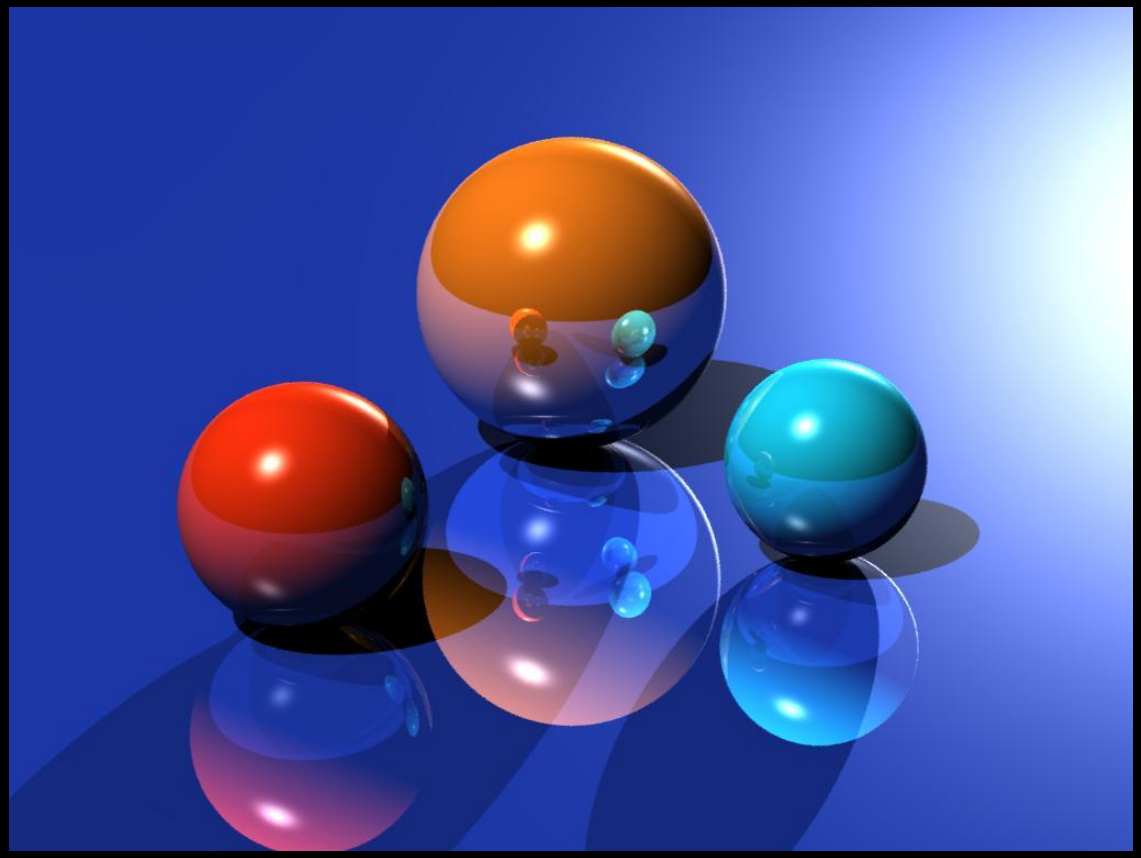
http://www.futuretech.blinkenlights.nl/misc/scene_1024x768.jpg



Ray Tracing

CS 148, Summer 2012
Introduction to Computer Graphics and Imaging

Justin Solomon

Recall:

Two Ways to Make an Image

1. Drawing

2. Photography

http://www.publicdomainpictures.net/view-image.php?picture=female-photographer&image=4360&large=1

http://bloodredpencil.blogspot.com/2010/09/need-help-get-help.html

Recall:

Two Ways to Make an Image

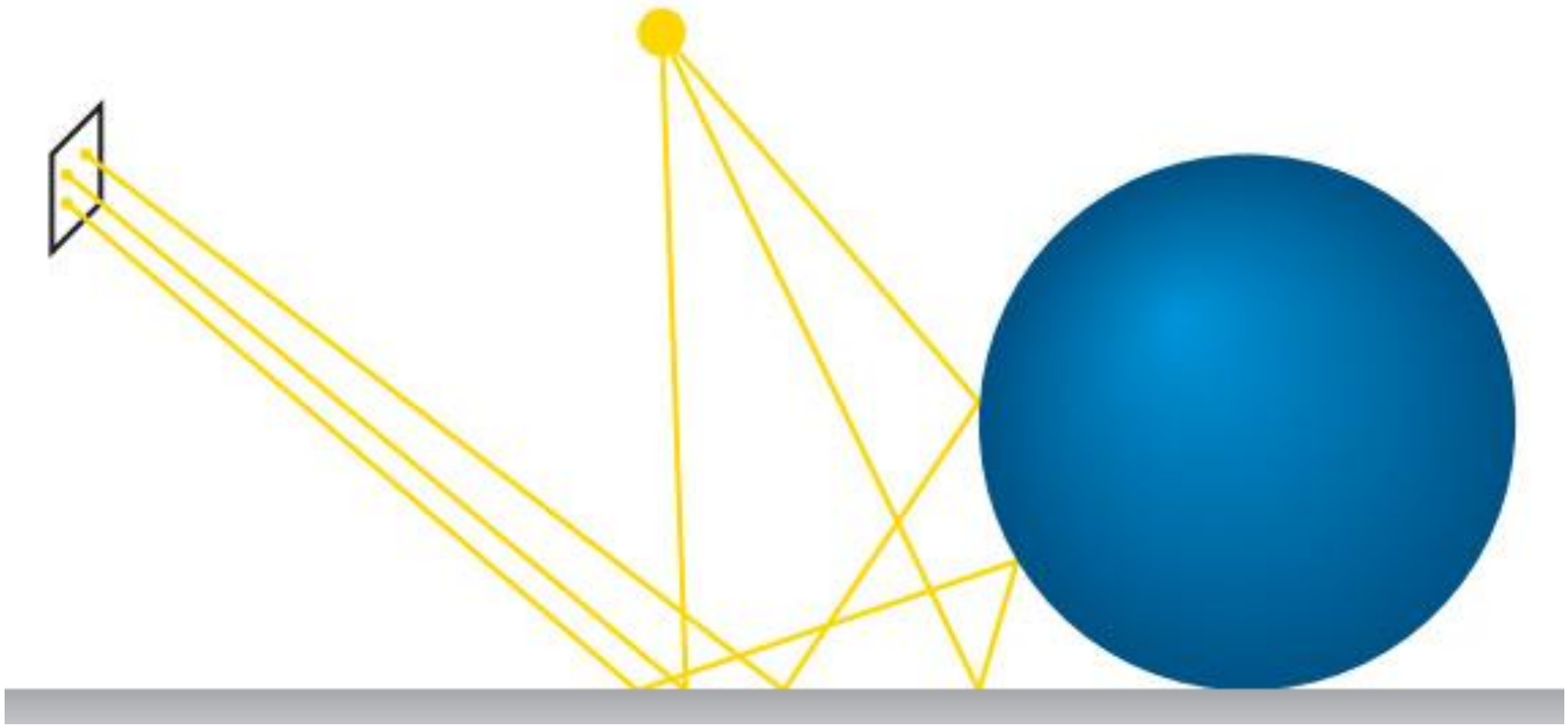
1. Drawing

2. Photography

http://www.publicdomainpictures.net/view-image.php?picture=female-photographer&image=4360&large=1

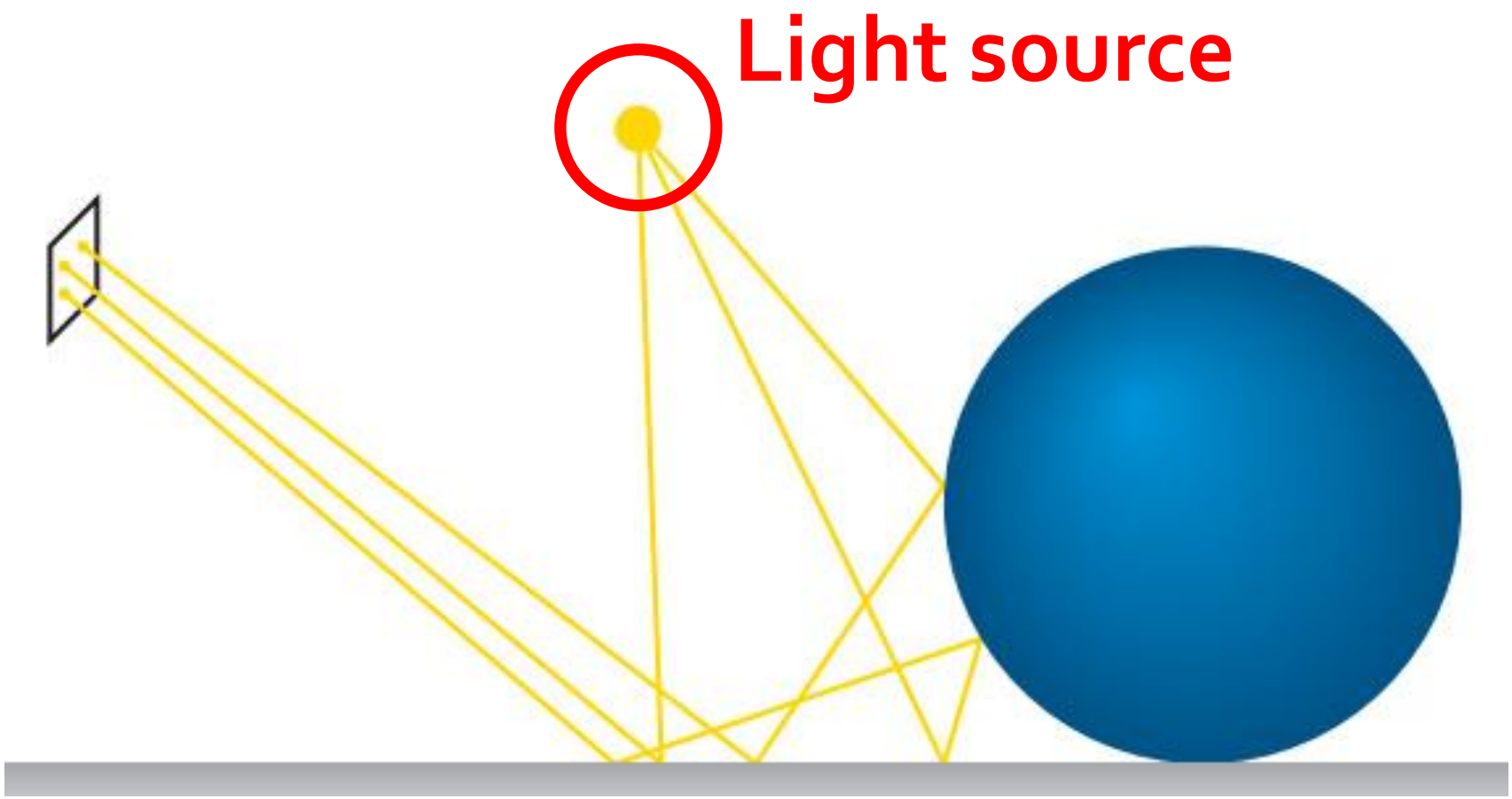
http://bloodredpencil.blogspot.com/2010/09/need-help-get-help.html

