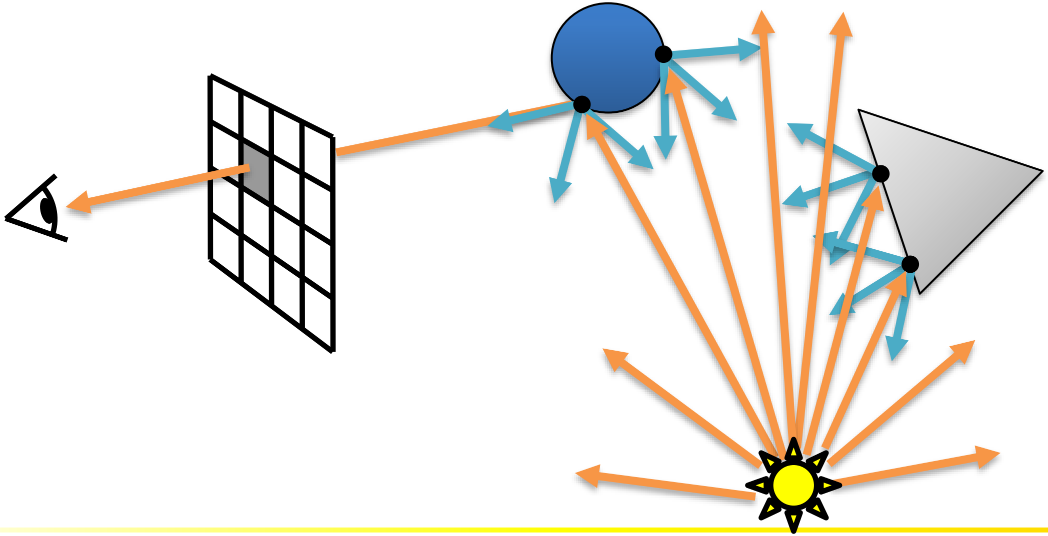
Ray Tracing

Prof. Dr. Markus Gross

ETH zürich