Paths of Light



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

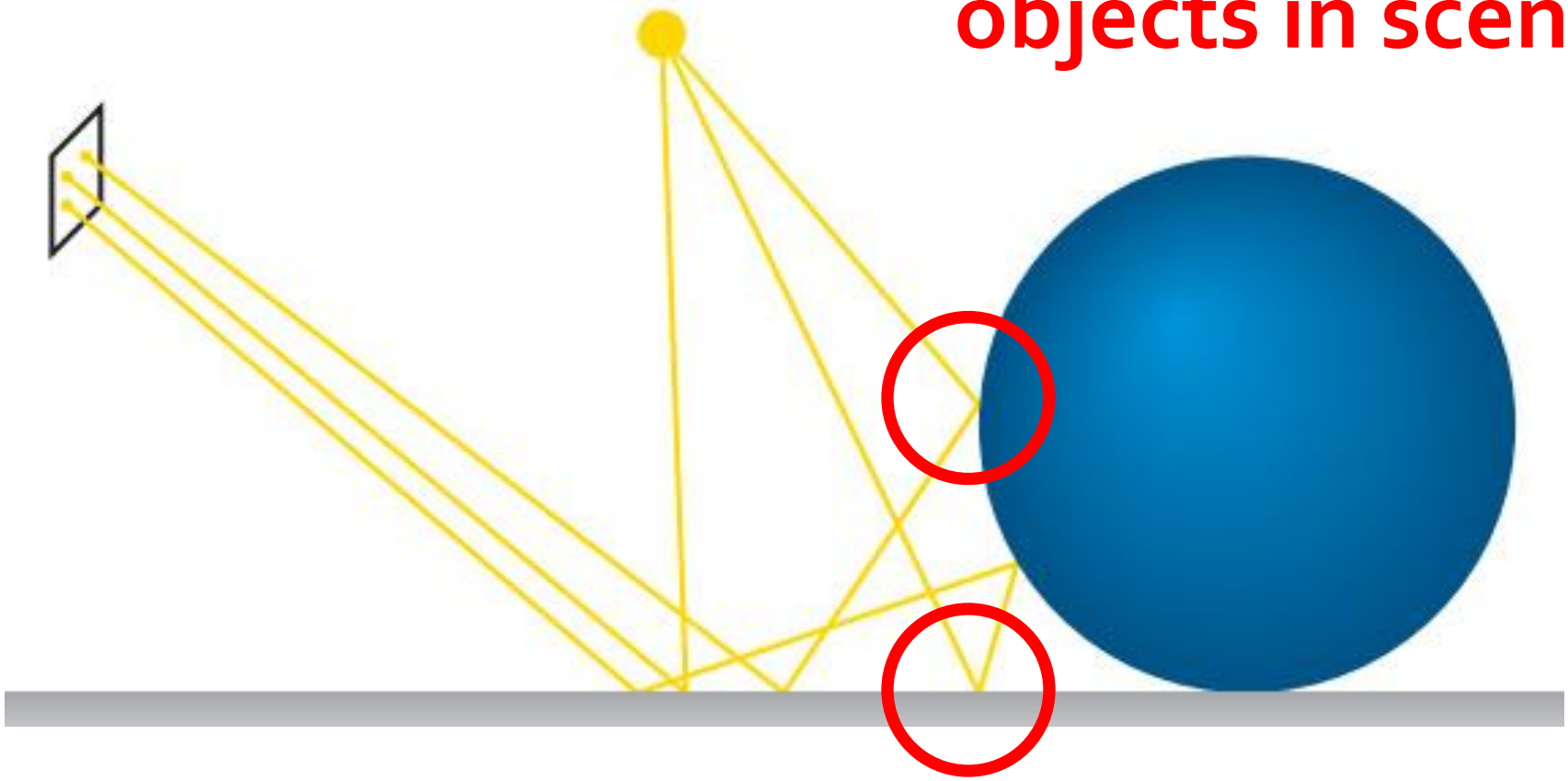
Paths of Light



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

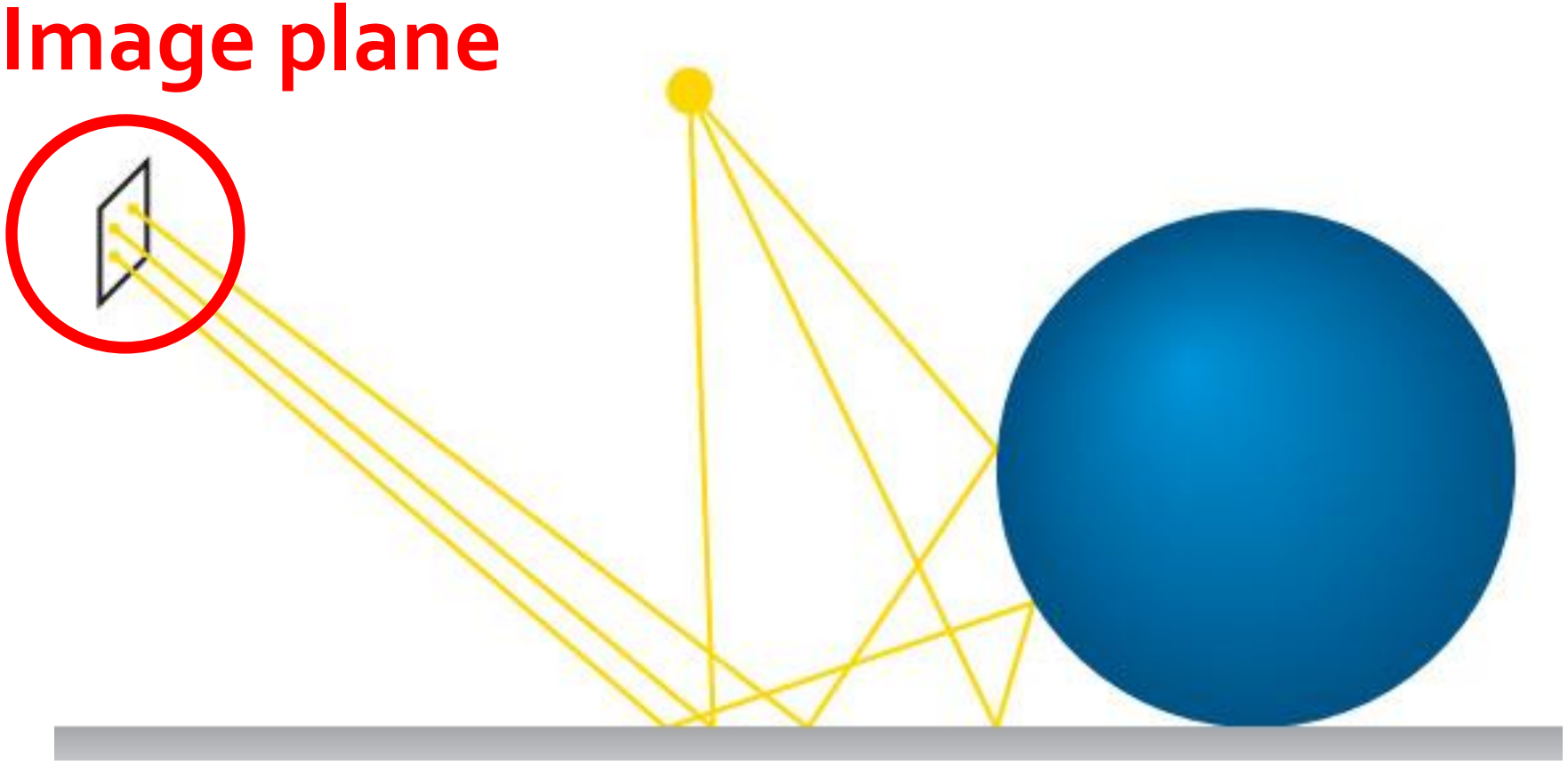
Paths of Light

Light interacts with objects in scene



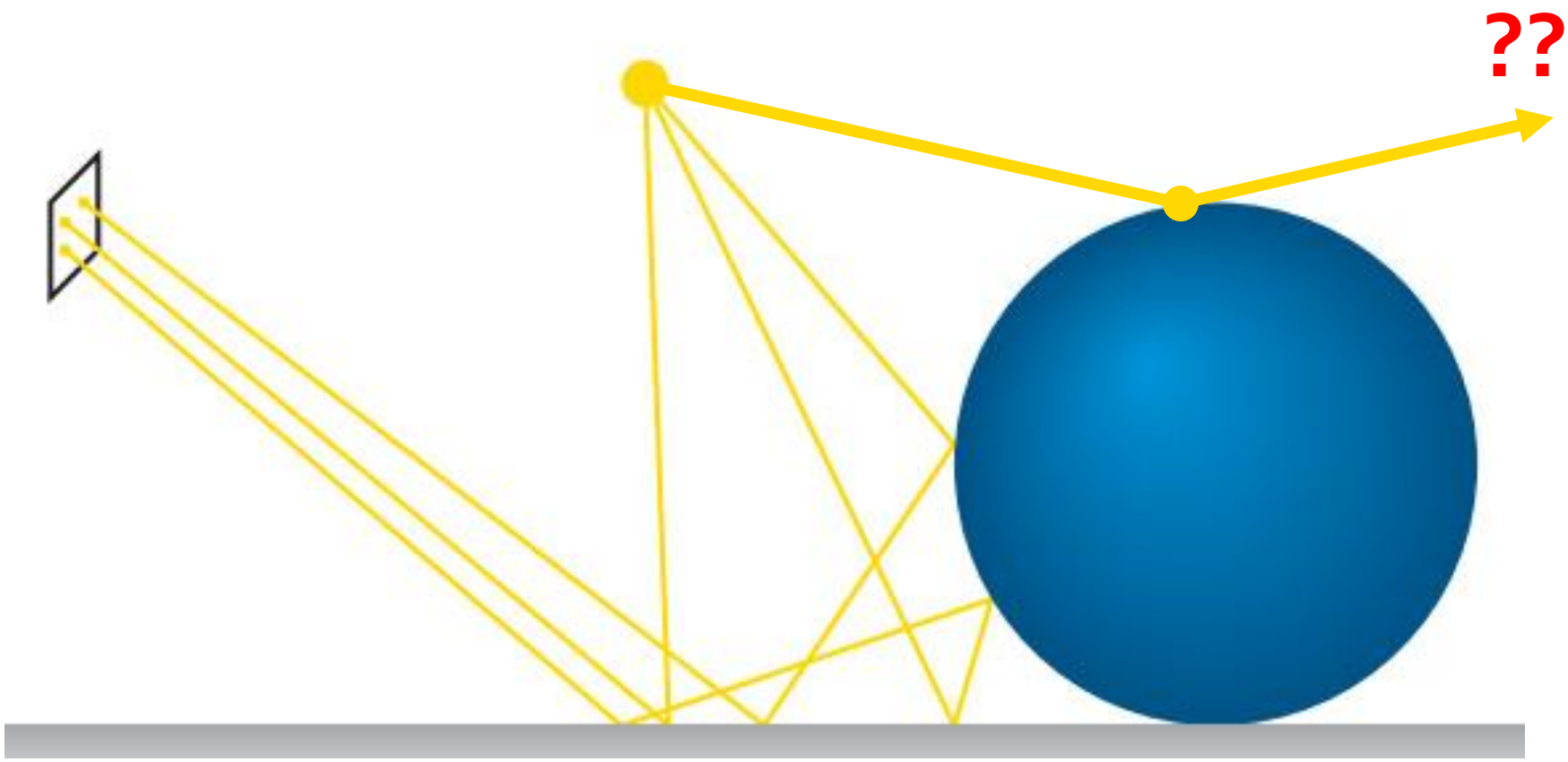
http://software.intel.com/file/37491

Paths of Light



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

Problem



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

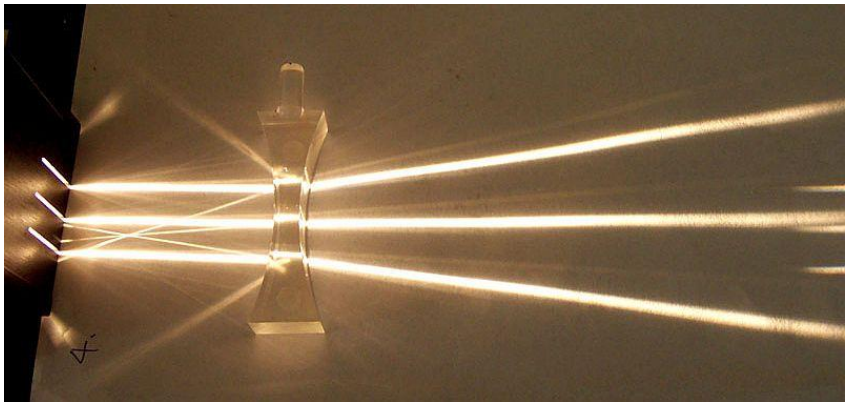
Most light doesn't reach the camera

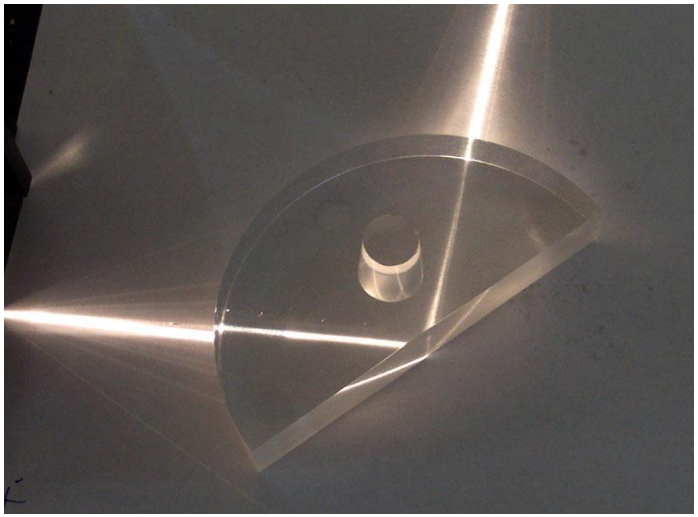
Problem

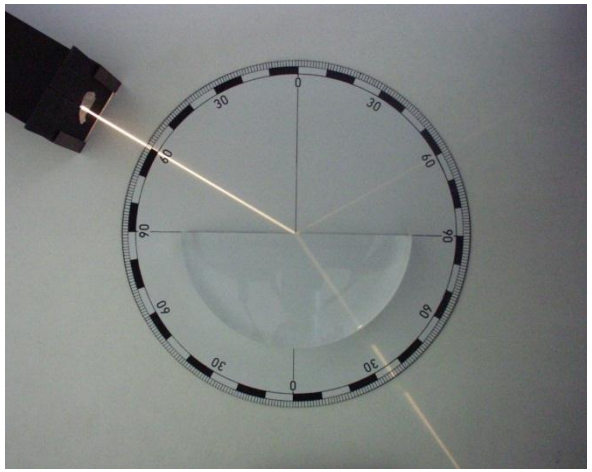
http://software.intel.com/file/37491

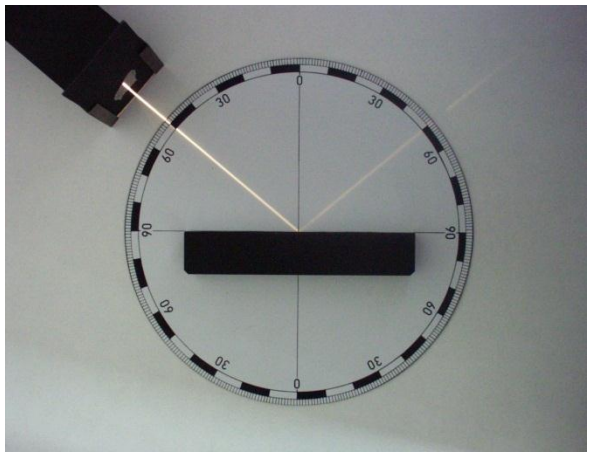
Most light doesn't reach the camera

Observation









Many lighting effects are reversible

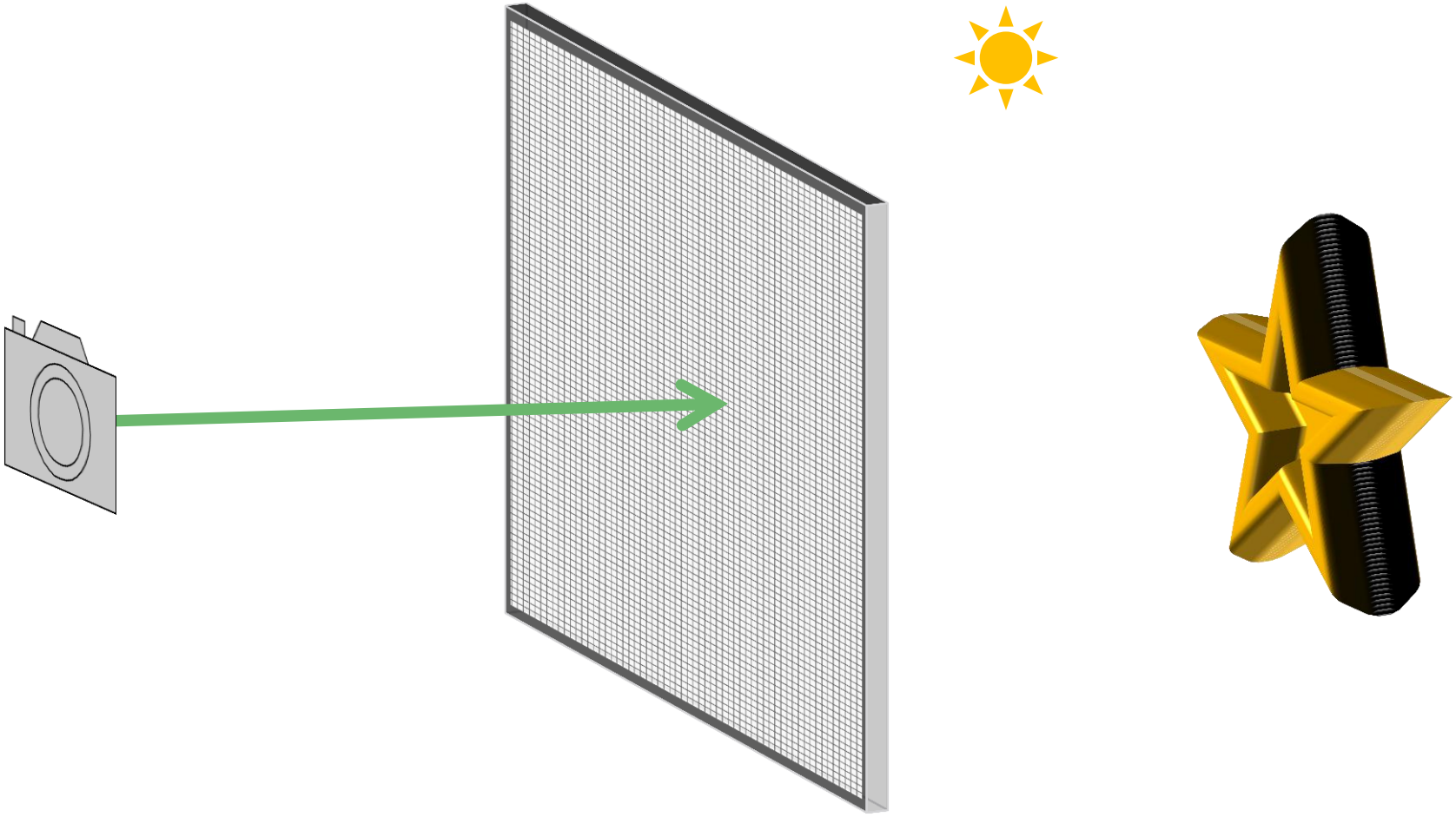
Forward Ray Tracing

Trace rays from light to camera

Backward Ray Tracing

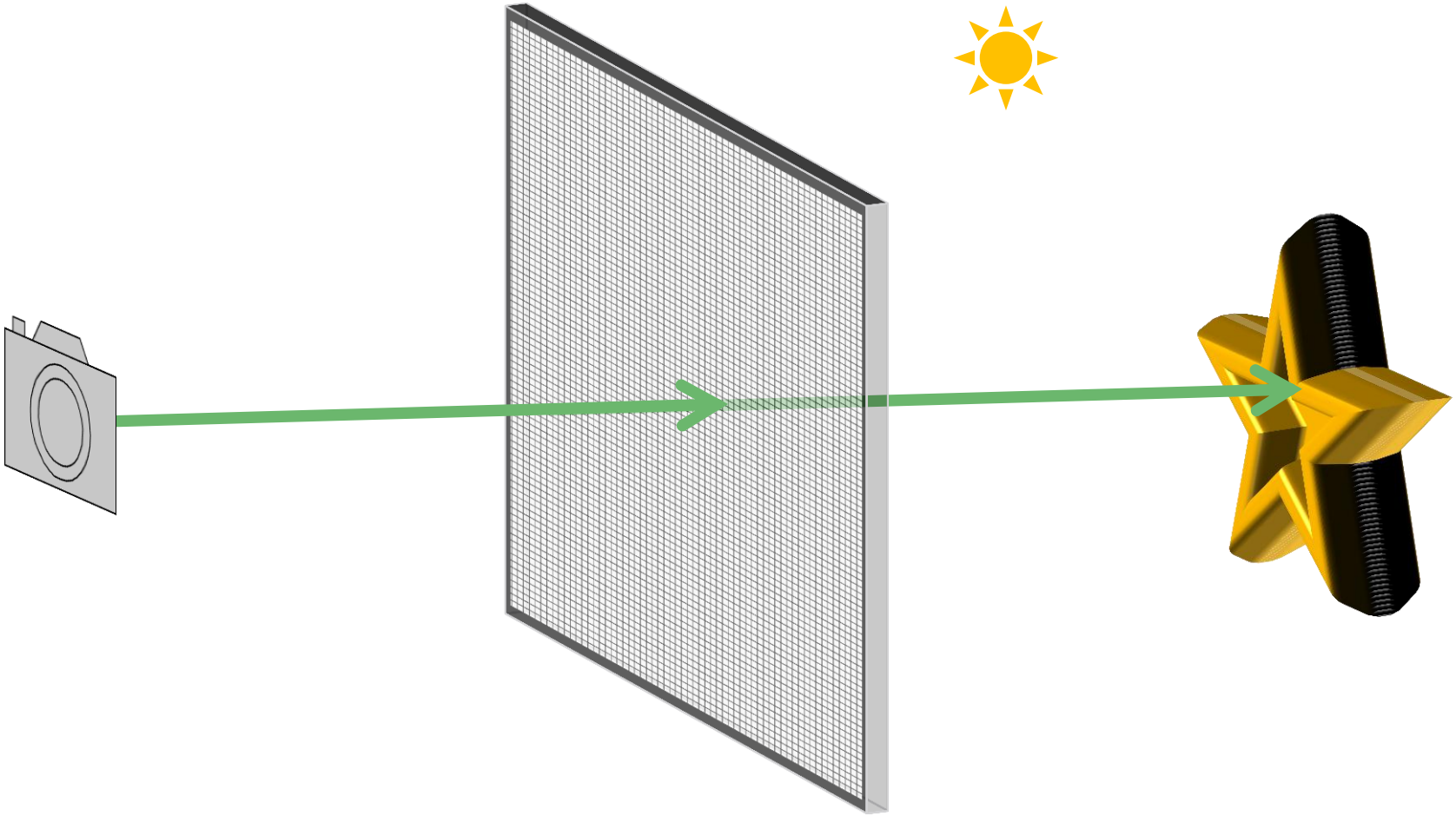
Trace rays from ~~light~~ *camera* to ~~camera~~ *light*

Ray Tracing Algorithm



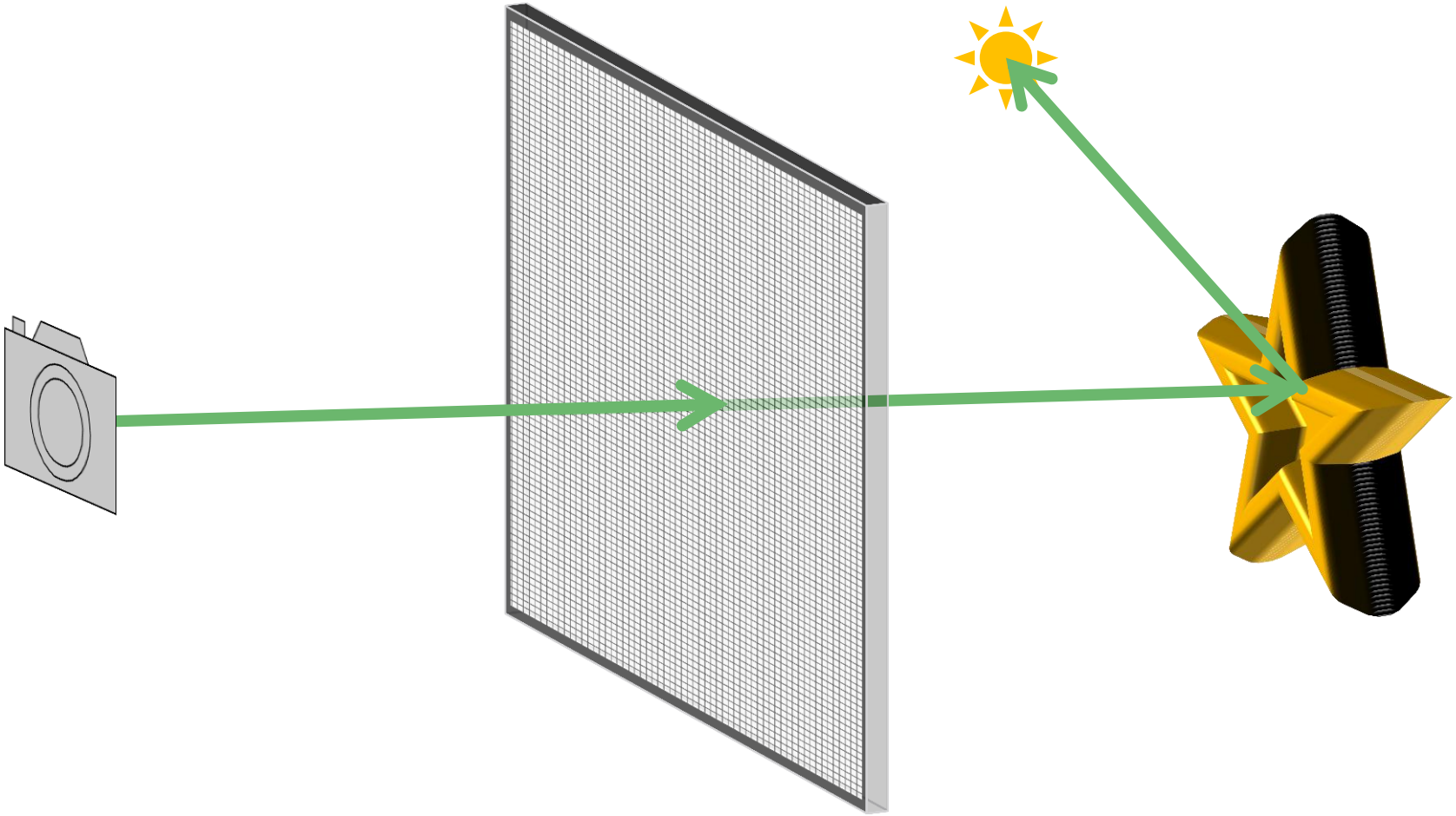
Draw viewing ray through each pixel

Ray Tracing Algorithm



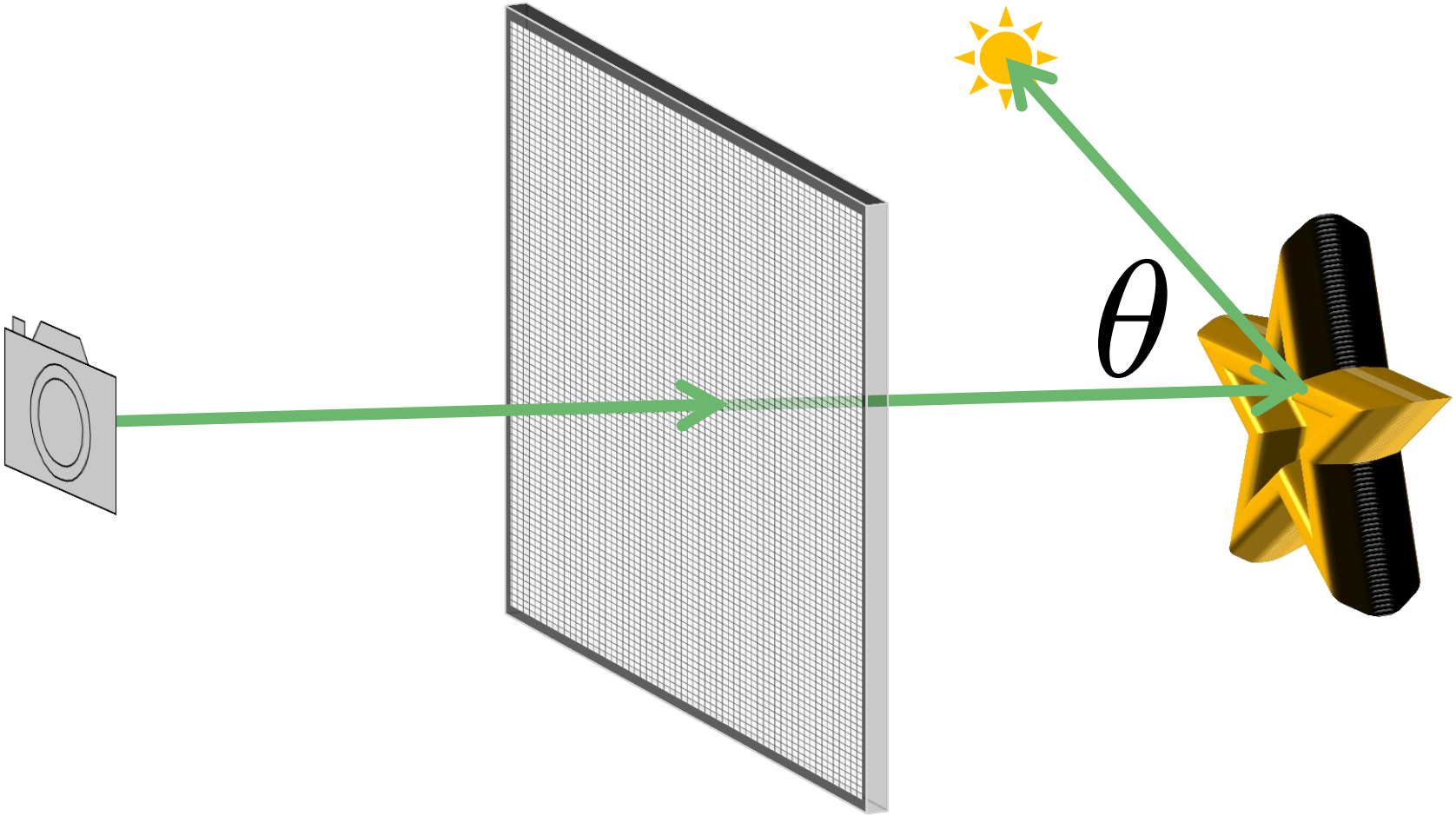
Intersect with geometry

Ray Tracing Algorithm



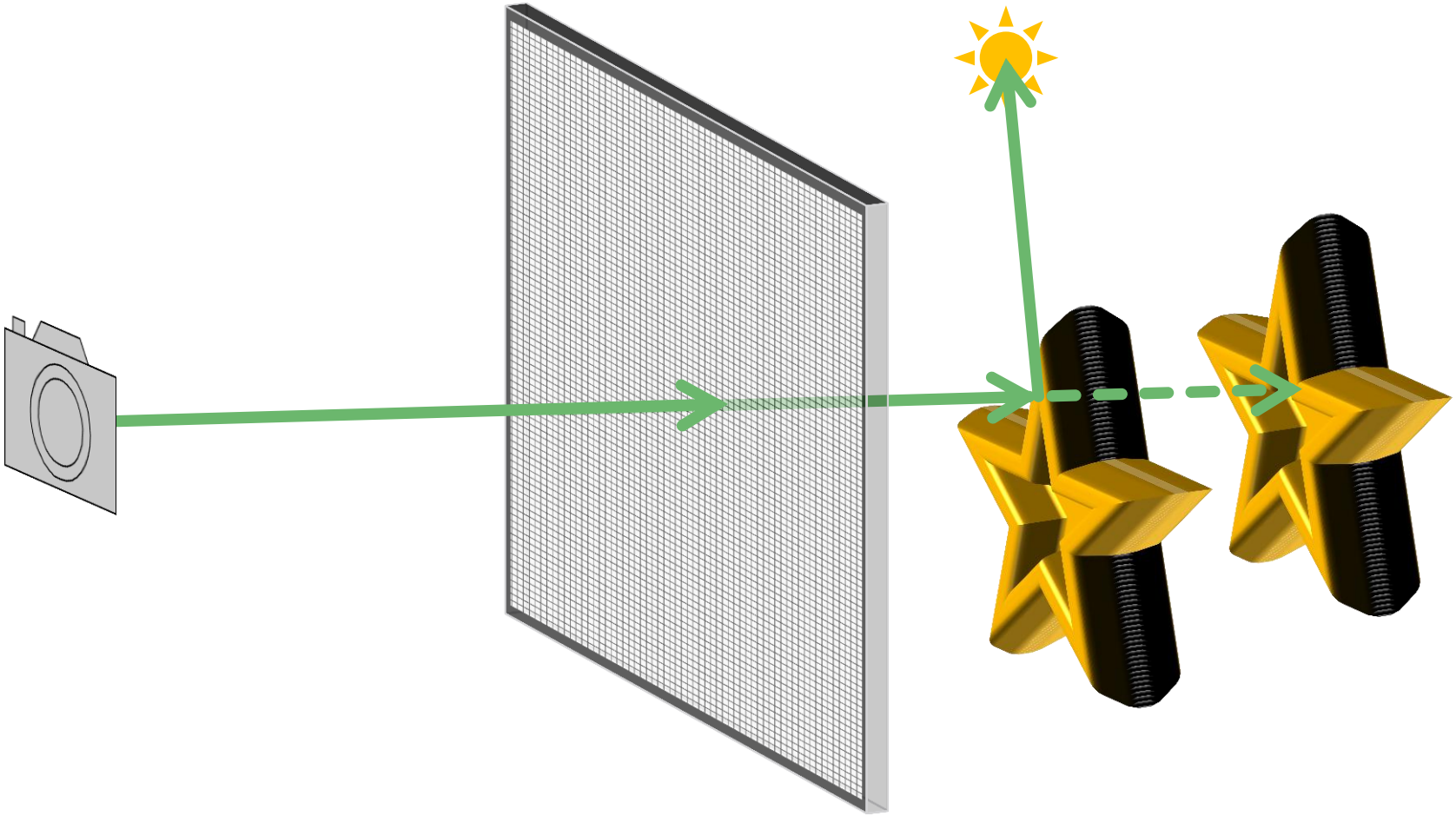
Compute light ray

Ray Tracing Algorithm



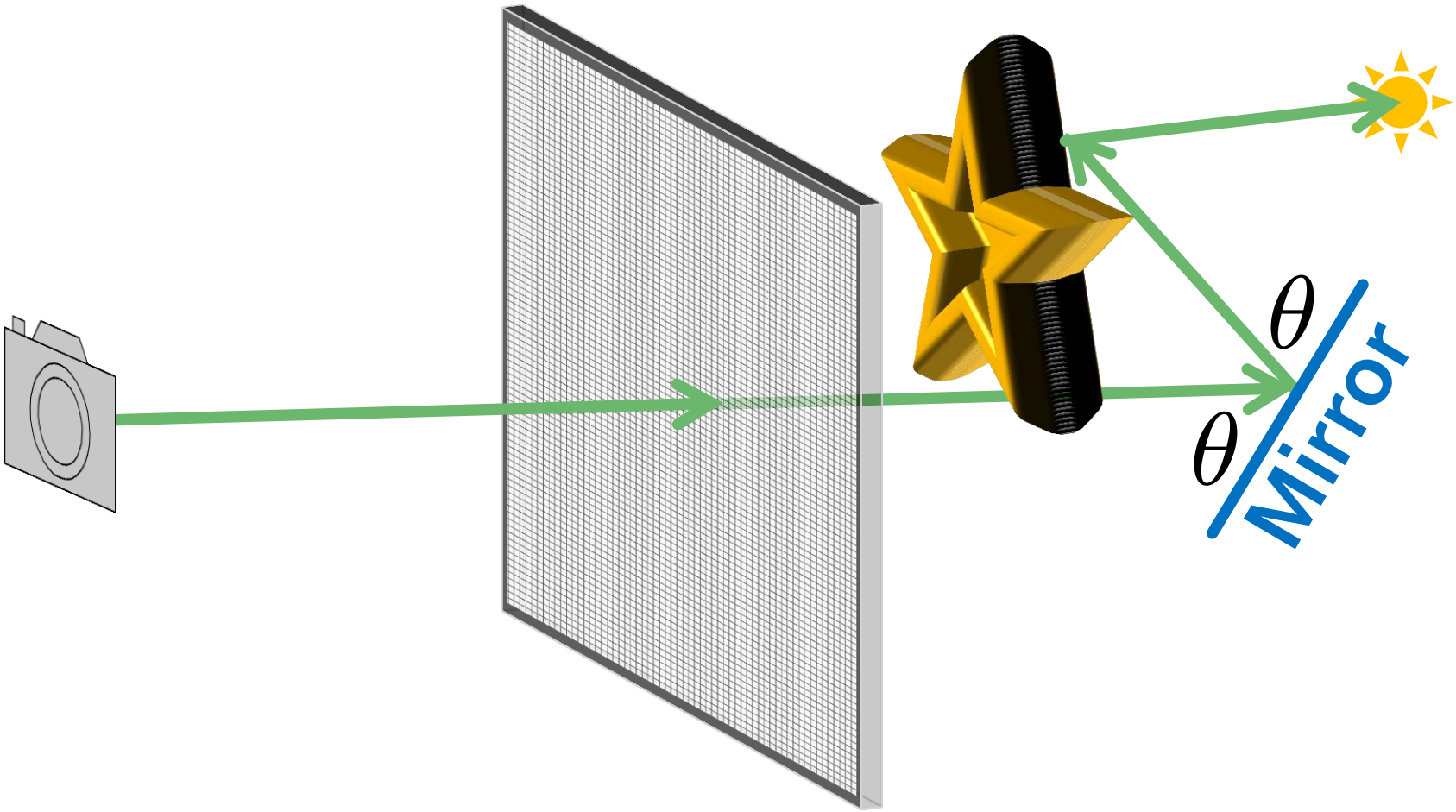
Evaluate shading

Dealing with Multiple Objects



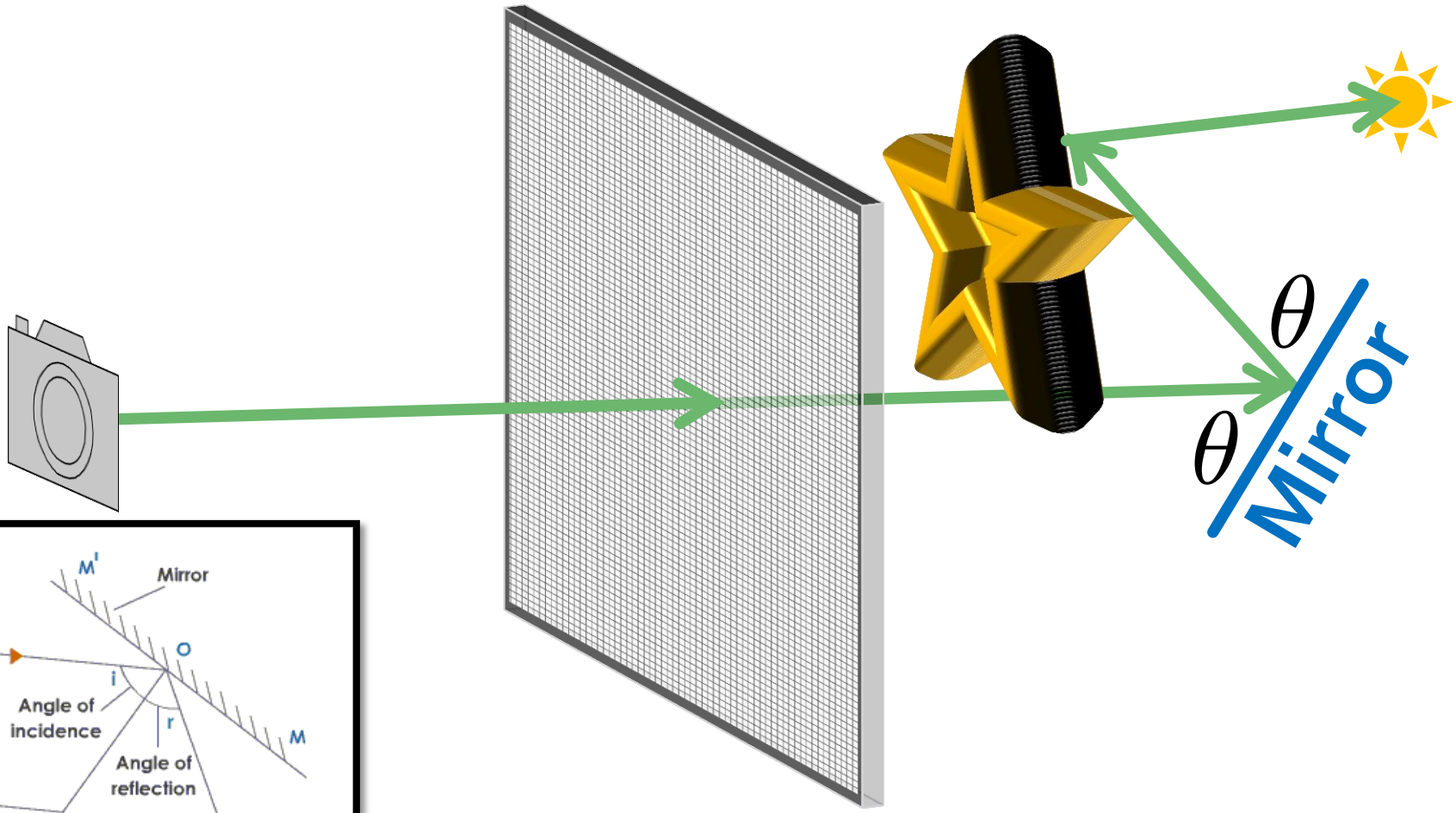
Take closest one

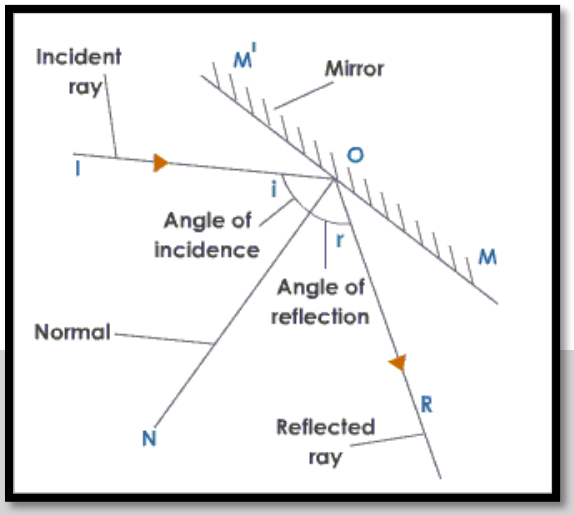
Dealing with Reflection/Refraction



Redirect ray

Dealing with Reflection/Refraction





Redirect ray

Outer Loop

Rasterization

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

Outer Loop

Rasterization

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

Ray Tracing

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

Outer Loop

Rasterization

Object-order rendering

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

Ray Tracing

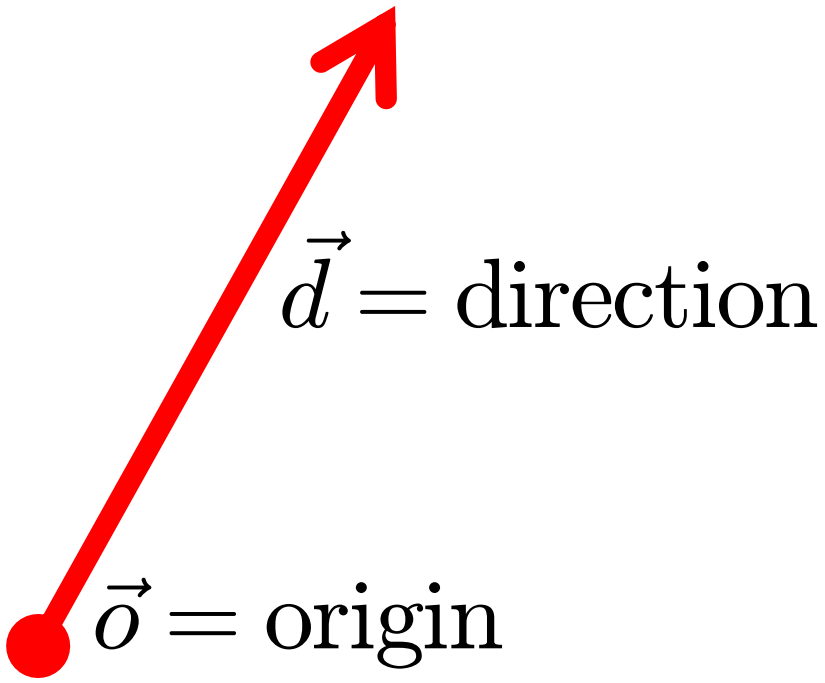
Image-order rendering

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

Simple Pseudocode

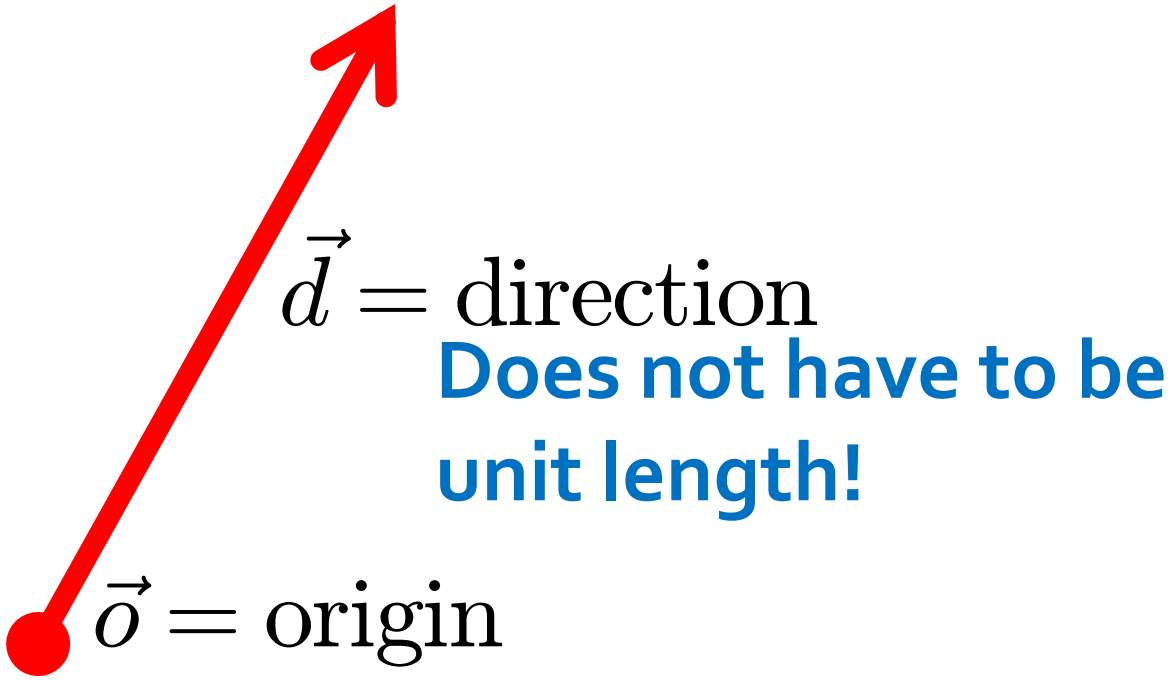
```
for each pixel do
    compute viewing ray
    if (ray hits an object with t>0) then
        Compute normal
        Evaluate shading model
    else
        Set pixel color to background color
```

Math of Ray Tracing



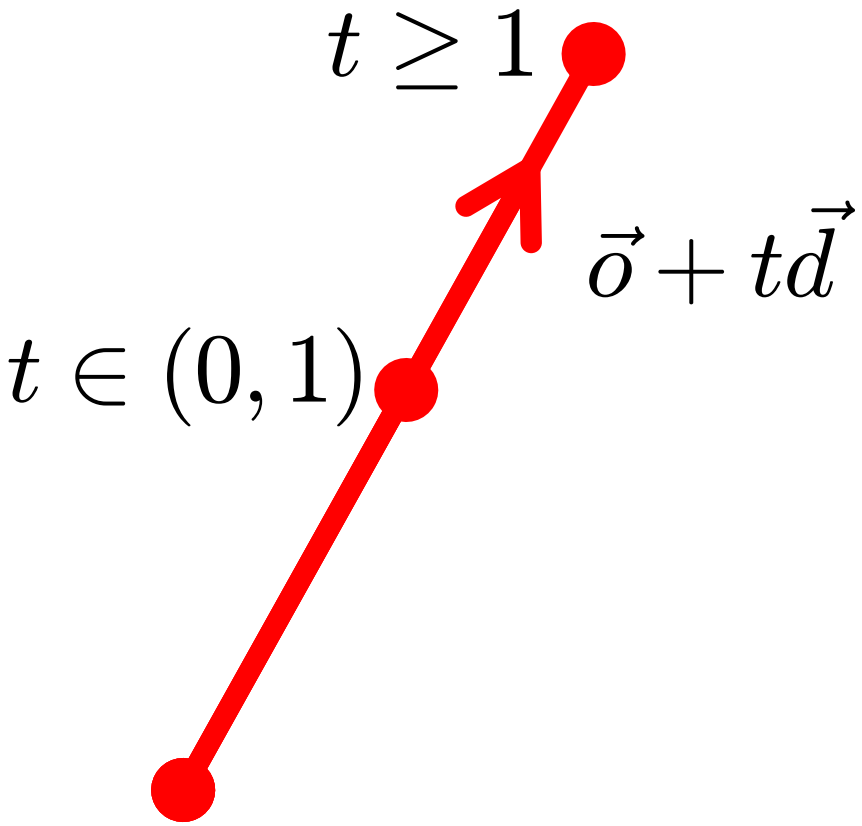
Ray

Math of Ray Tracing



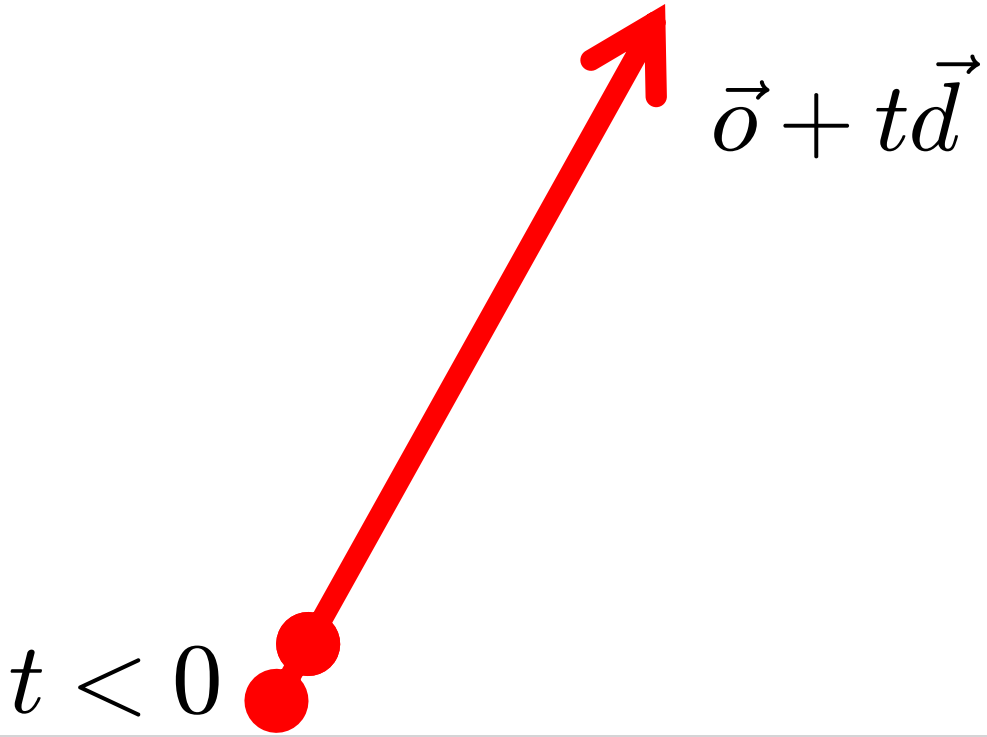
Ray

Math of Ray Tracing



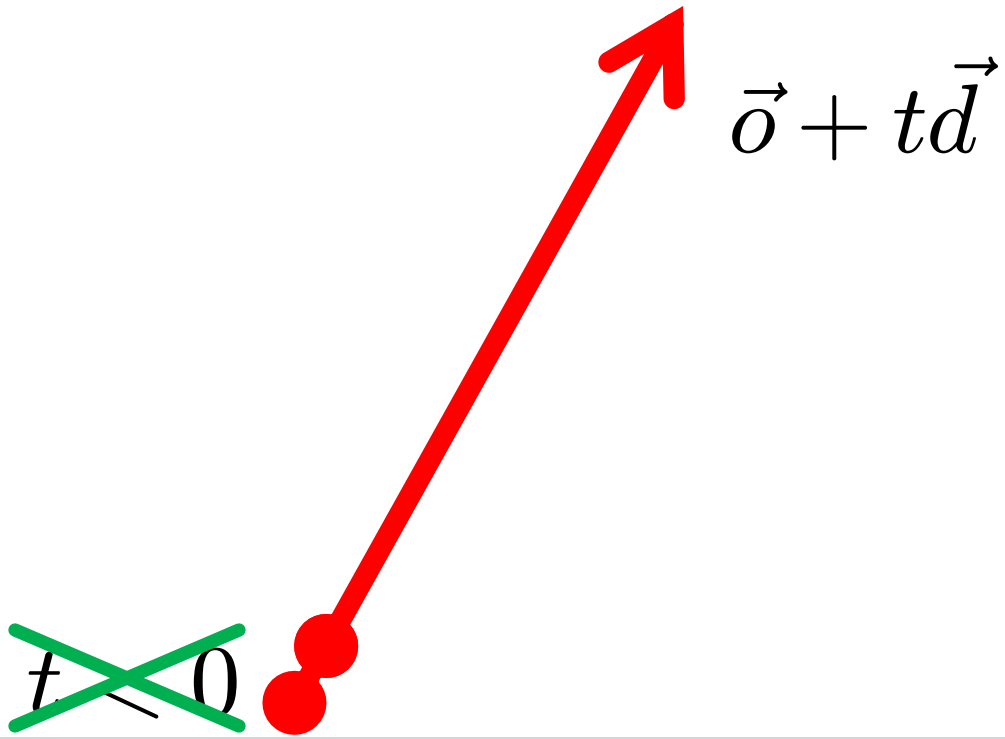
Point on Ray

Math of Ray Tracing



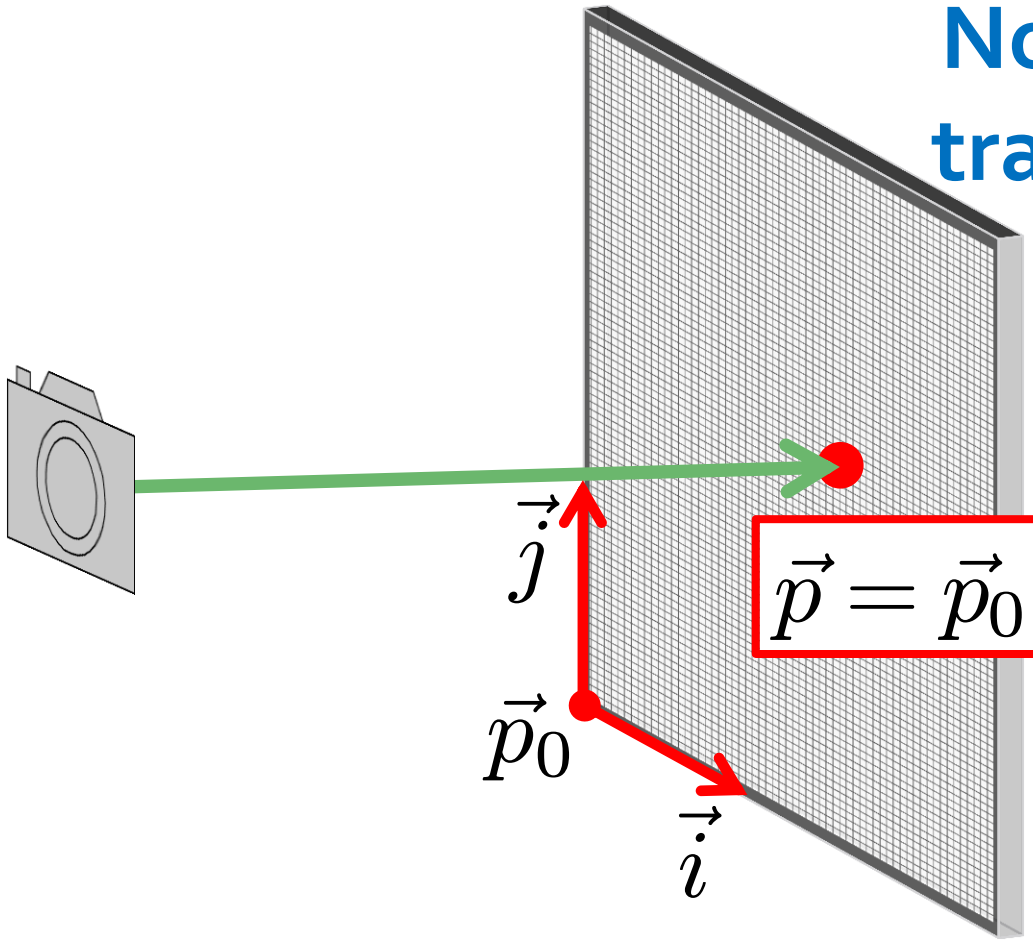
Point on Ray

Math of Ray Tracing



Point on Ray

Math of Ray Tracing



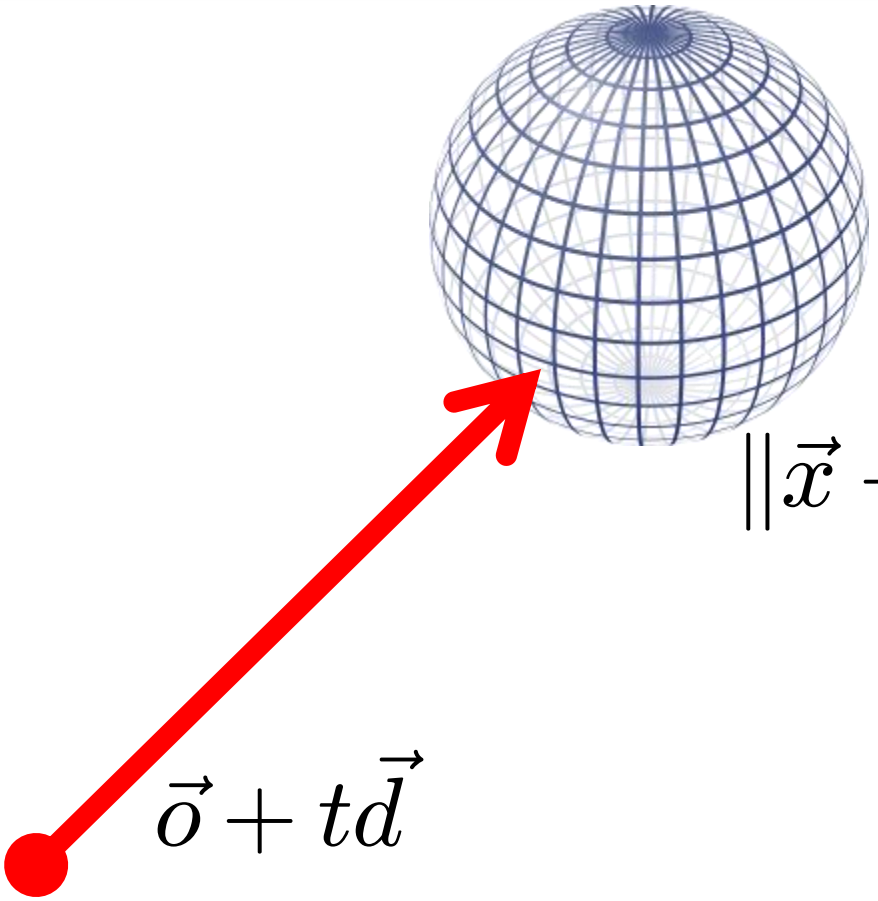
No perspective transformation needed!

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

$$u \in [0, w], v \in [0, h]$$

Generating rays

Math of Ray Tracing



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

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

Ray-object intersection

Math of Ray Tracing

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

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

Ray-object intersection

Math of Ray Tracing

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

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

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

Ray-object intersection

Math of Ray Tracing

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

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

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

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

Ray-object intersection

Math of Ray Tracing

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

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

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

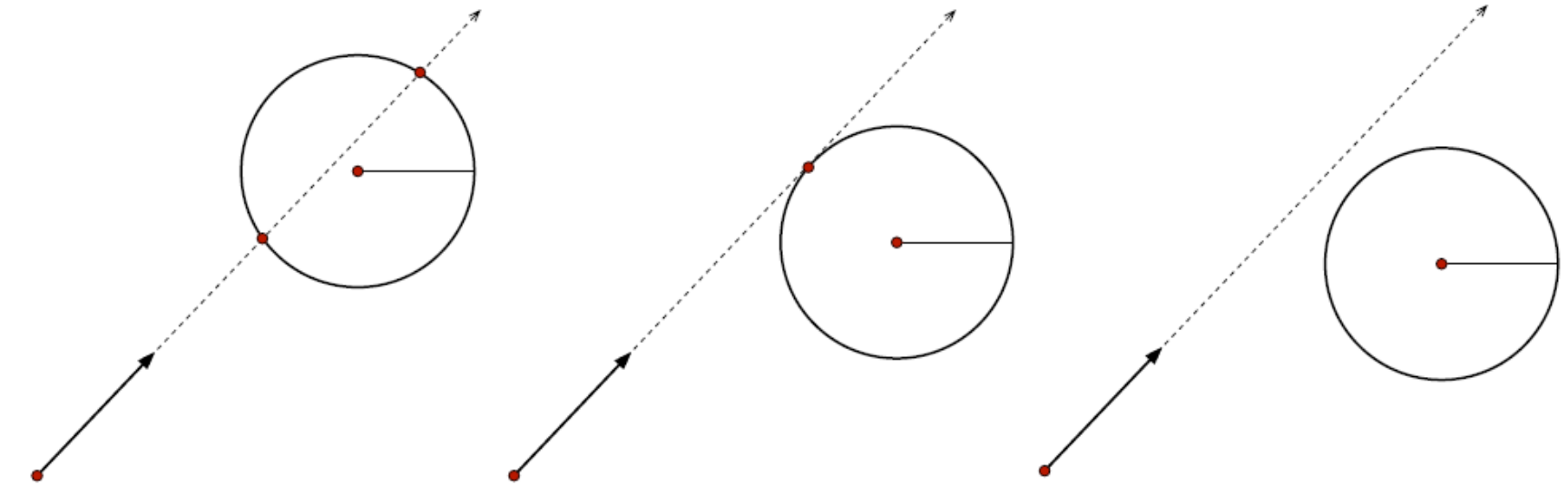
$$(\vec{o} + t\vec{d} - \vec{c}) \cdot (\vec{o} + t\vec{d} - \vec{c}) = R^2$$

Quadratic in t

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

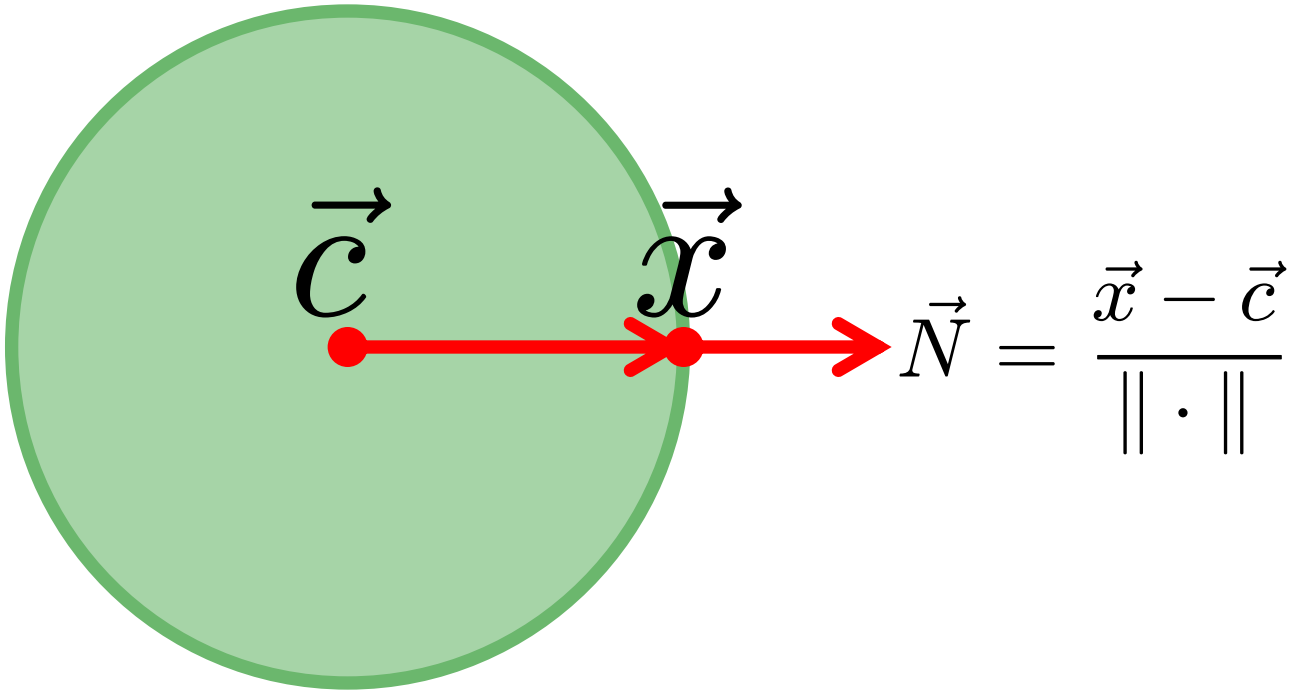
Ray-object intersection

Math of Ray Tracing



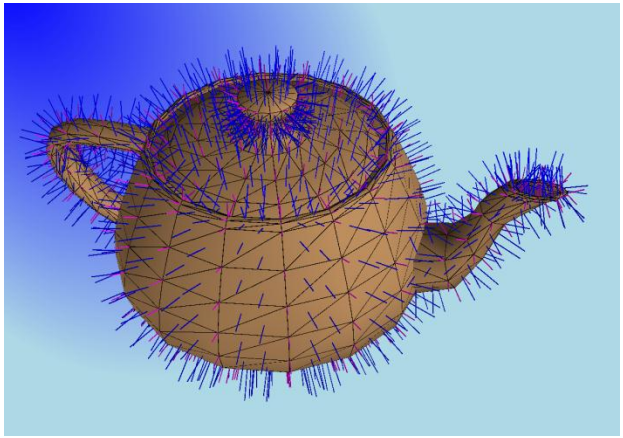
Ray-object intersection

Math of Ray Tracing

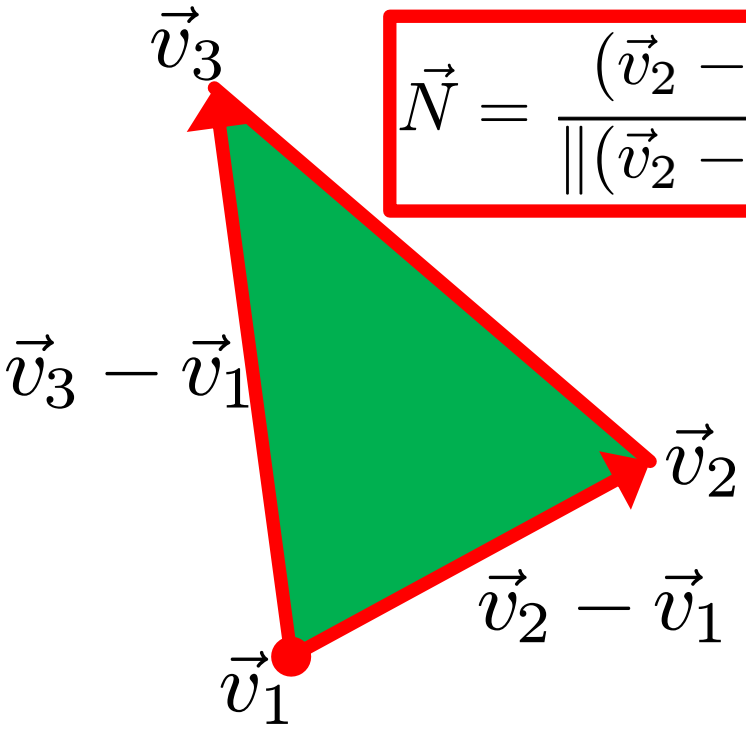


Normals

Math of Ray Tracing



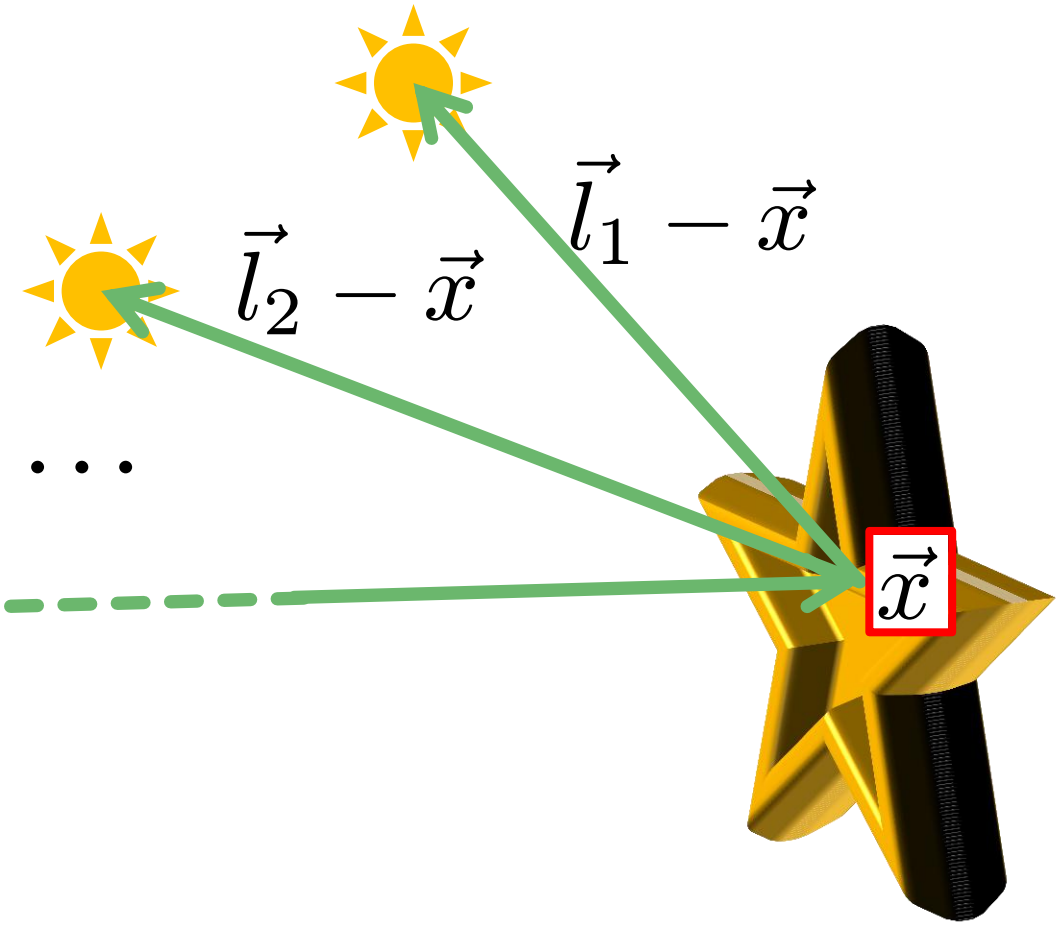
http://designjk.files.wordpress.com/2011/12/teapot-decorated.png



$$\vec{N} = \frac{(\vec{v}_2 - \vec{v}_1) \times (\vec{v}_3 - \vec{v}_1)}{\|(\vec{v}_2 - \vec{v}_1) \times (\vec{v}_3 - \vec{v}_1)\|}$$

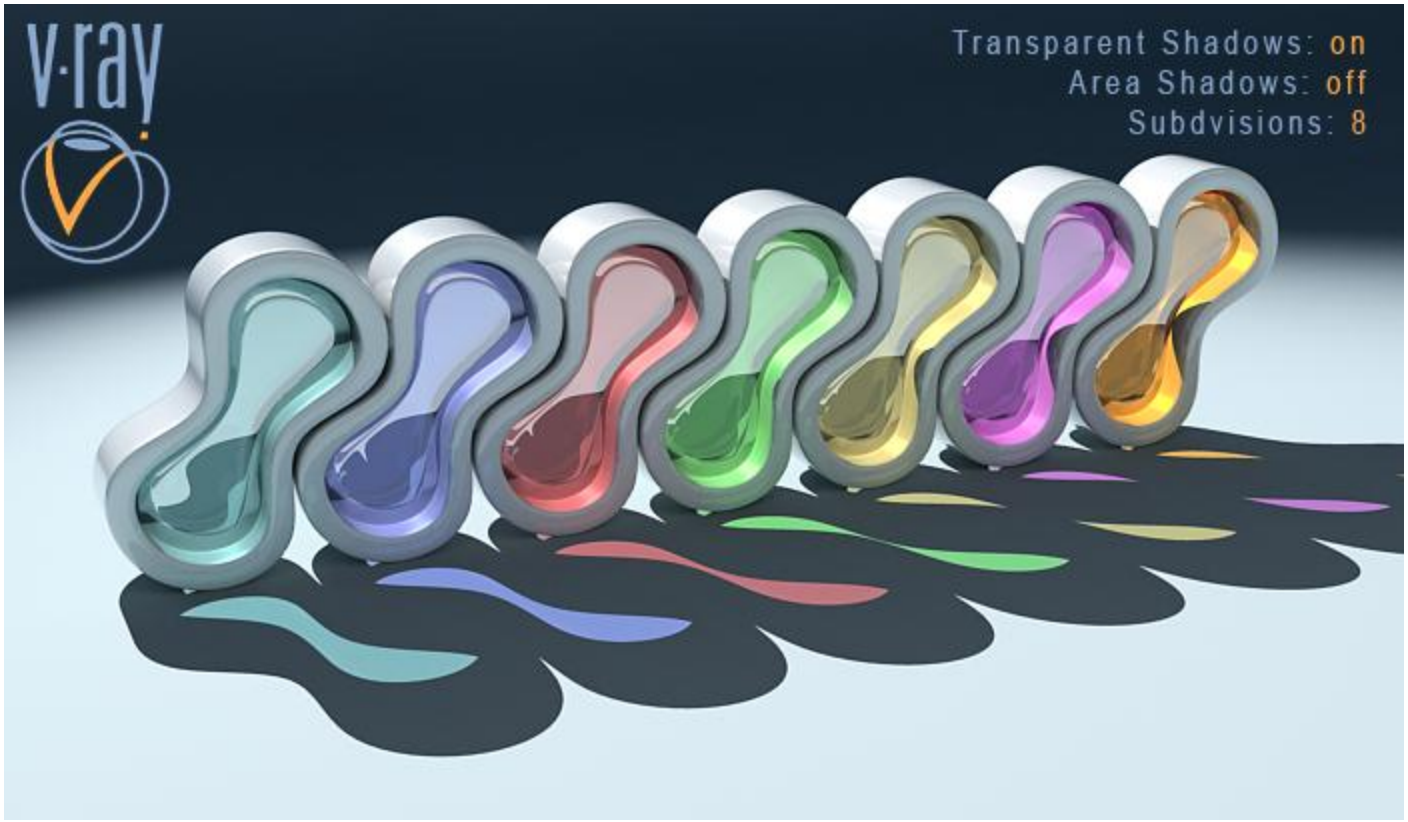
Normals

Math of Ray Tracing



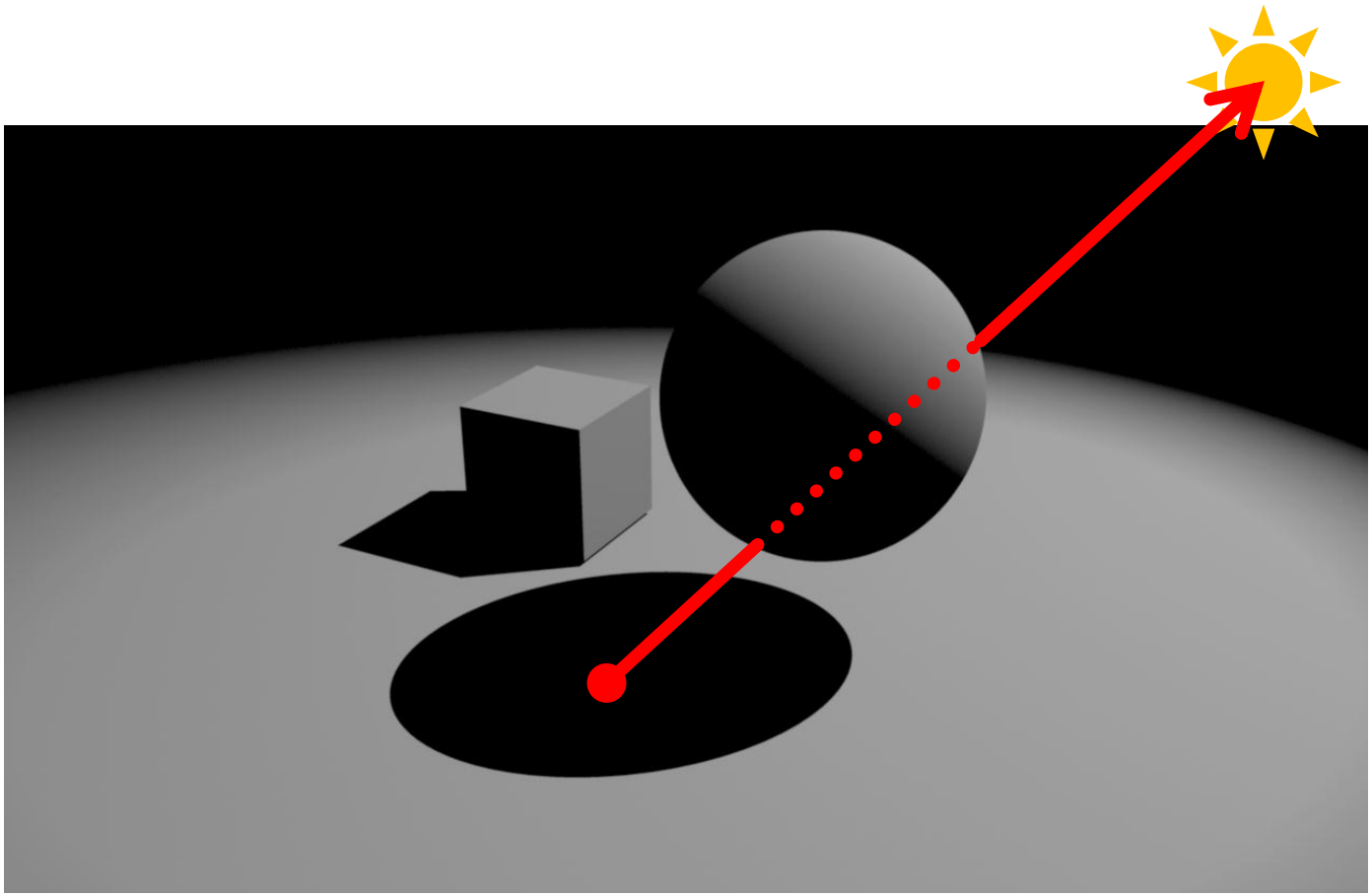
Light rays

Shadows



http://www.spot3d.com/vray/help/150SP1/images/examples/shadows/03_raytraced_colored.png

Shadows



http://2.bp.blogspot.com/-2CpxoSdtNE8/T5nMCIgobpI/AAAAAAAAAJ4/XgaZ168HpKg/s1600/P0008_1004.png

Intersect light ray with scene

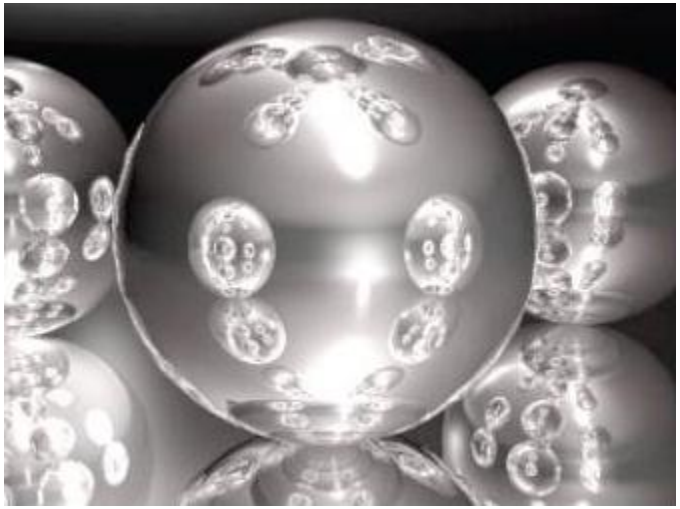
Object-Oriented Ray Tracing

```
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

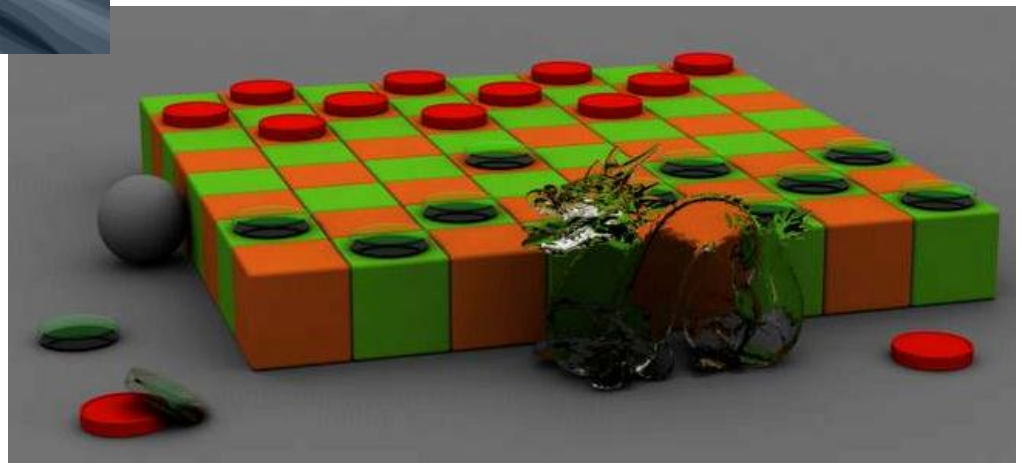


Lots of bounces

Fewer bounces

Save time; avoid infinite recursion

Instancing



http://www.deepakantony.com/wp-content/uploads/2012/04/ambientocclusion_instancing.png
http://members.optusnet.com.au/pennywyatt/Images/People.png

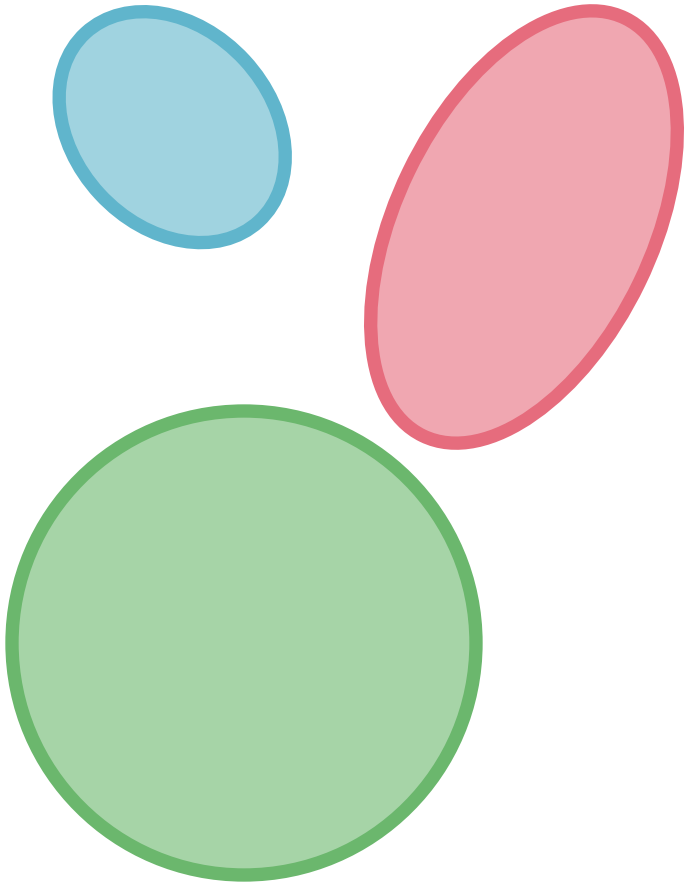
Deal with repeated objects

Instancing

http://members.optusnet.com.au/pennywyatt/Images/People.png

Deal with repeated objects

Instancing

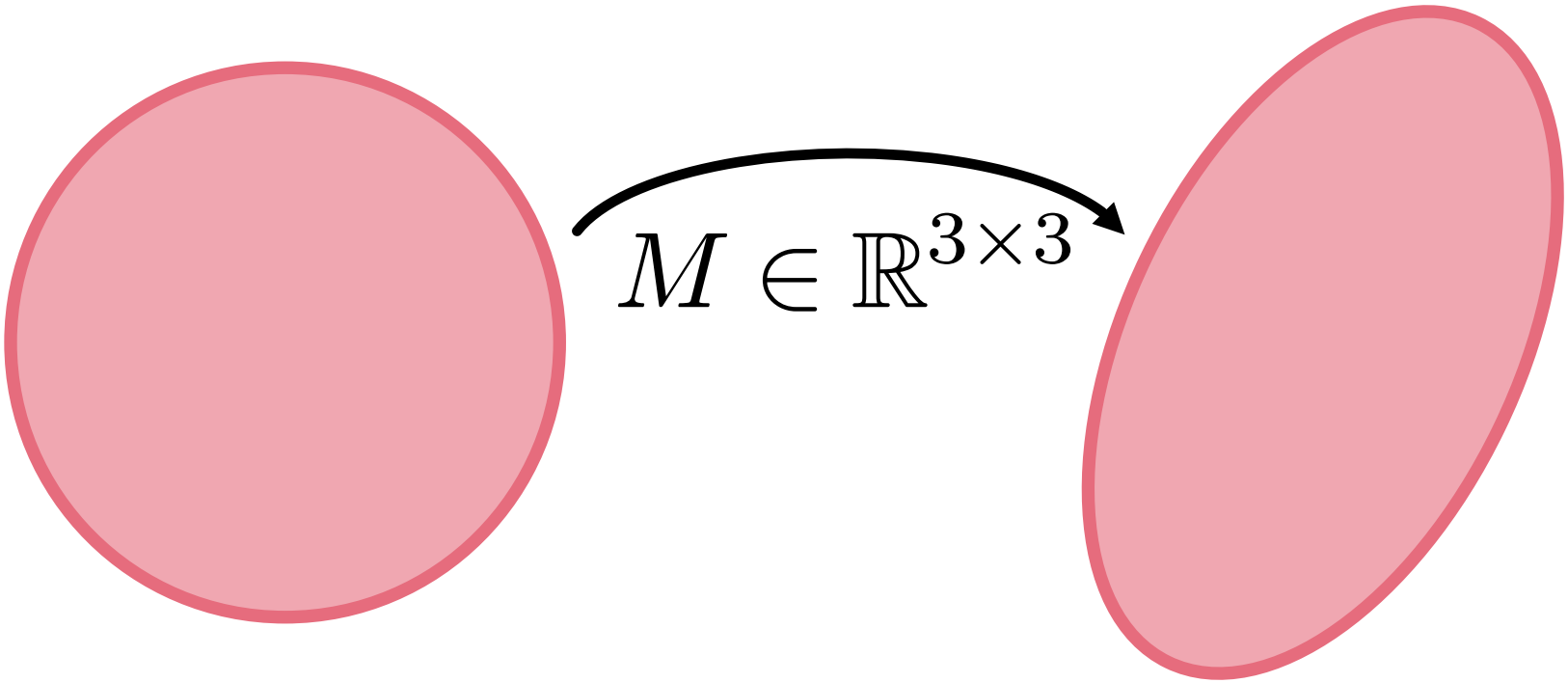


Each instance stores a transformation matrix M

(can stretch, rotate, etc!)

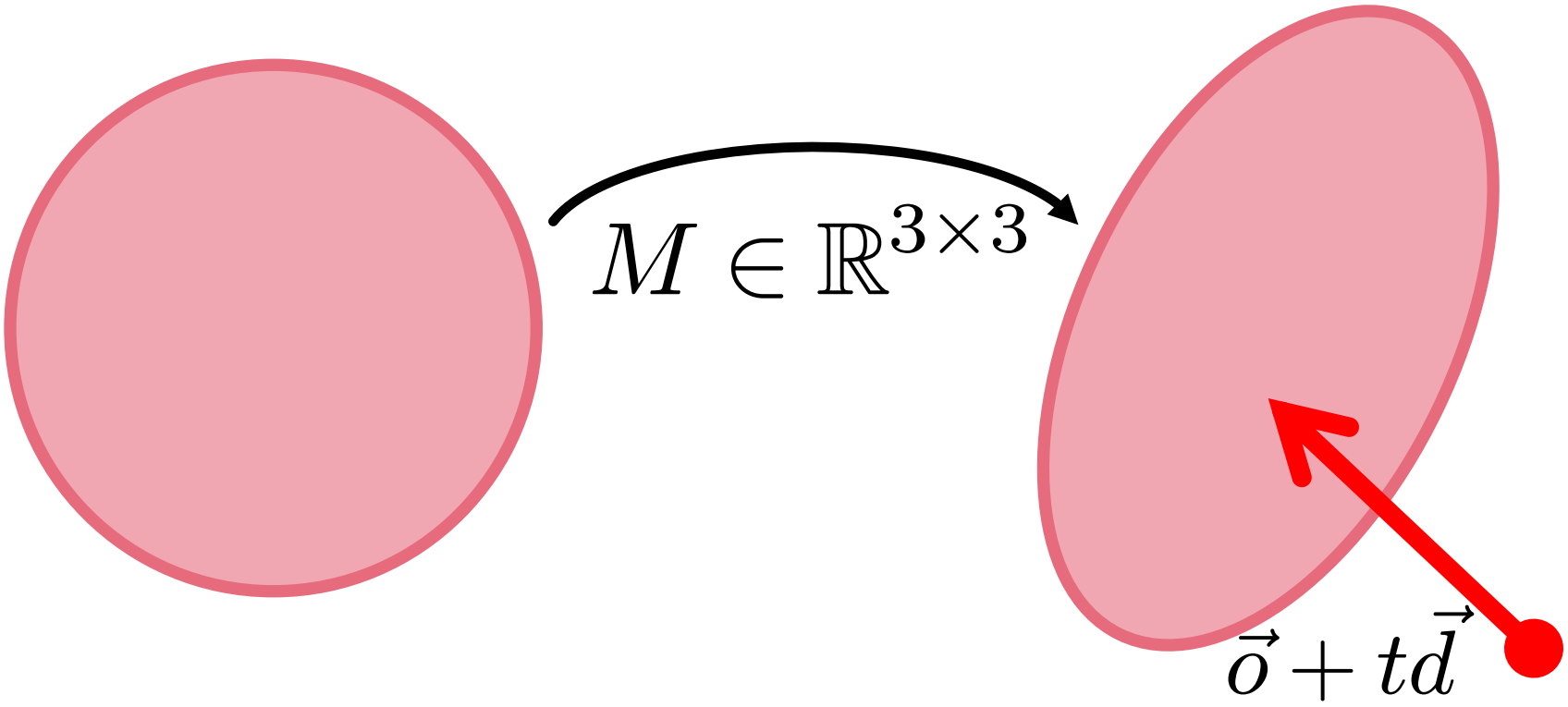
Deal with repeated objects

Instancing



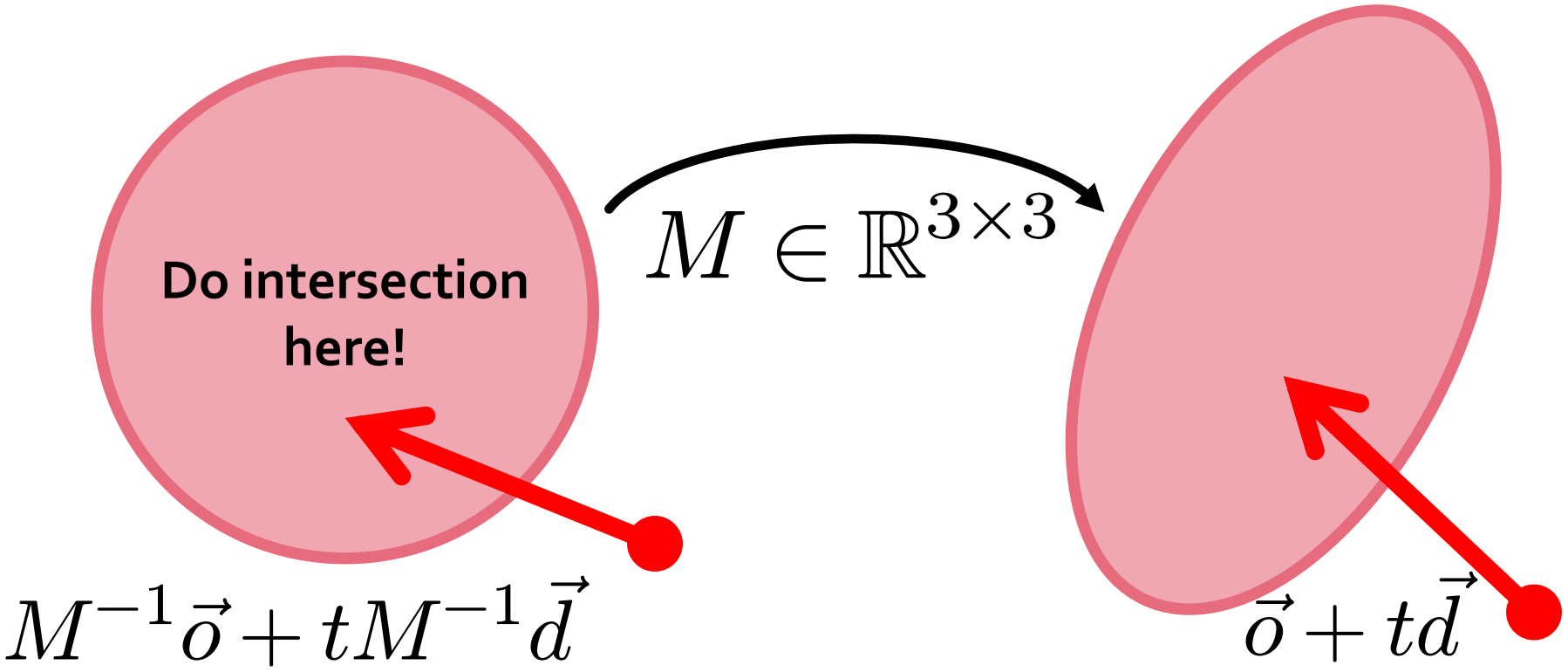
Deal with repeated objects

Instancing



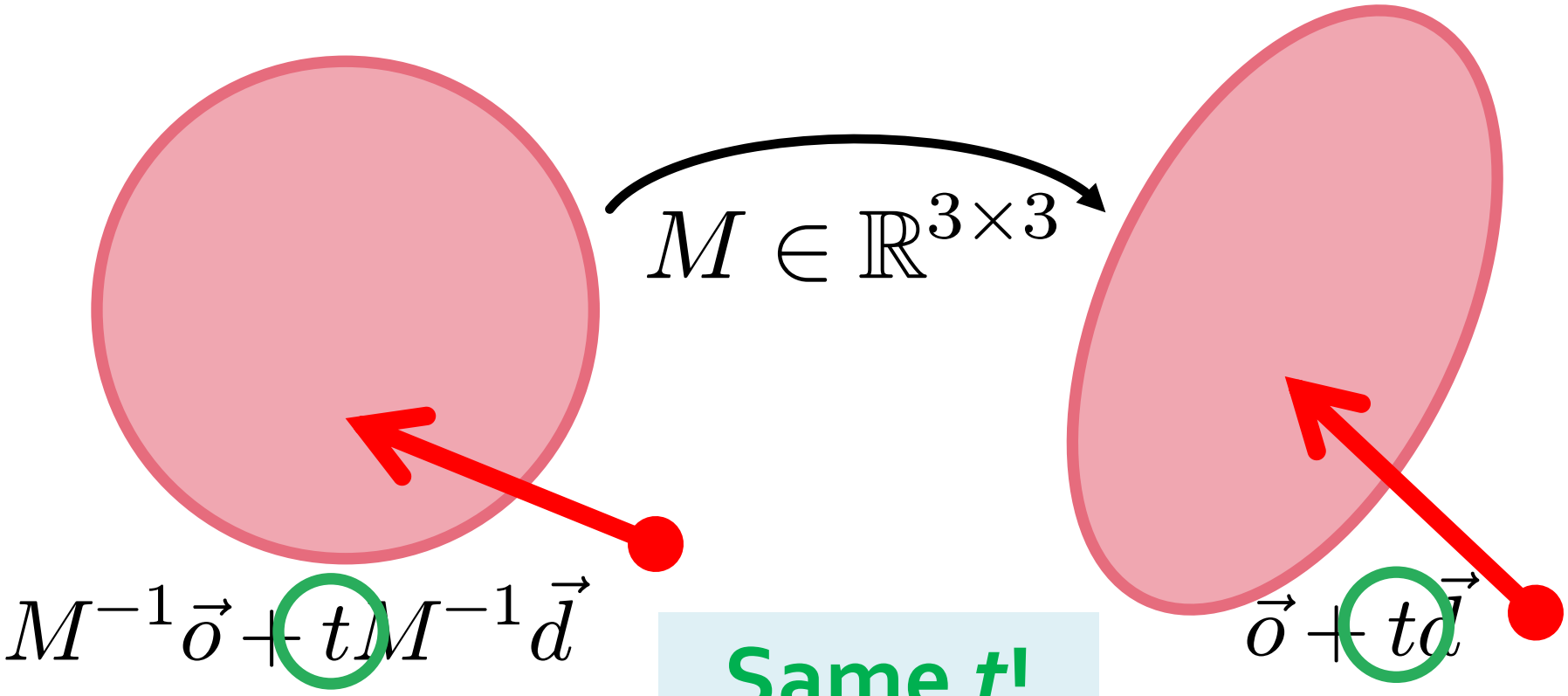
Deal with repeated objects

Instancing



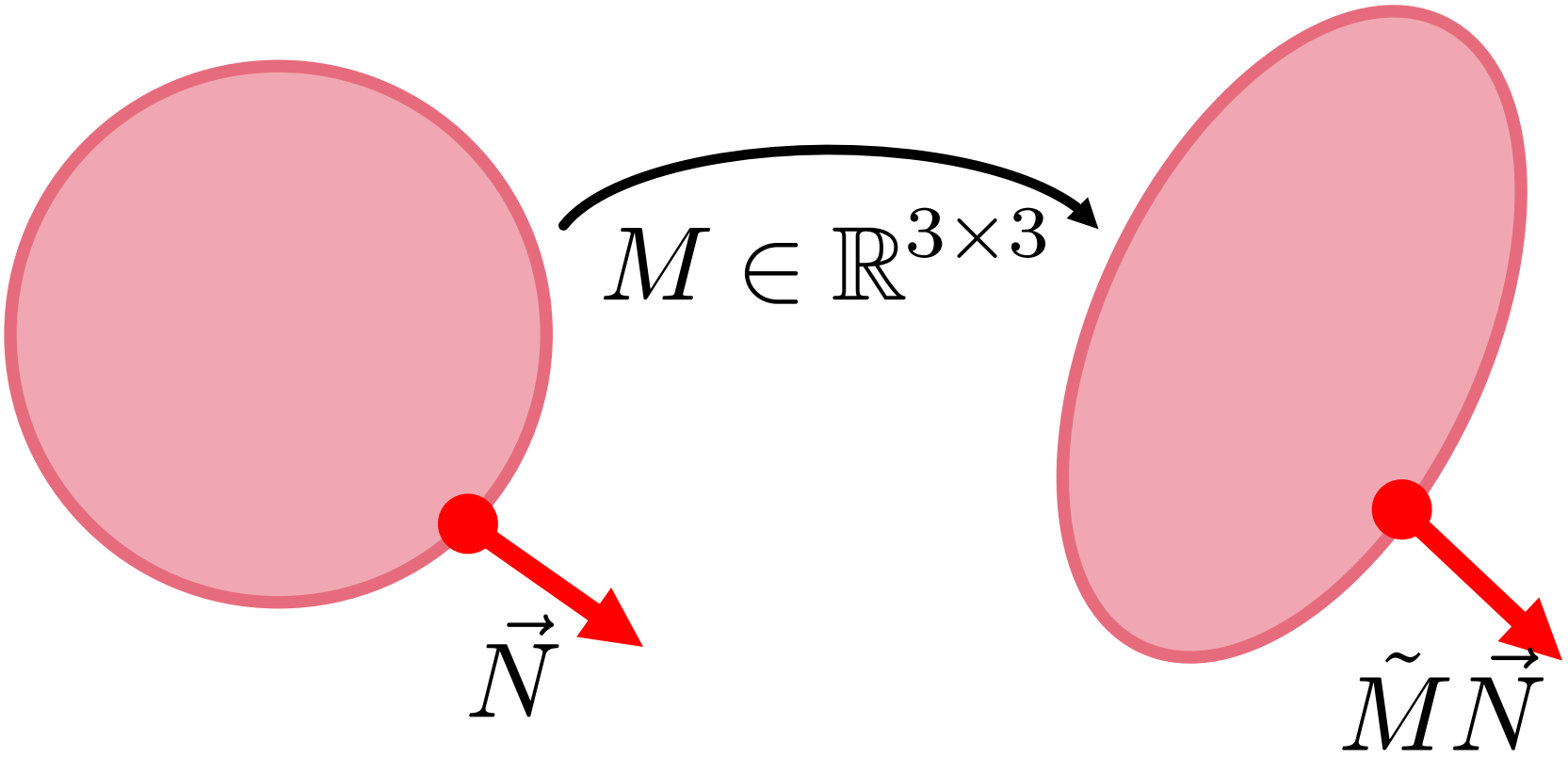
Deal with repeated objects

Instancing



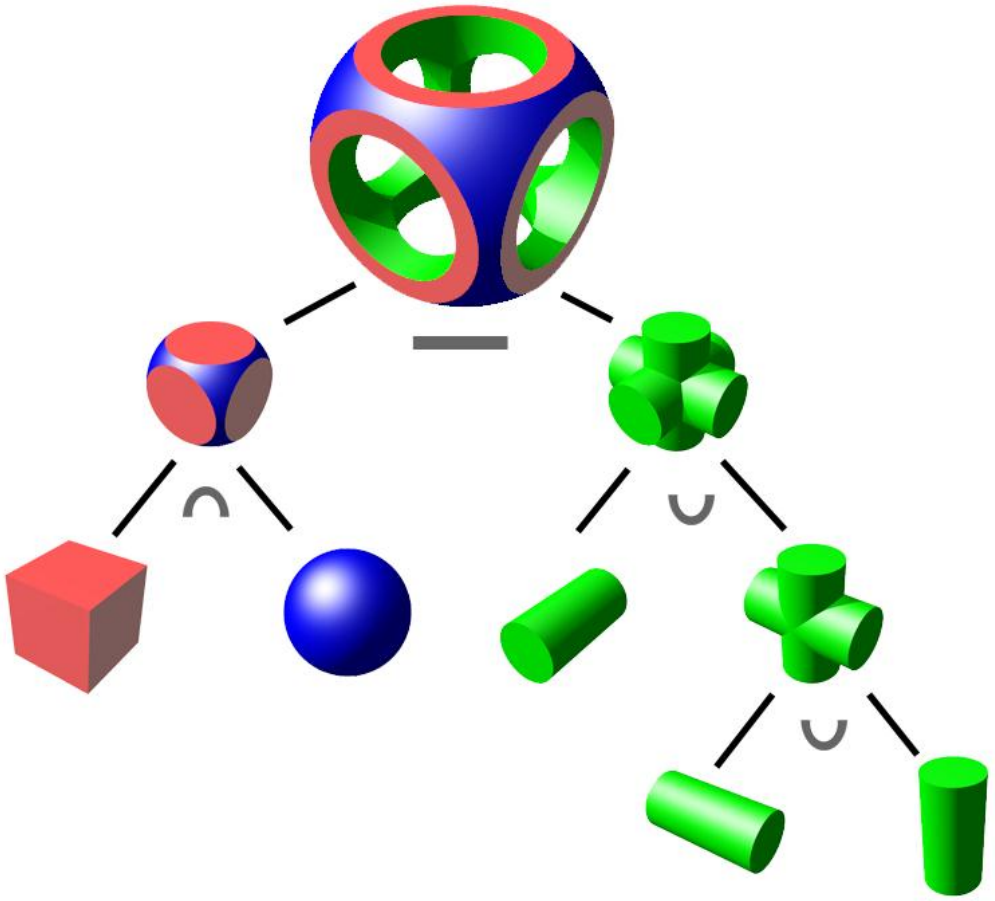
Deal with repeated objects

Instancing



Deal with repeated objects

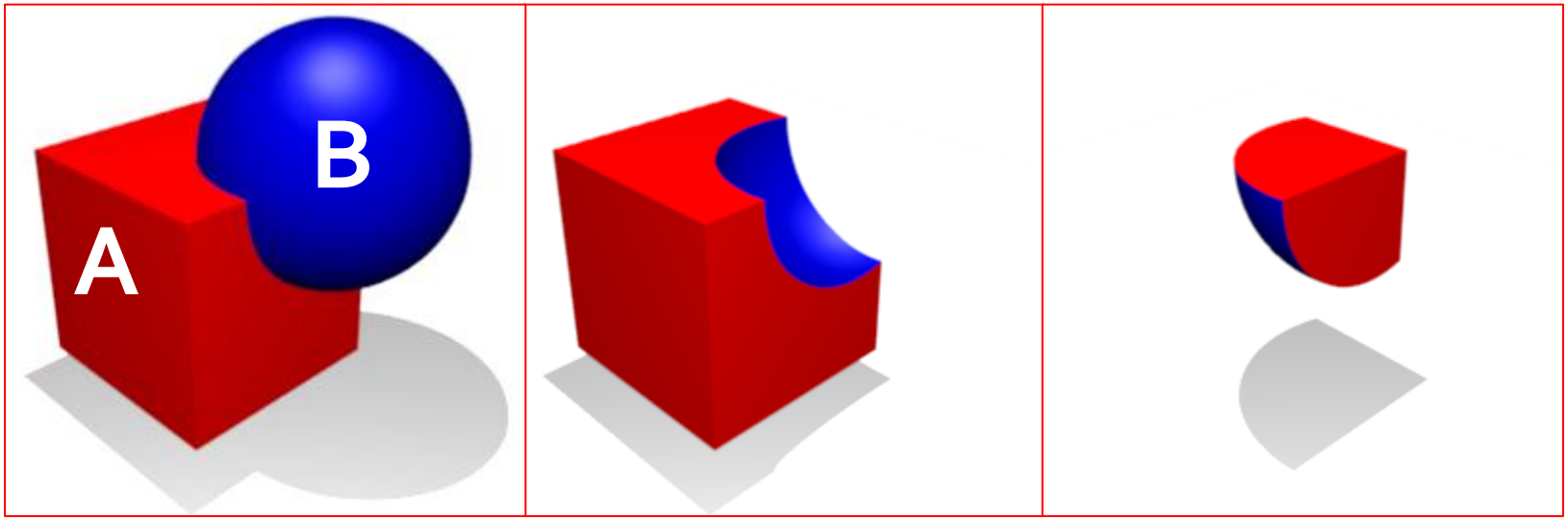
Constructive Solid Geometry



http://upload.wikimedia.org/wikipedia/commons/8/8b/Csg_tree.png

Binary operations for shape

Constructive Solid Geometry

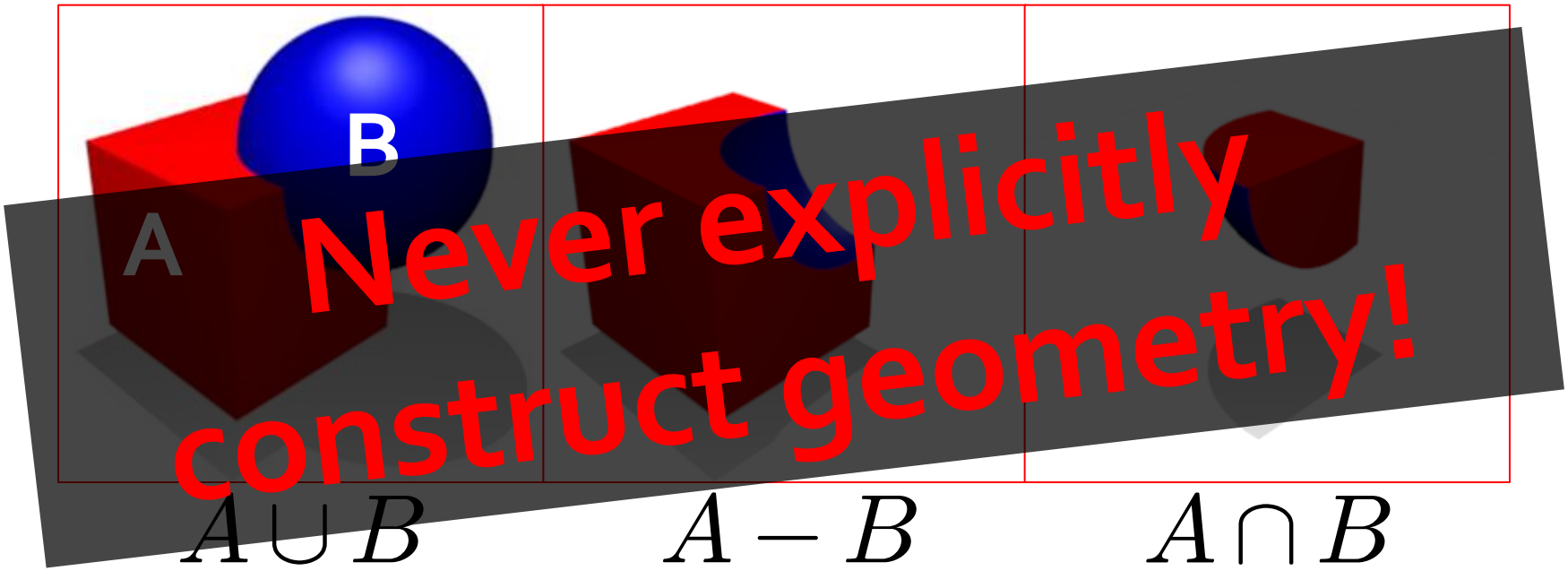


$A \cup B$ $A - B$ $A \cap B$

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

Binary operations for shape

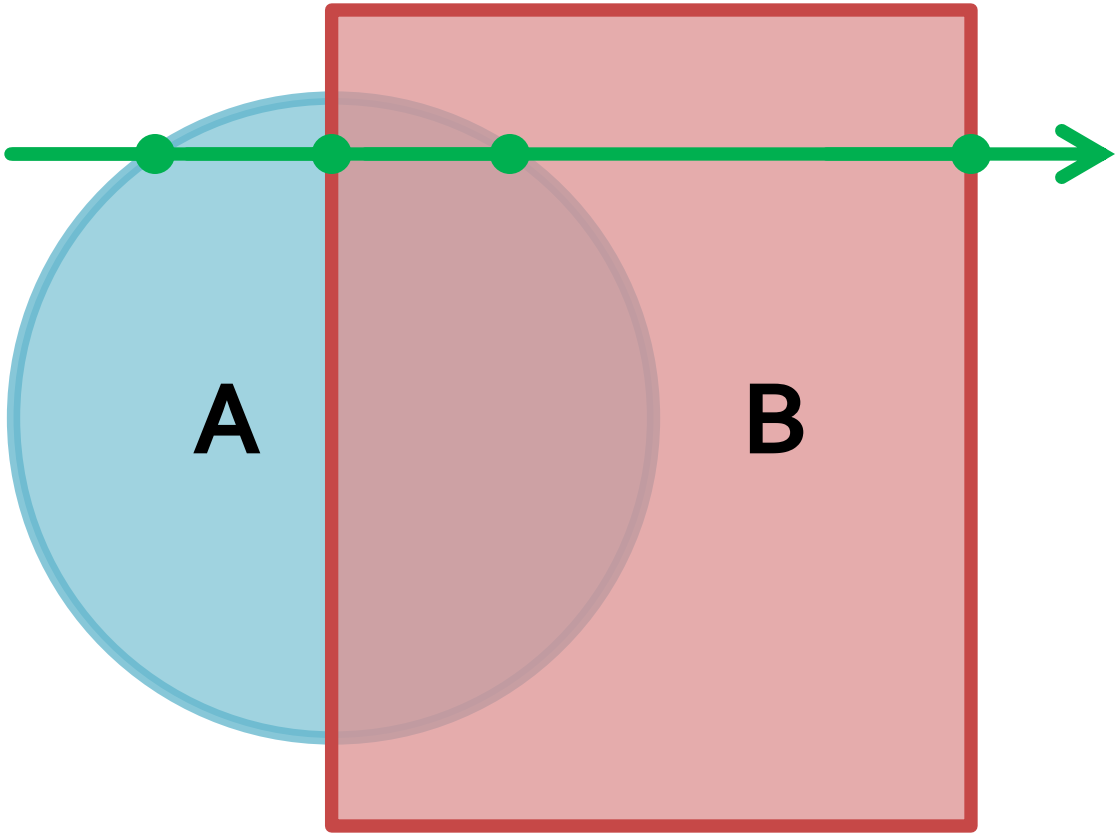
Constructive Solid Geometry



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

Binary operations for shape

Constructive Solid Geometry



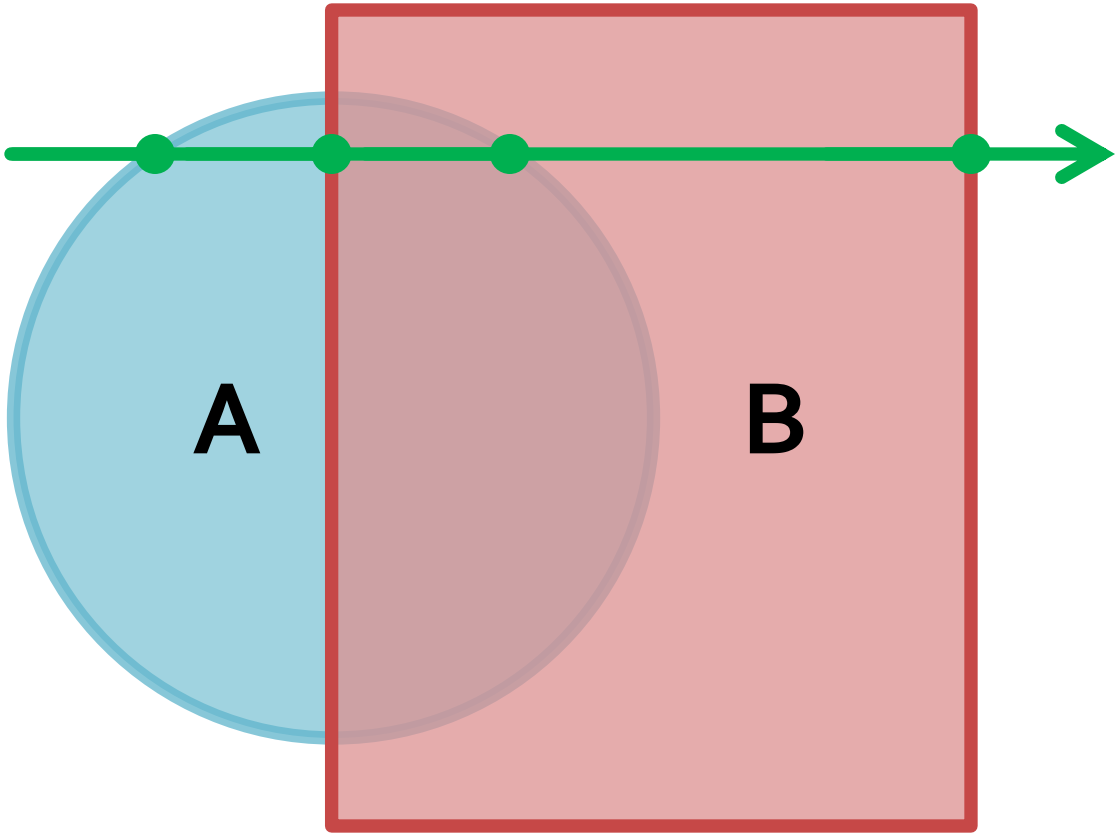
$$in(A) = [1, 3]$$

$$in(B) = [2, 6]$$

$$in(A \cap B) =$$

Store "inside intervals"

Constructive Solid Geometry



$$in(A) = [1, 3]$$

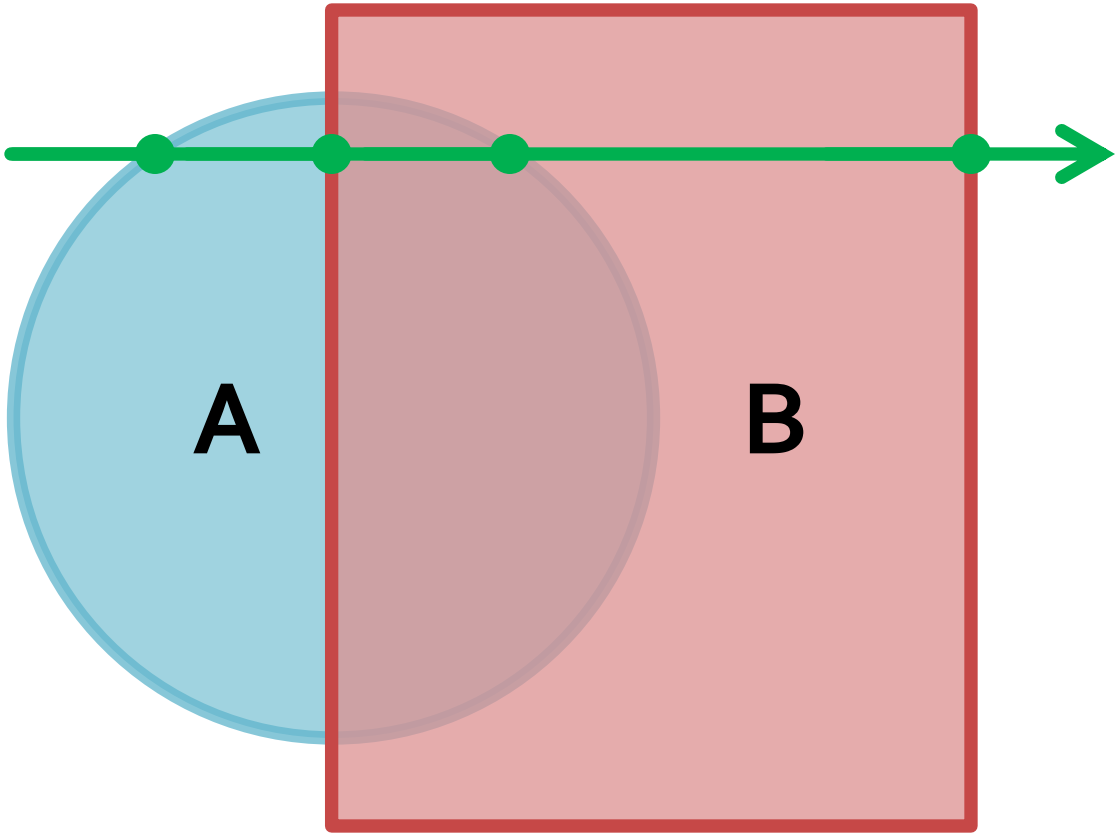
$$in(B) = [2, 6]$$

$$in(A \cap B) = [2, 3]$$

$$in(A \cup B) =$$

Store "inside intervals"

Constructive Solid Geometry



$$in(A) = [1, 3]$$

$$in(B) = [2, 6]$$

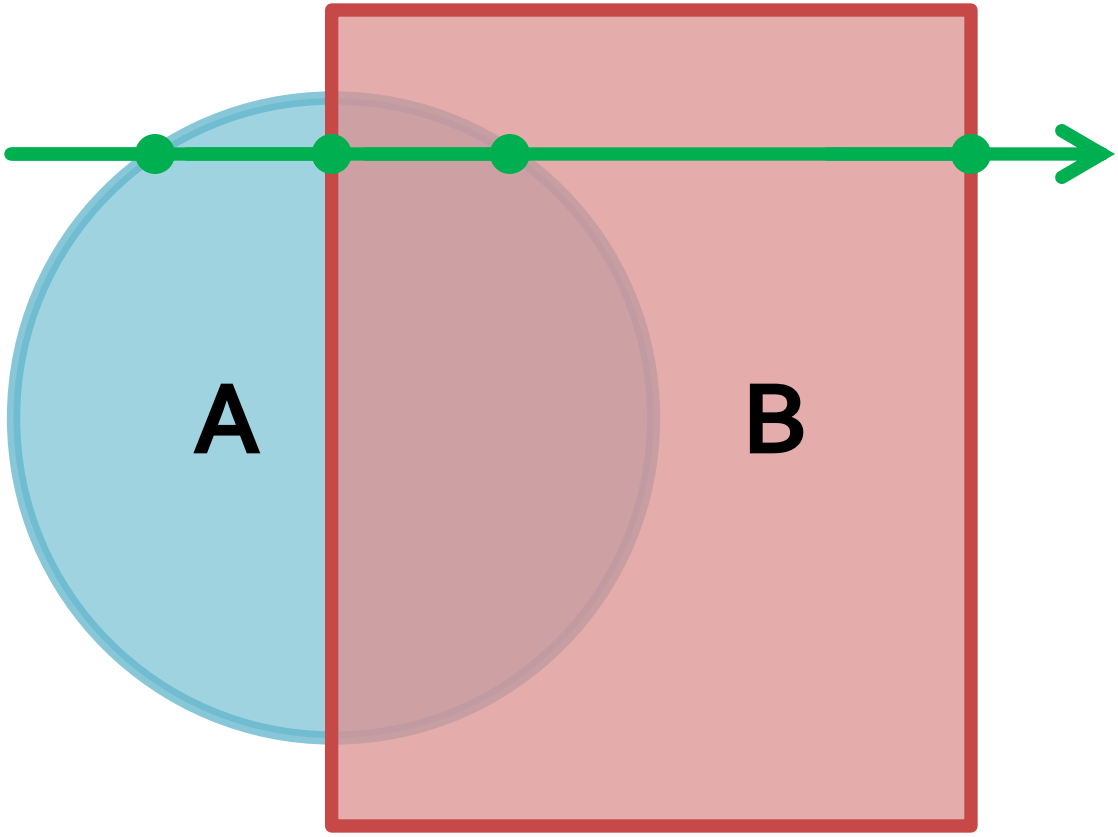
$$in(A \cap B) = [2, 3]$$

$$in(A \cup B) = [1, 6]$$

$$in(A - B) =$$

Store "inside intervals"

Constructive Solid Geometry



$$in(A) = [1, 3]$$

$$in(B) = [2, 6]$$

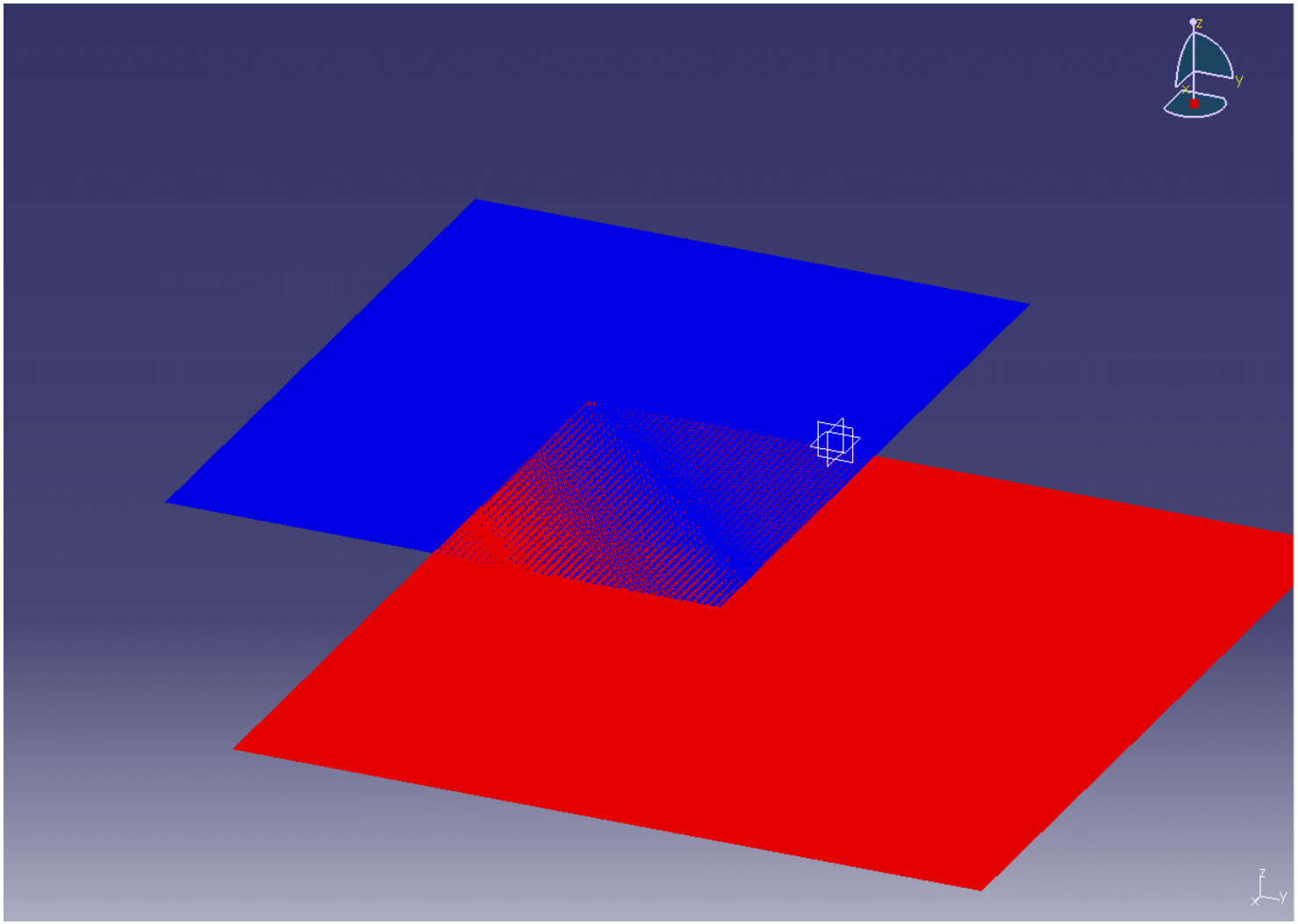
$$in(A \cap B) = [2, 3]$$

$$in(A \cup B) = [1, 6]$$

$$in(A - B) = [1, 2)$$

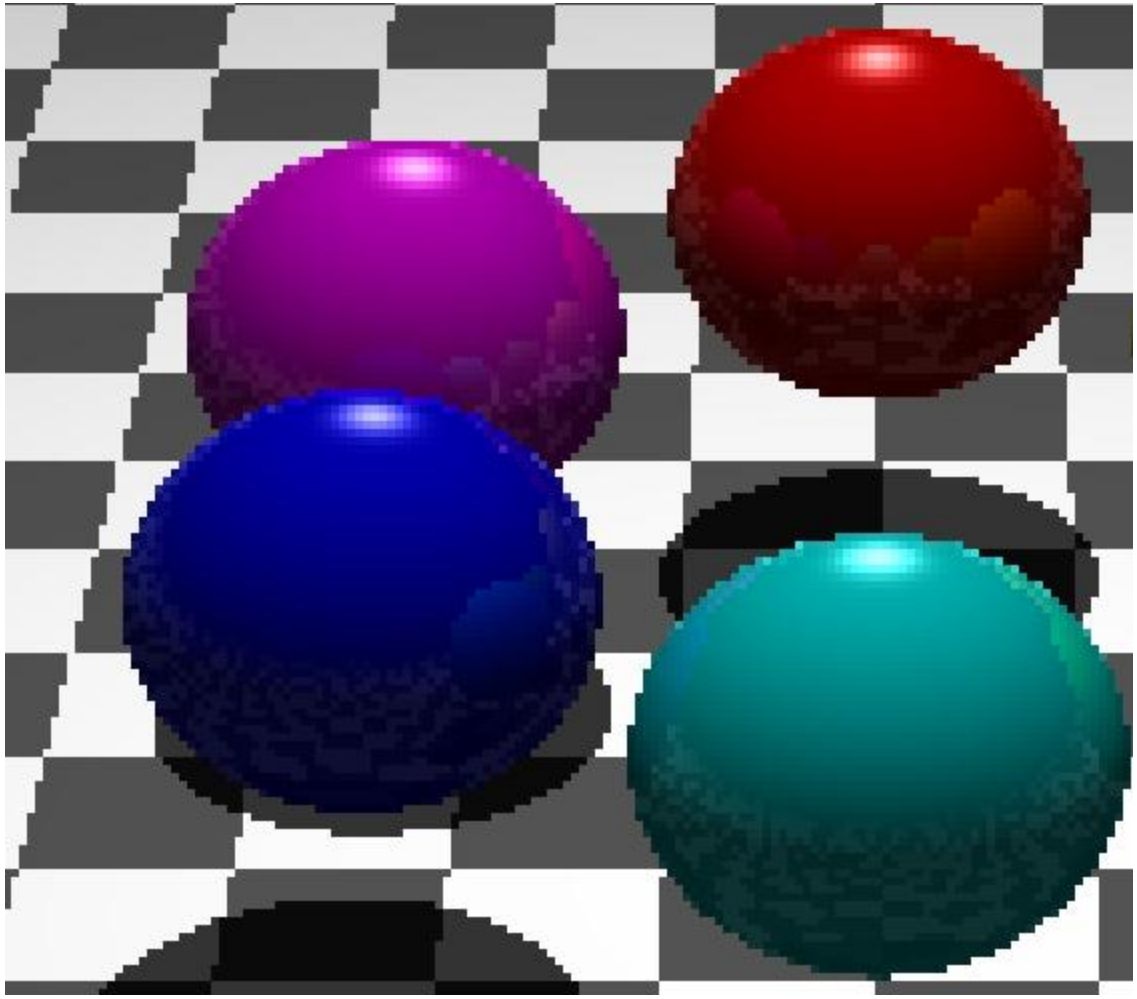
Store "inside intervals"

Recall: Depth Fighting



http://ps-2.kevo09.com/CATIA-B18/basug_C2/basugbt1510.htm

Aliasing: Still Problematic!

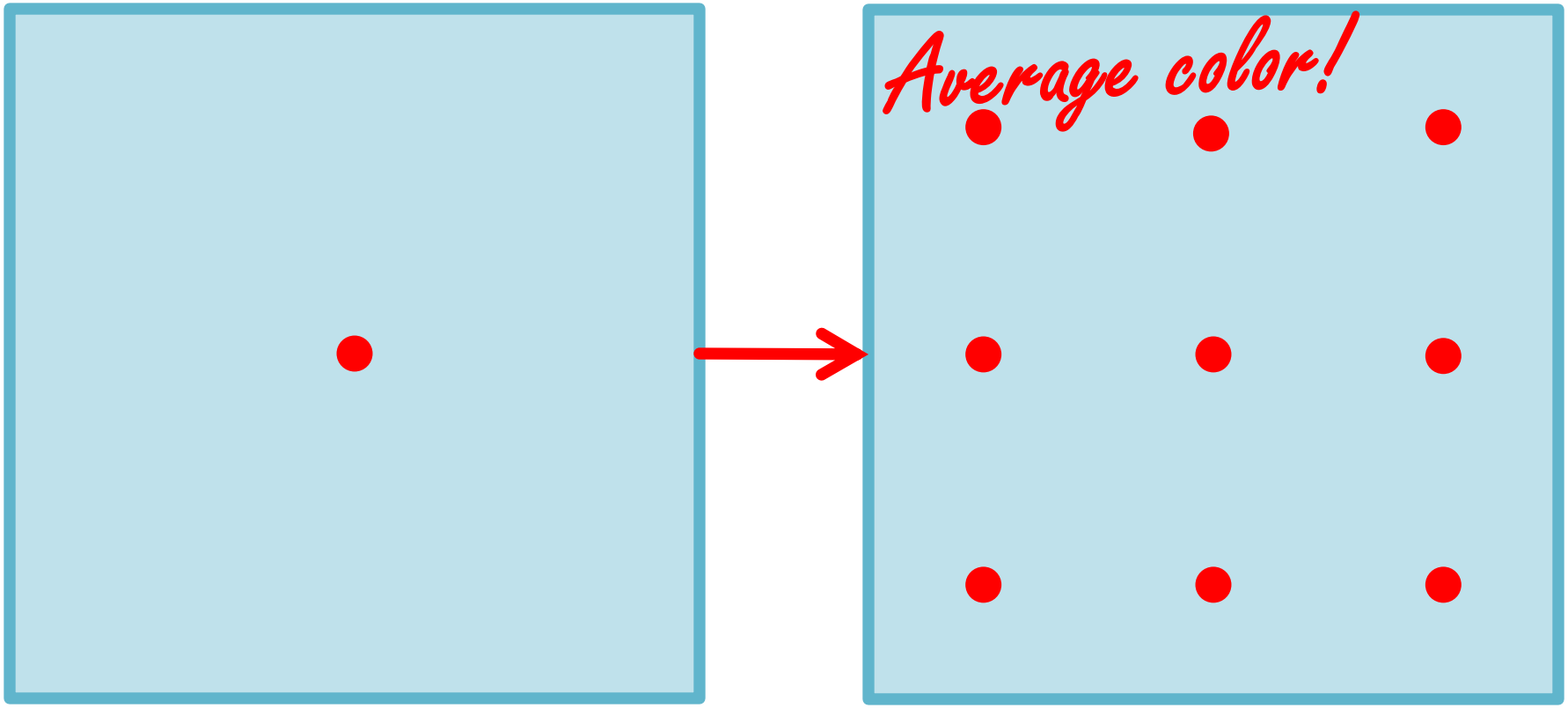


http://www.hackification.com/2008/08/31/experiments-in-ray-tracing-part-8-anti-aliasing/

Distribution Ray Tracing

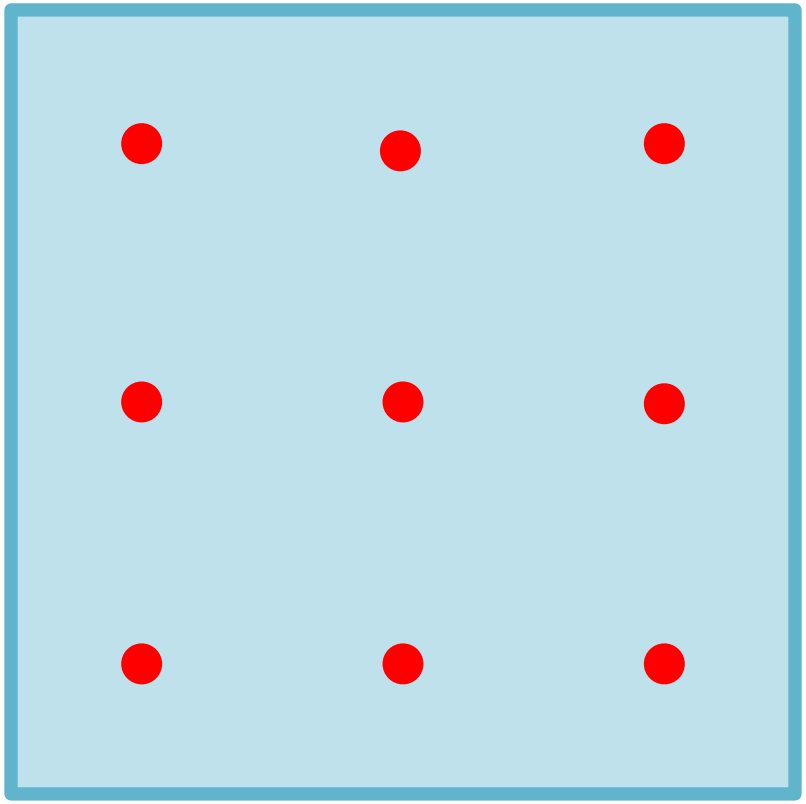
Scatter rays to make ray tracing less crisp.

Distribution Antialiasing



Multiple rays per pixel

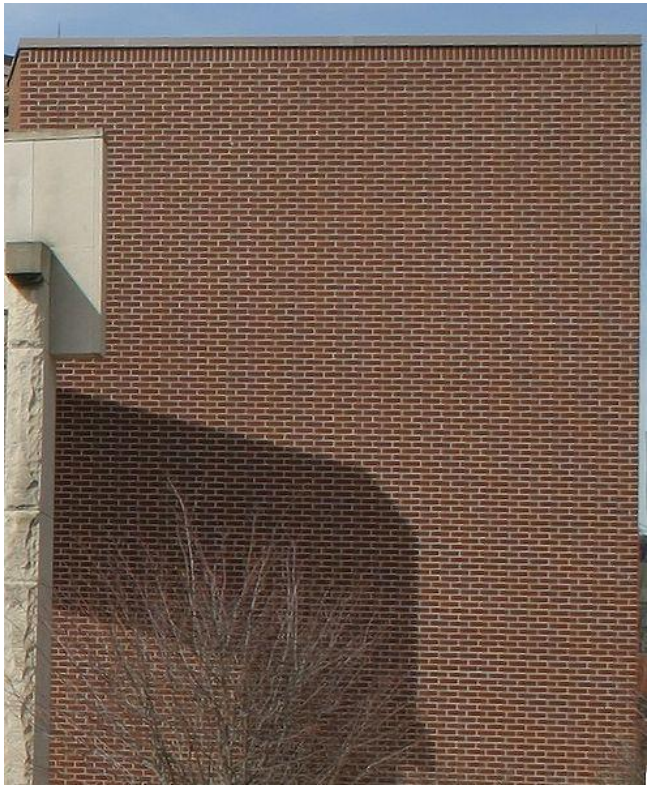
Distribution Antialiasing

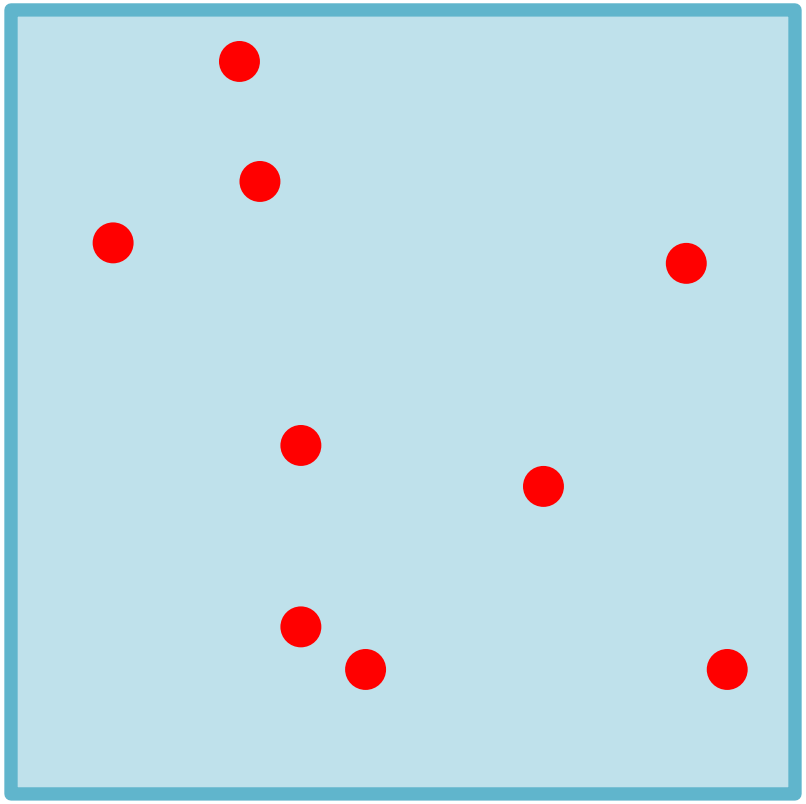


http://upload.wikimedia.org/wikipedia/commons/f/fb/Moire_pattern_of_bricks_small.jpg

Multiple rays per pixel

Random Sampling

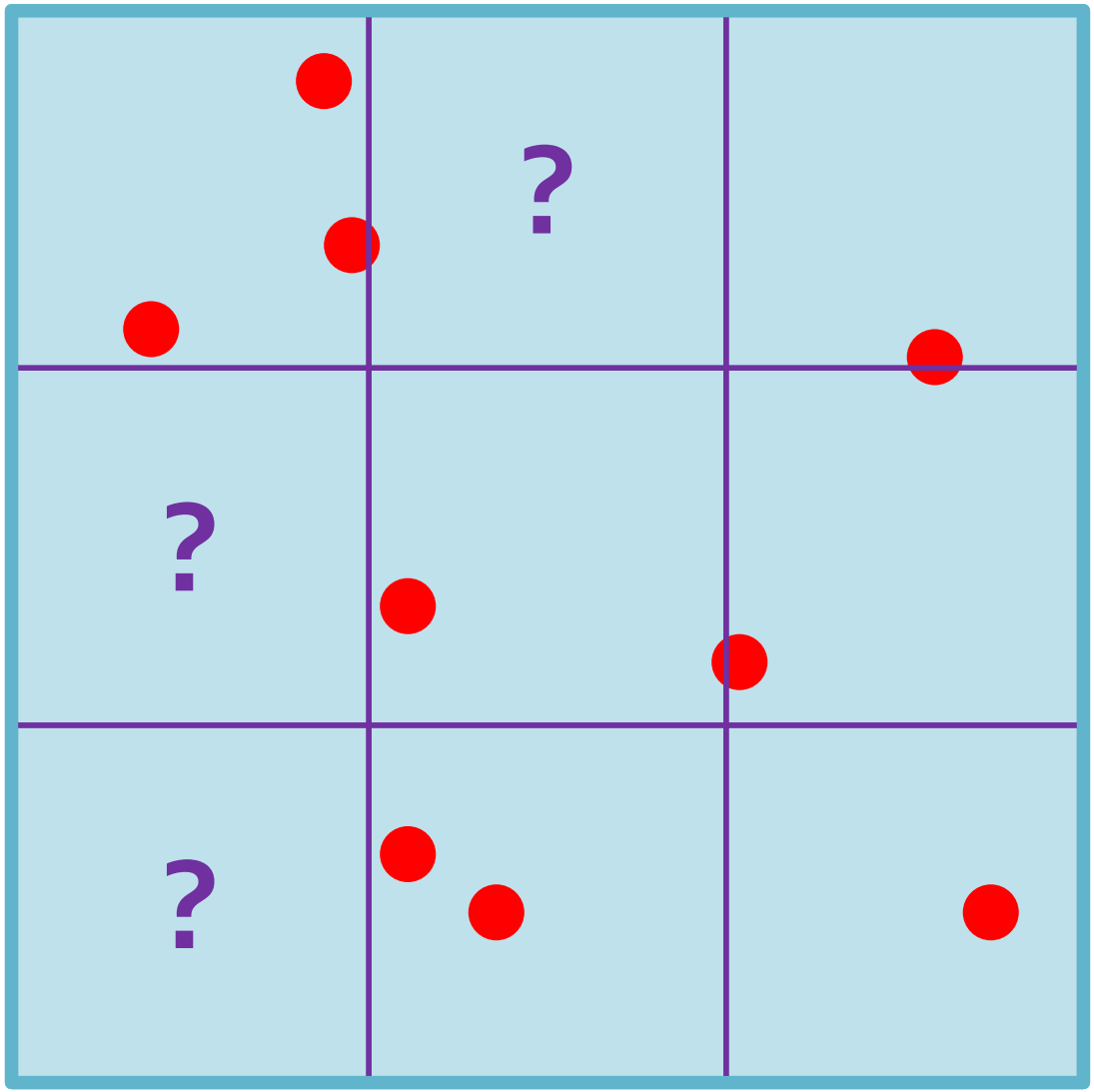




http://en.wikipedia.org/wiki/File:Moire_pattern_of_bricks.jpg

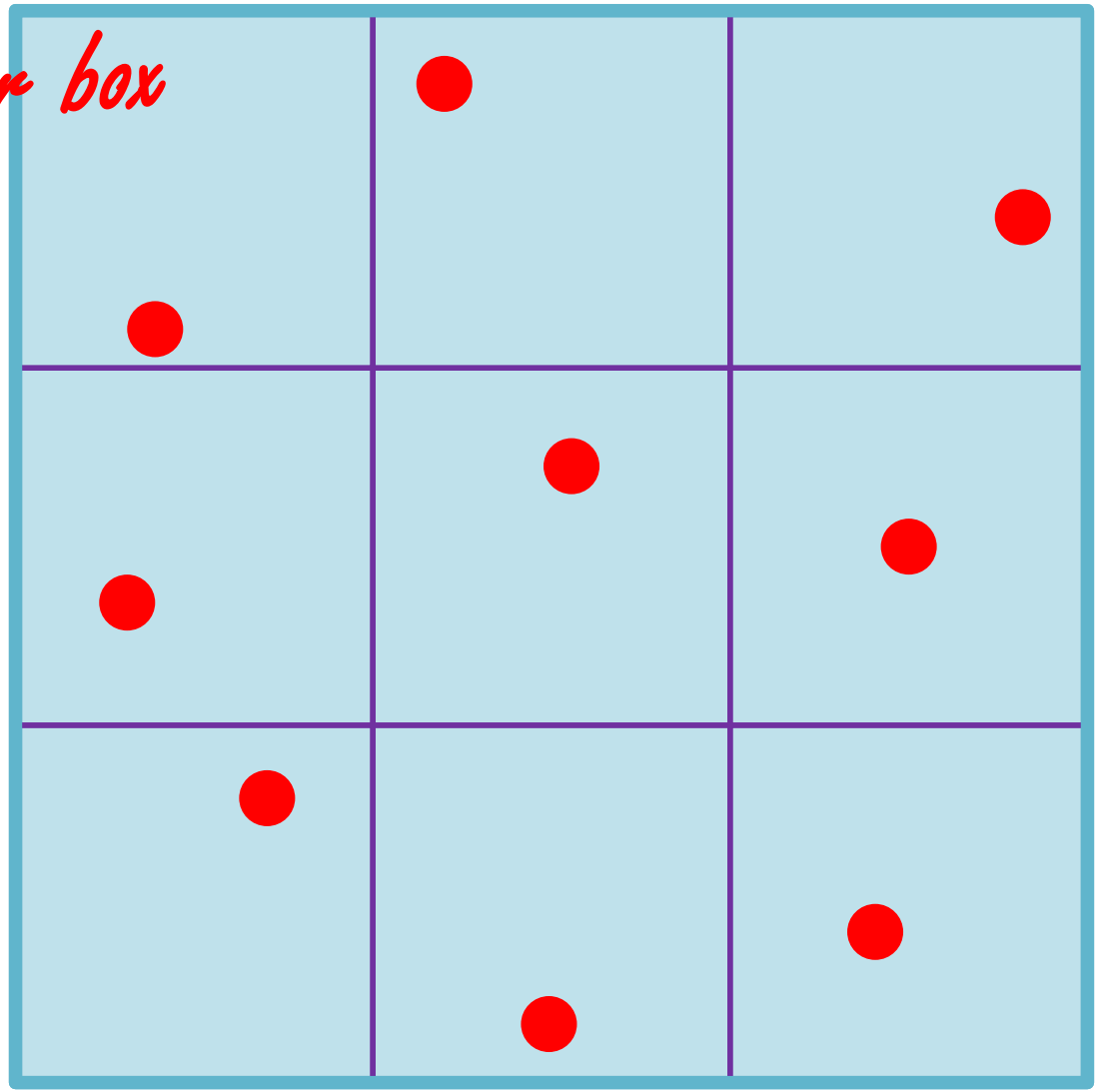
Remove Moiré patterns

Not Enough Samples!

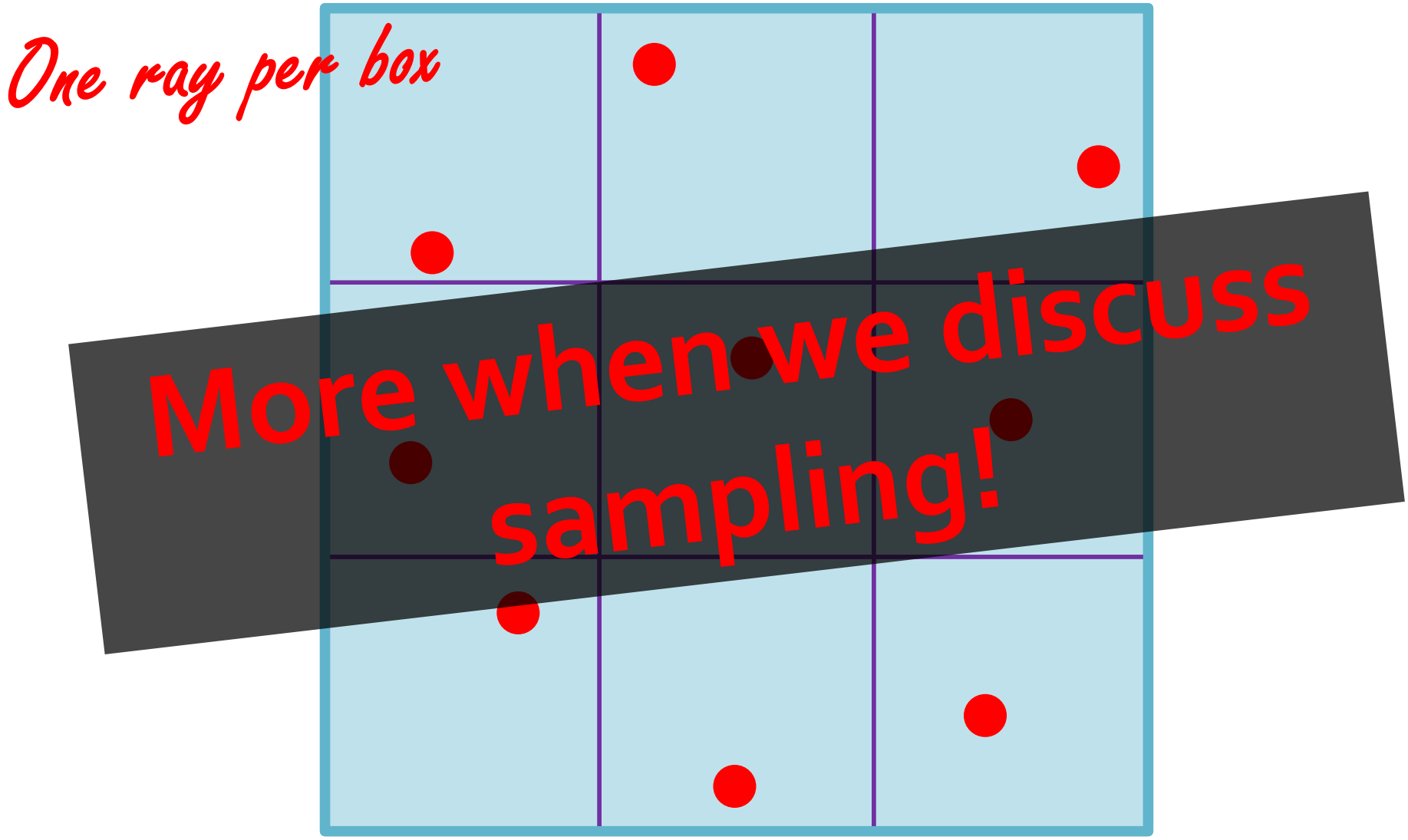


Stratified (Jittered) Sampling

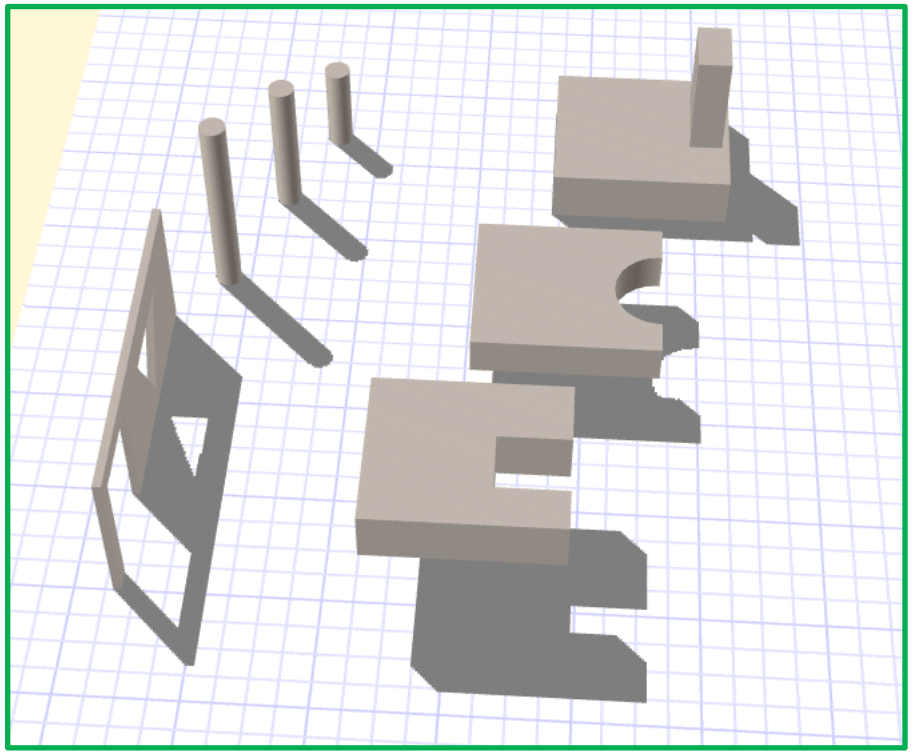
One ray per box



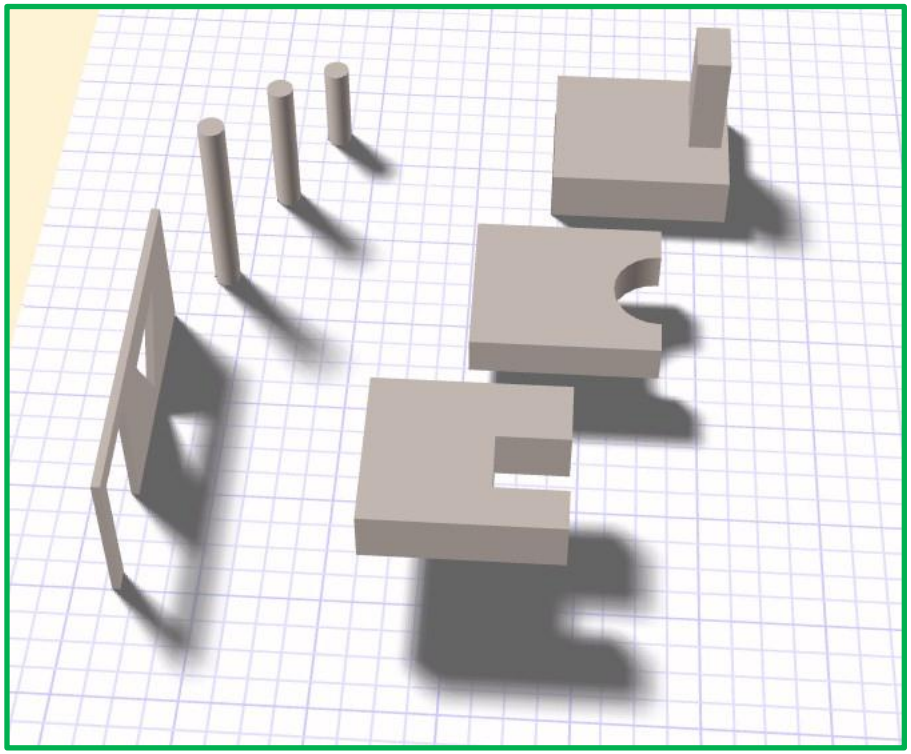
Stratified (Jittered) Sampling



Soft Shadows



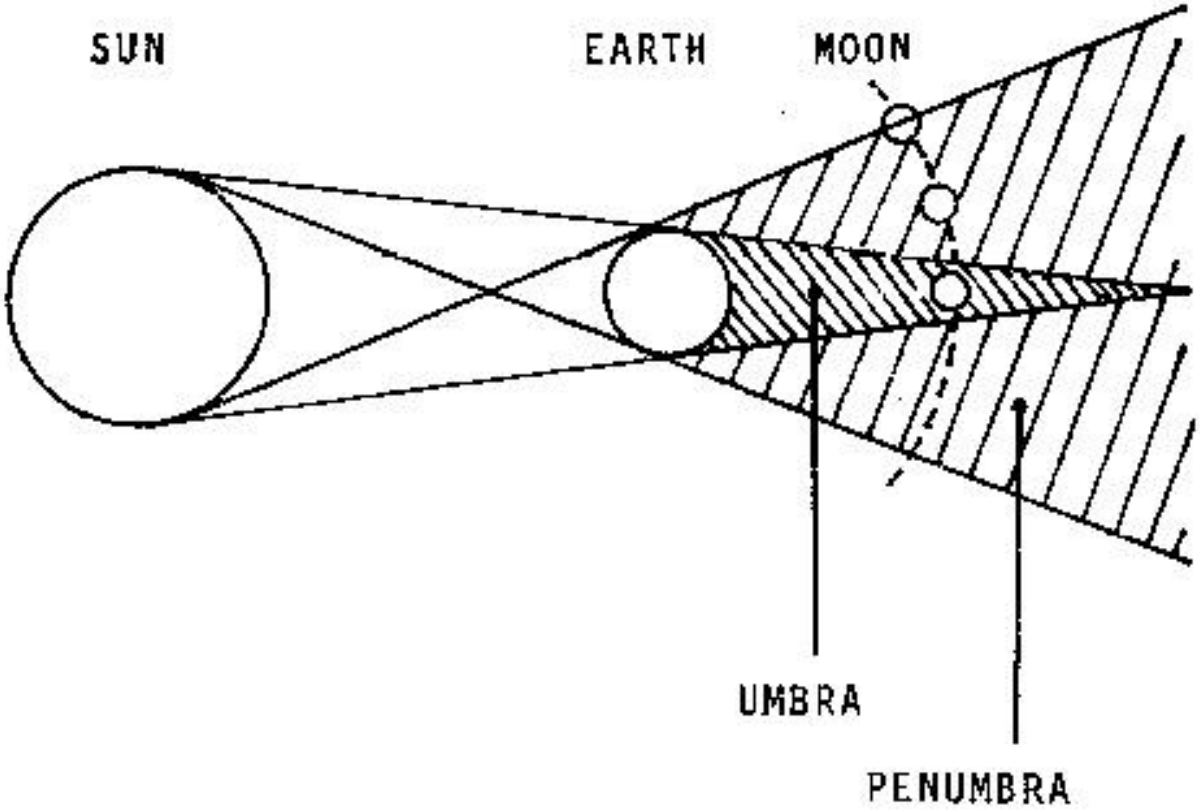
Hard shadows



Soft shadows

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

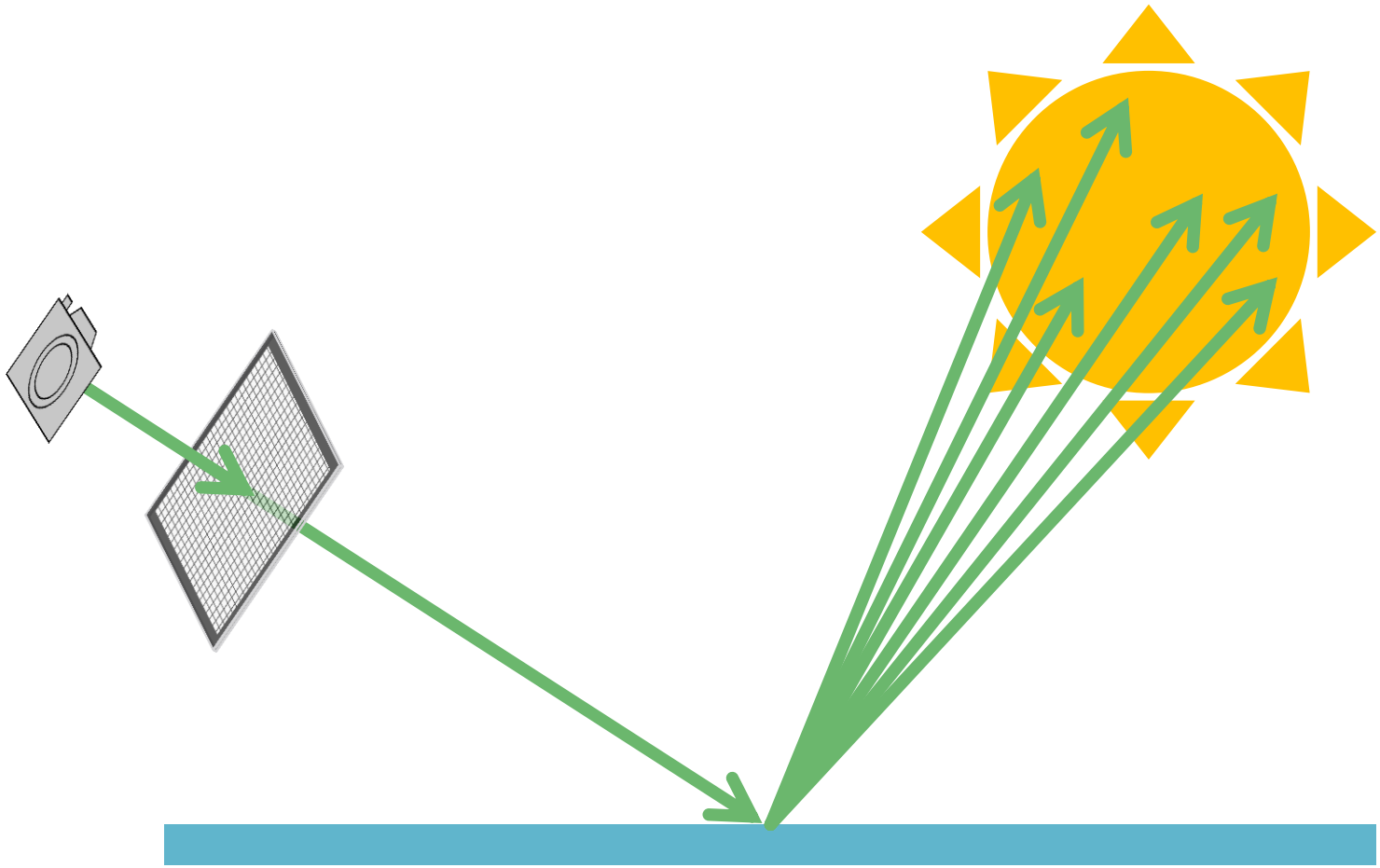
Soft Shadows



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

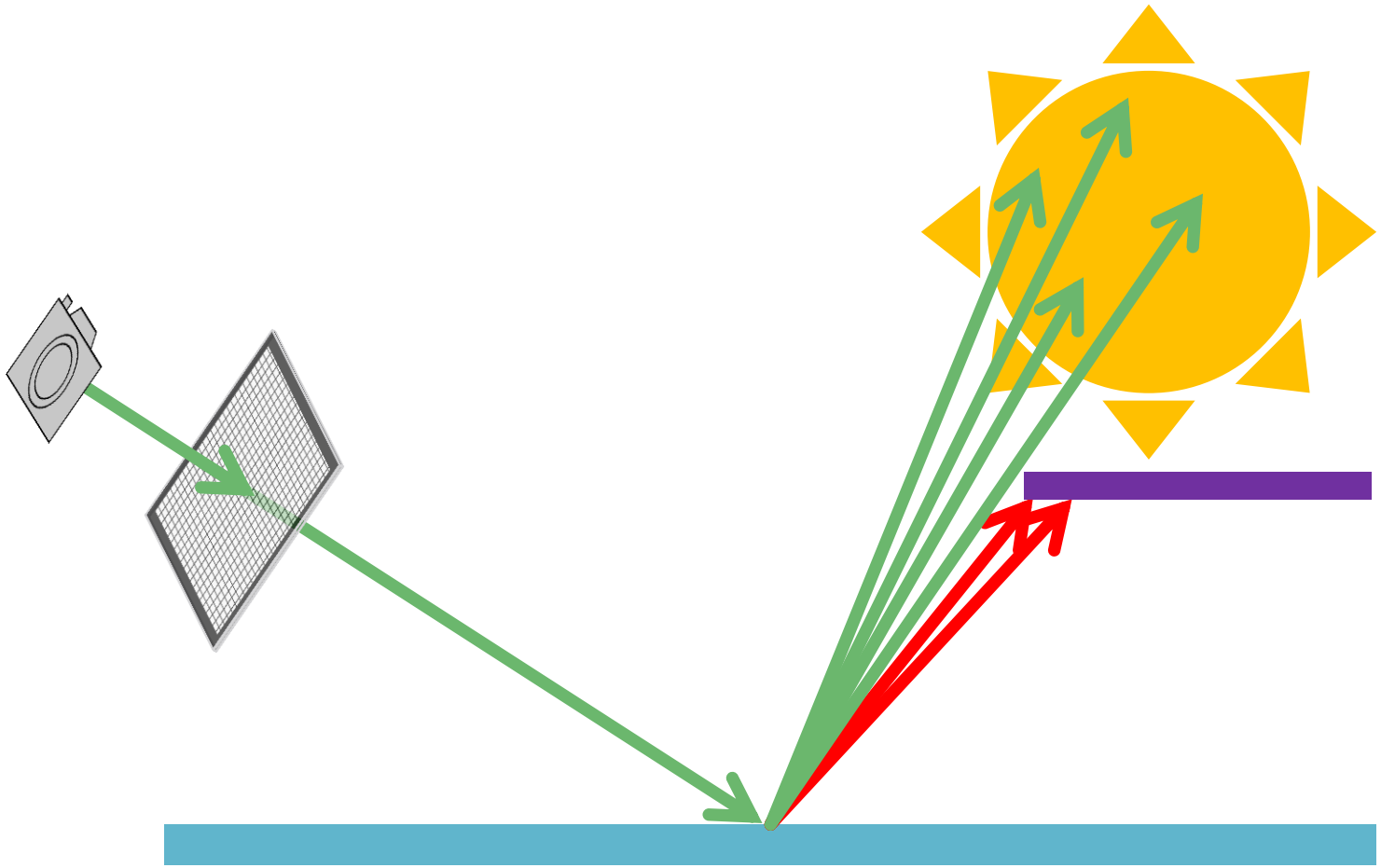
Lights aren't all point sources

Distribution Soft Shadows



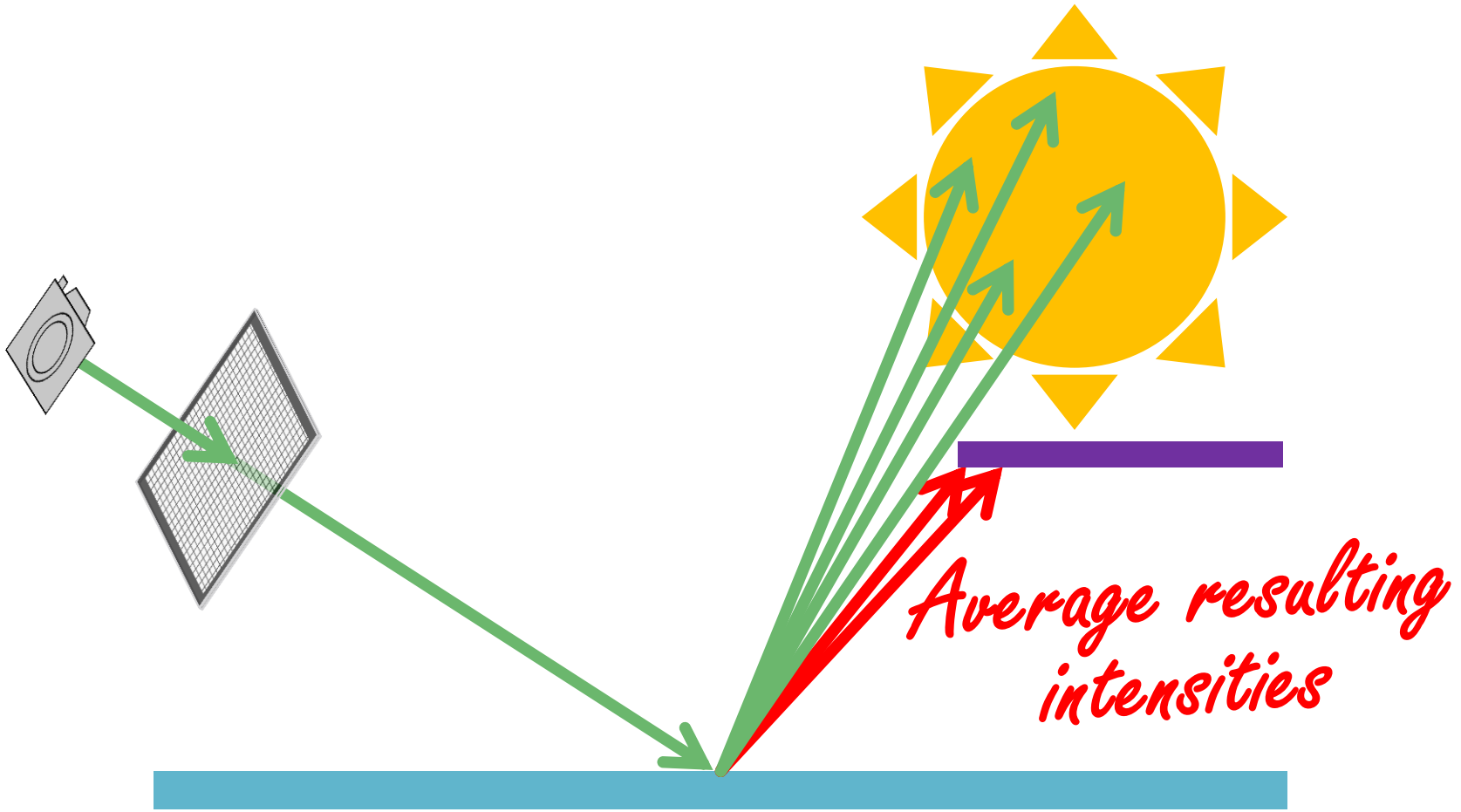
Randomly sample light rays

Distribution Soft Shadows



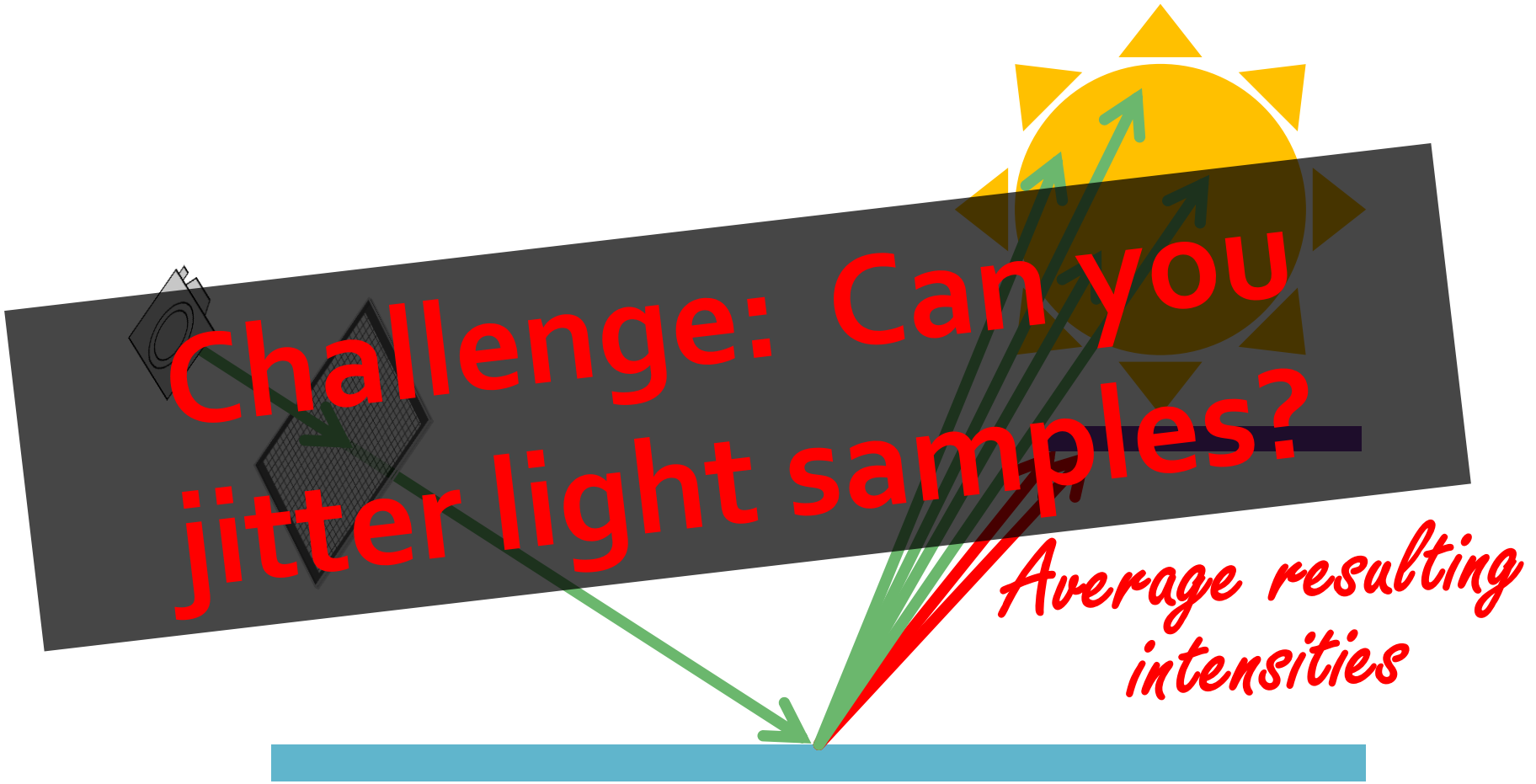
Shadow computed per ray

Distribution Soft Shadows



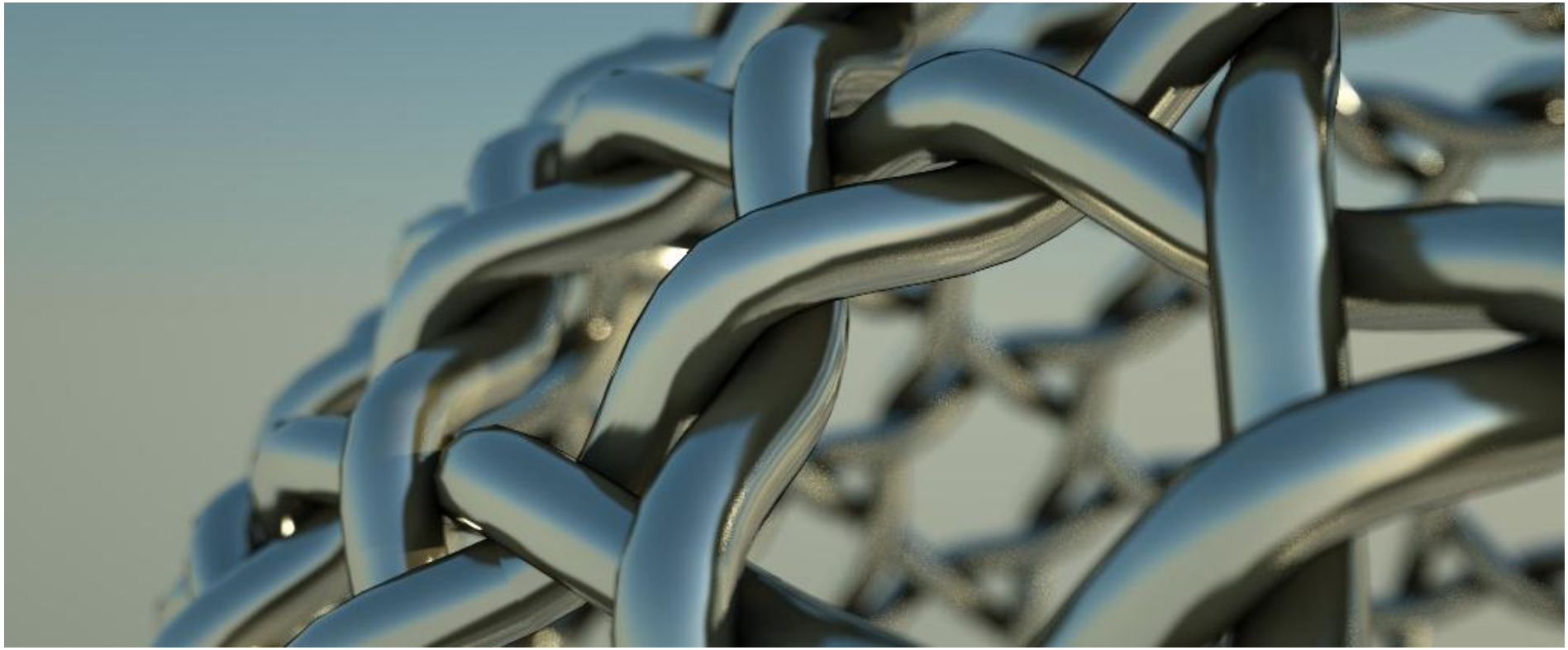
Shadow computed per ray

Distribution Soft Shadows



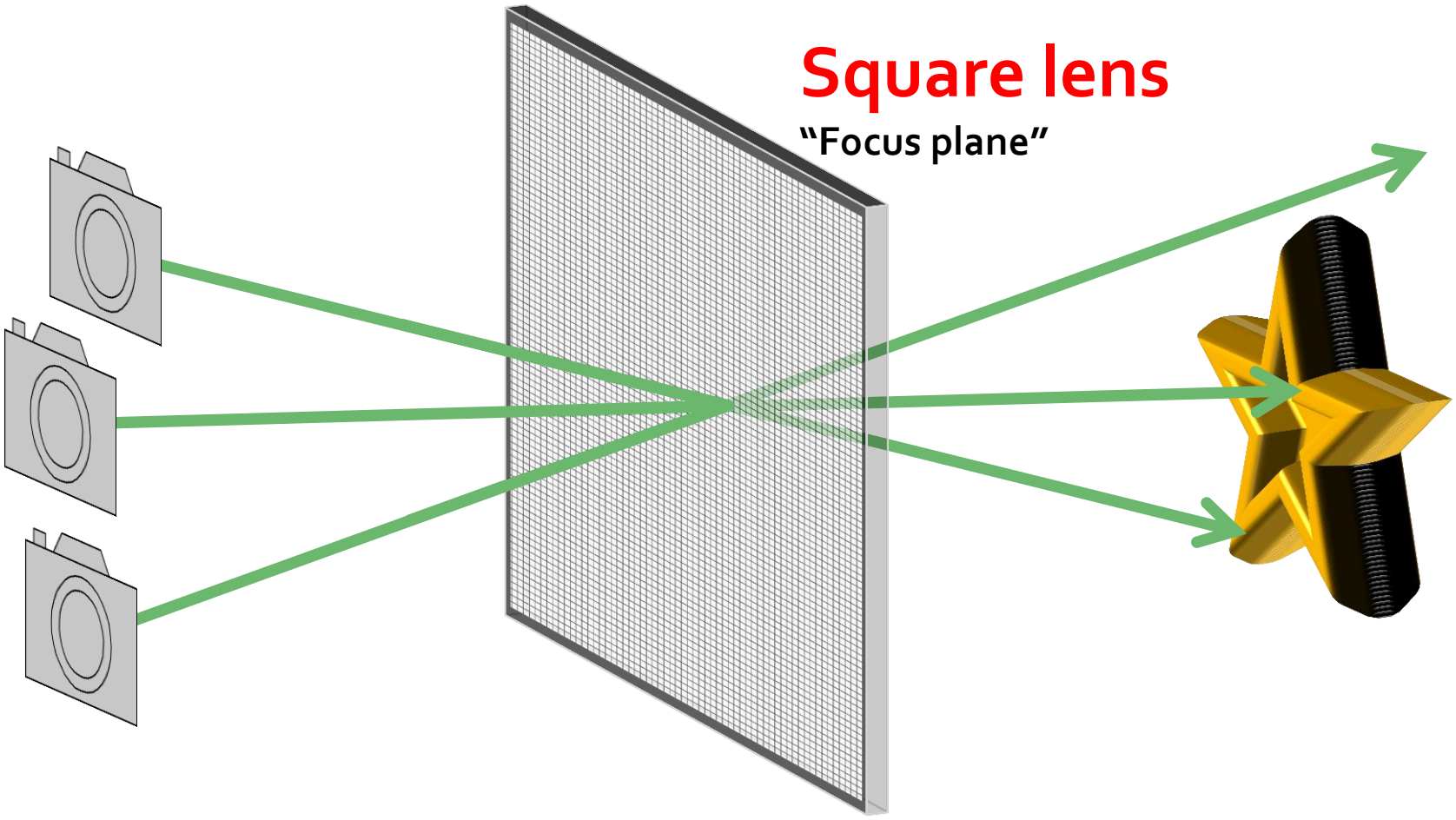
Shadow computed per ray

Depth of Field



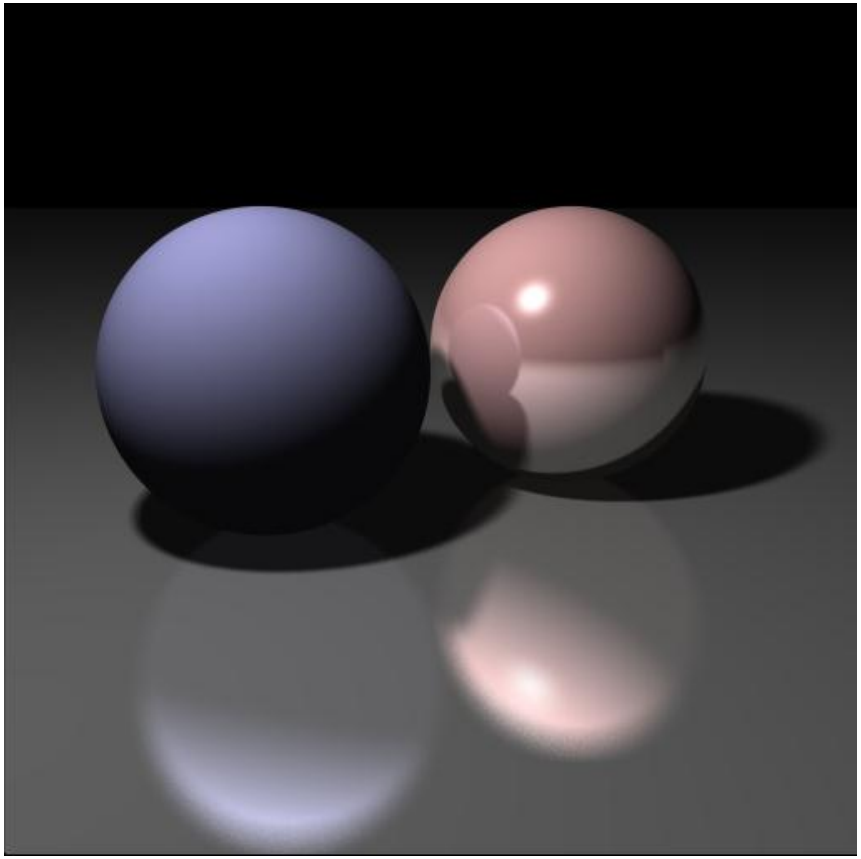
http://liam887.files.wordpress.com/2010/08/weaver.jpg

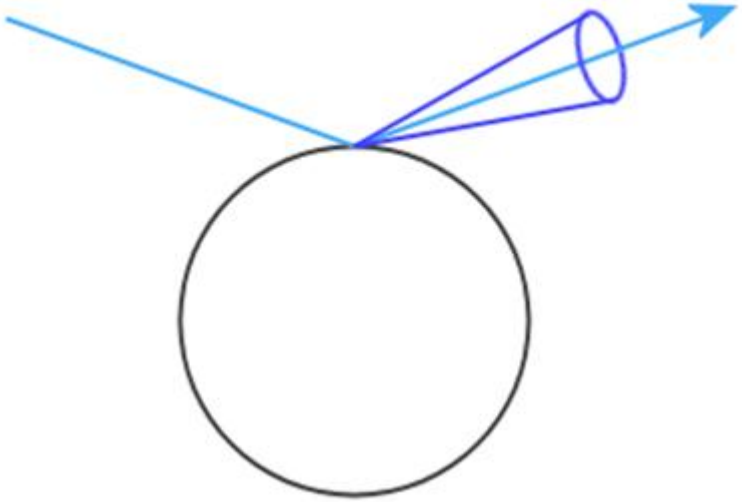
Distribution Depth of Field



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

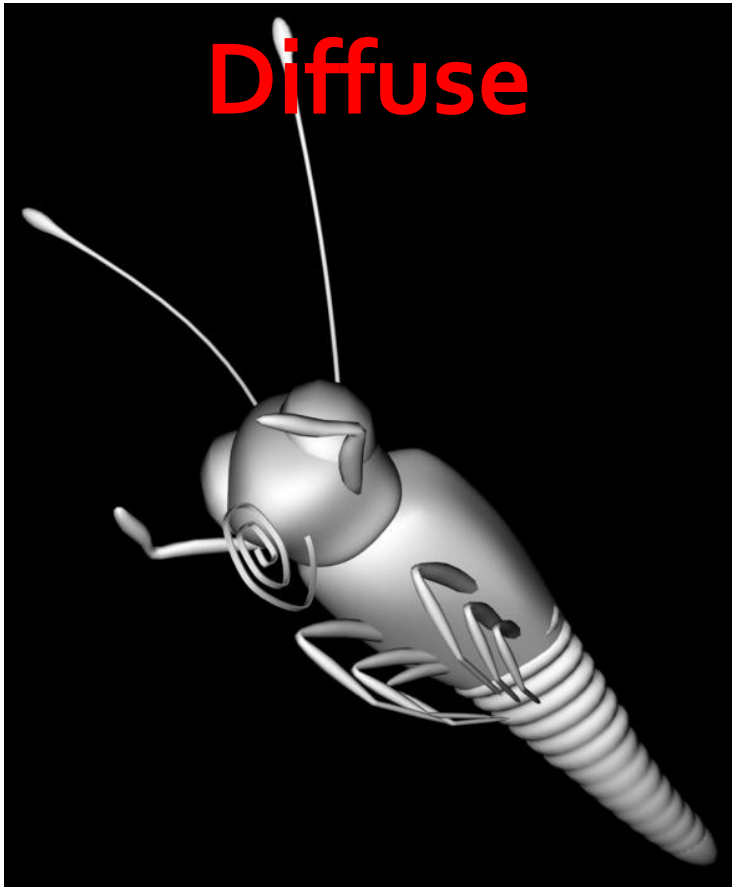
Randomly sample reflected rays

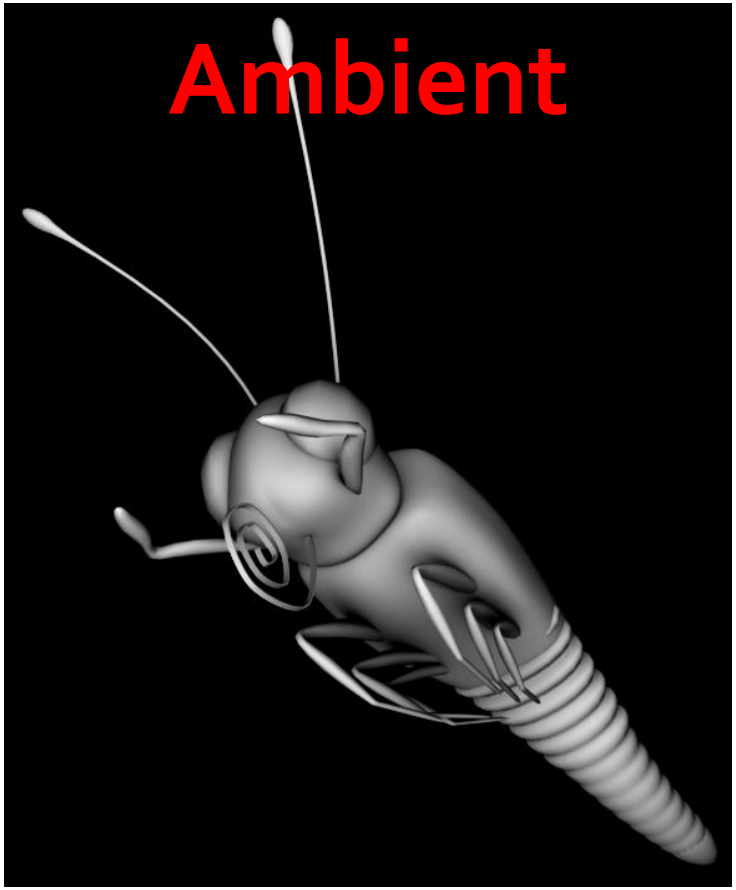
Motion Blur

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

Randomly sample positions

Ambient Occlusion

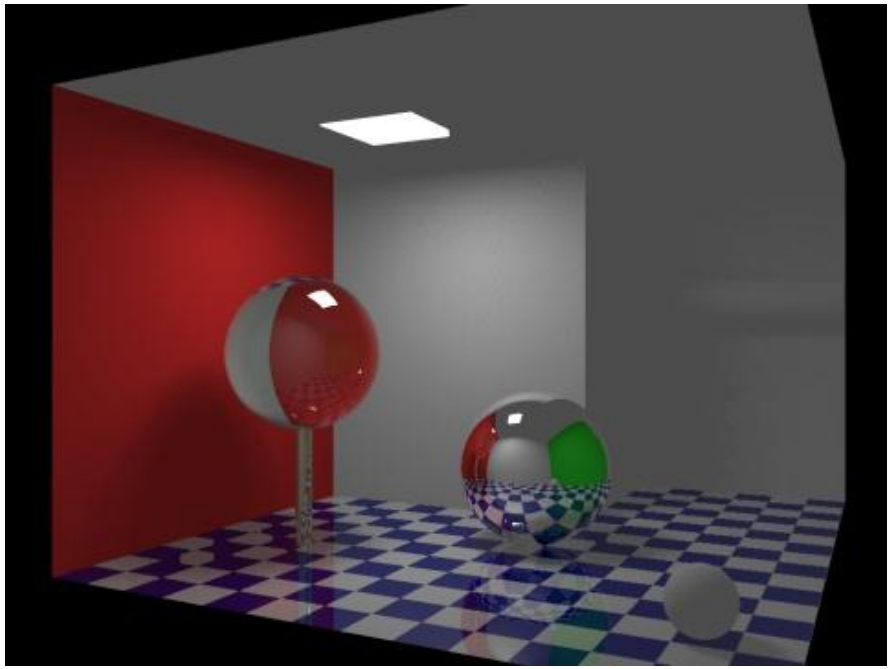


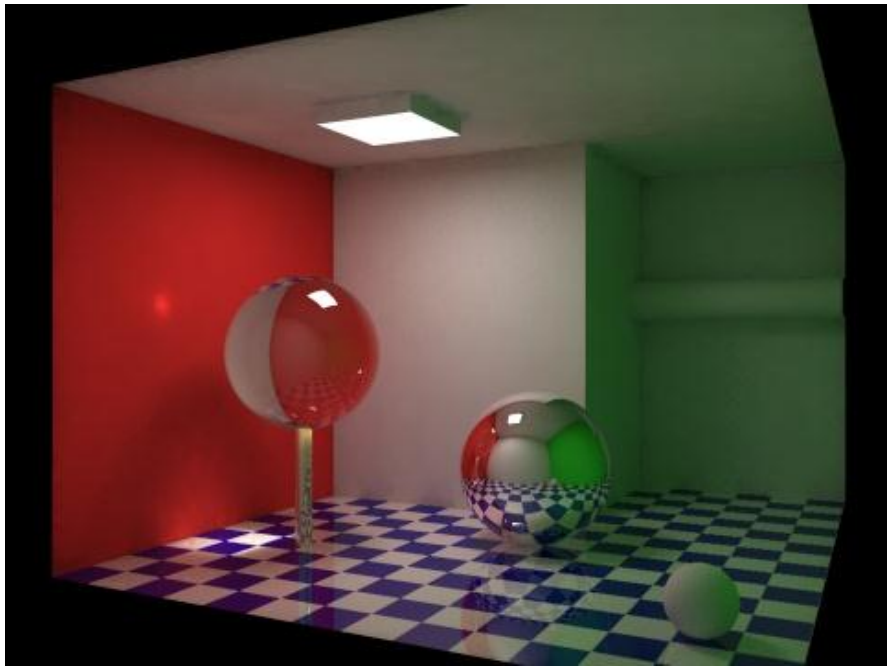


http://en.wikipedia.org/wiki/Ambient_occlusion

Brightness depends on nearby geometry

Global Illumination

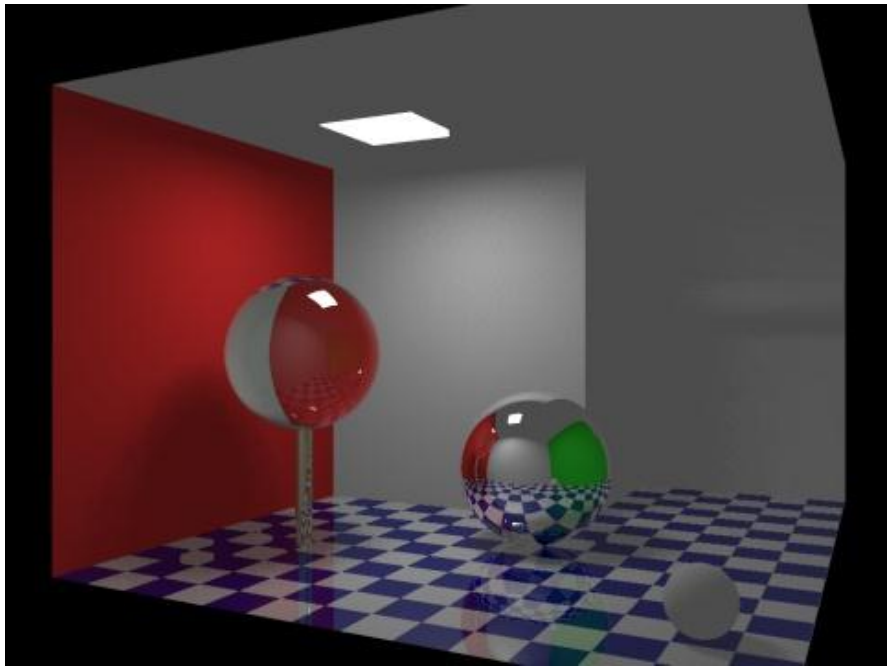


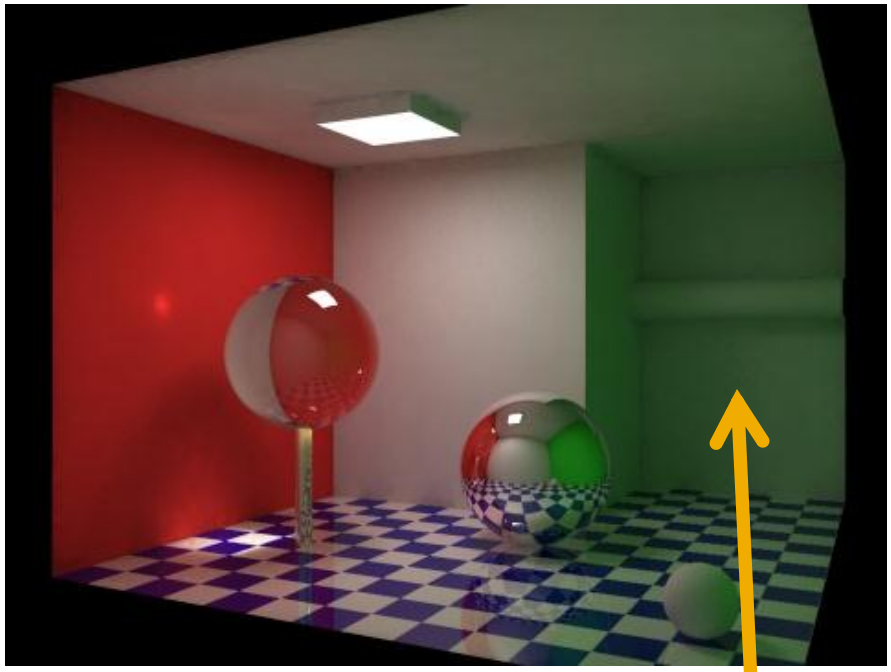


http://en.wikipedia.org/wiki/Global_illumination

Account for indirect lighting

Global Illumination



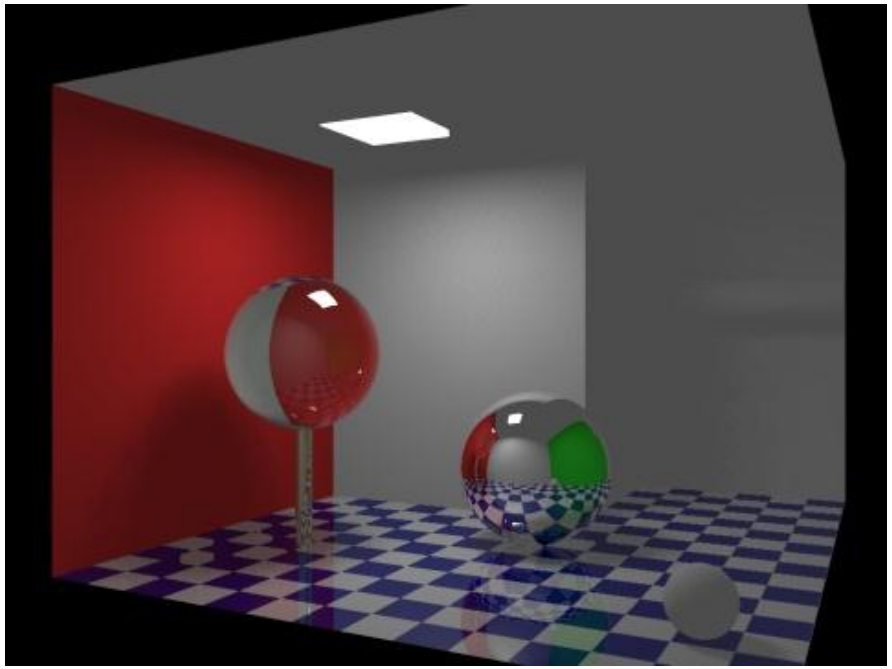


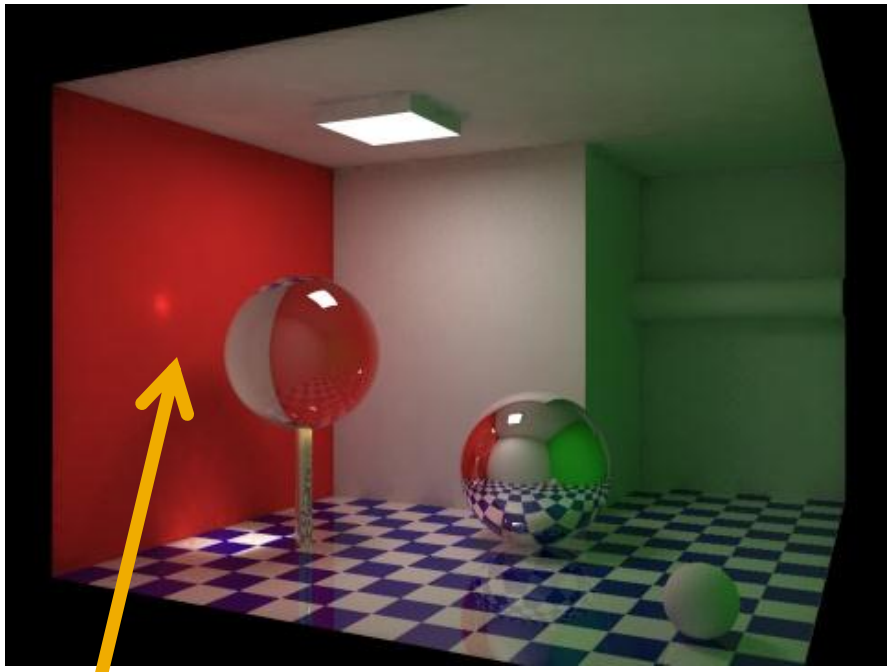
Diffuse inter-reflection

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

Account for indirect lighting

Global Illumination





Caustics

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

Account for indirect lighting

Asymptotic Speed

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

- Rasterization:

  $O(bm)$

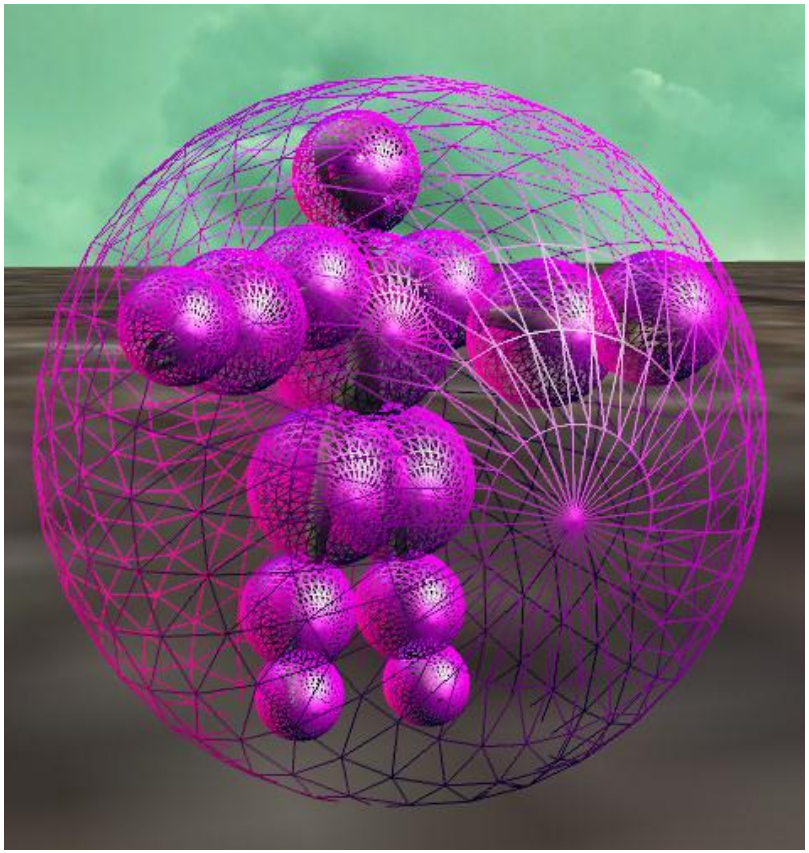
- Ray tracing:

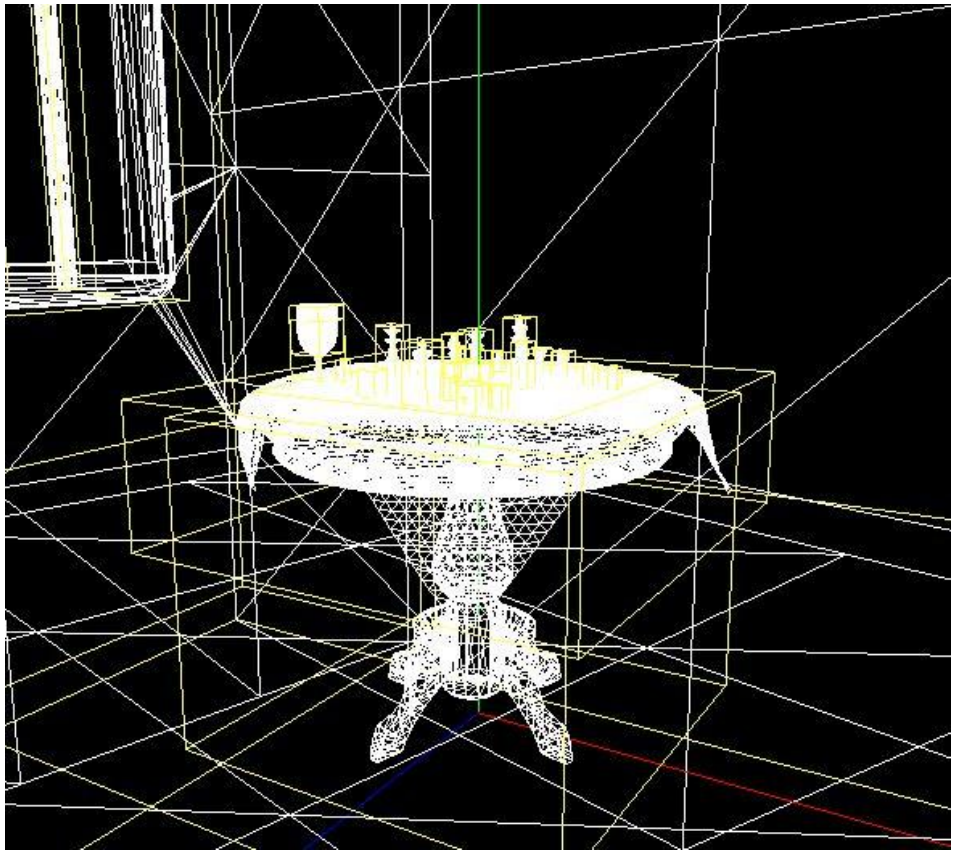
  $O(bp)$

- Accelerated ray tracing:

  $O(p \log b)$

Accelerated Ray Tracing

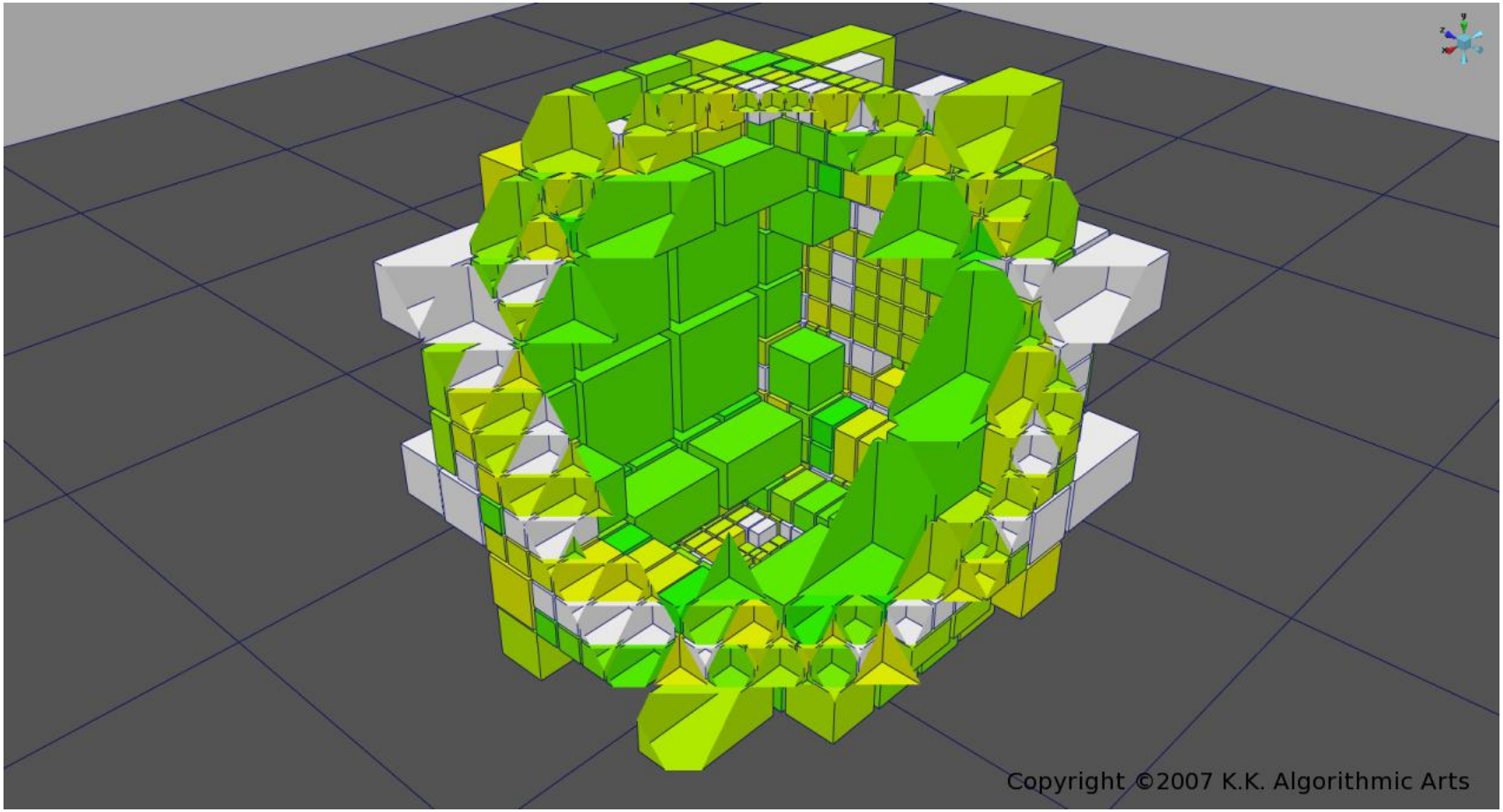




http://sopra.le-gousteau.de/images/2/2e/Bv_bvh_bspheres.jpg
http://graphics.ucsd.edu/courses/rendering/2004/ssaha/index_files/finalbvh.jfif

Bounding Volume Hierarchy (BVH)

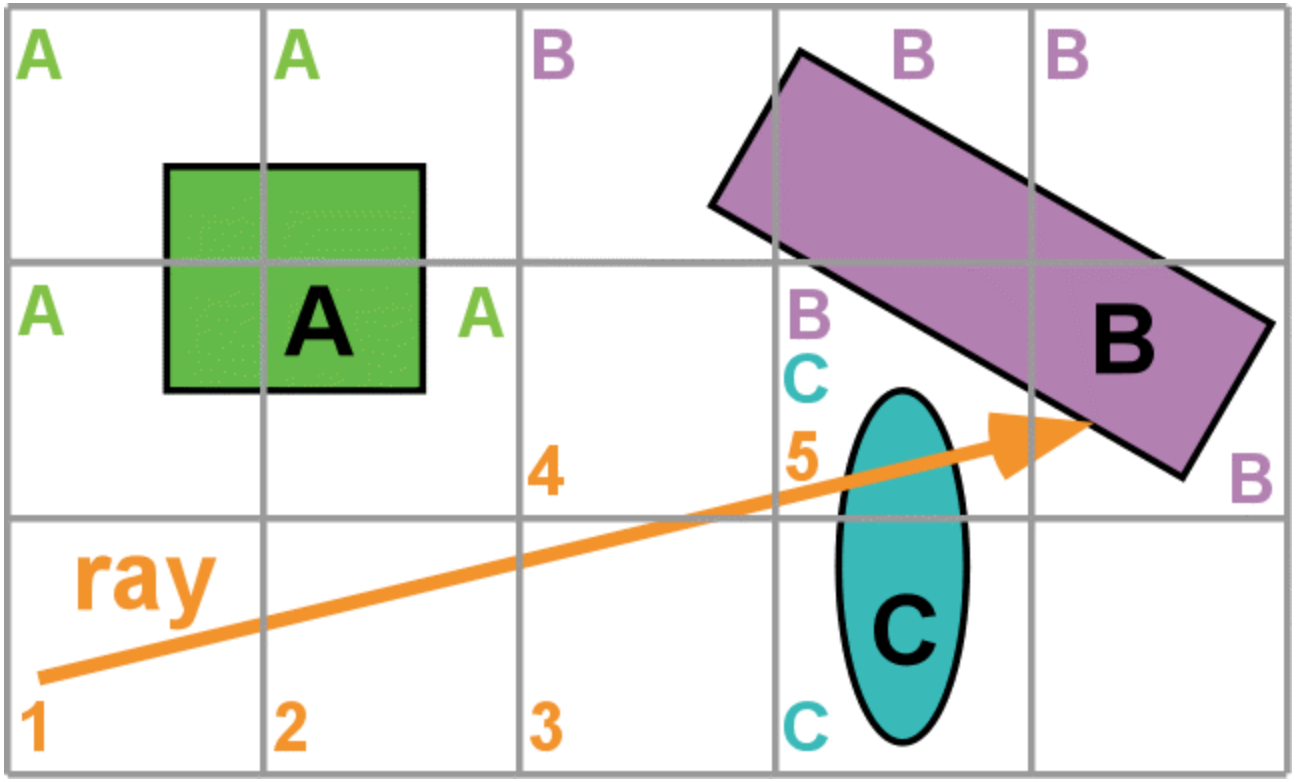
Accelerated Ray Tracing



http://algorithmicarts.com/images/2008/bsptree-cutaway1269x684.png

Binary Space Partitioning (BSP) trees

Accelerated Ray Tracing



http://www.cs.berkeley.edu/~sequin/CS184/IMGS/spatialSubdiv.gif

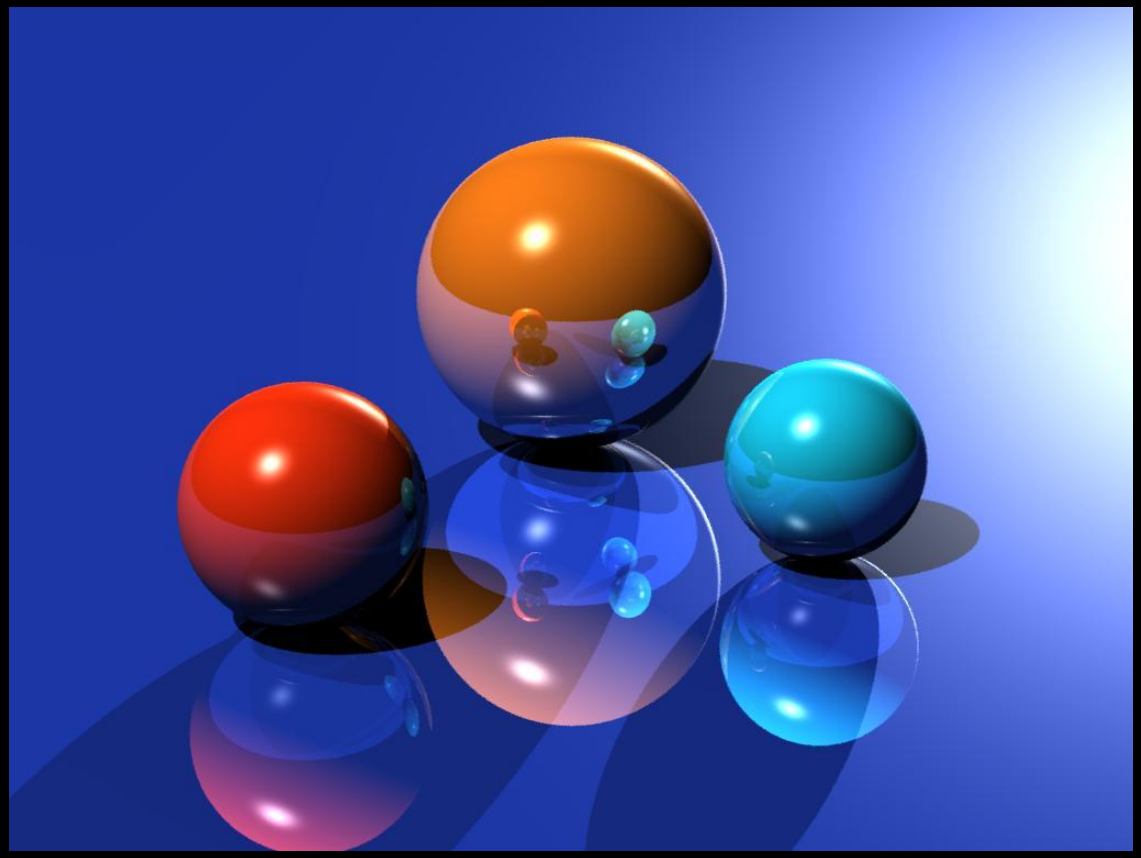
Uniform subdivision grid

Challenge: Real-Time Ray Tracing

[NVIDIA, SIGGRAPH 2008]

"Embarrassingly parallel," but not SIMD?

http://www.futuretech.blinkenlights.nl/misc/scene_1024x768.jpg



Ray Tracing

CS 148, Summer 2012
Introduction to Computer Graphics and Imaging

Justin Solomon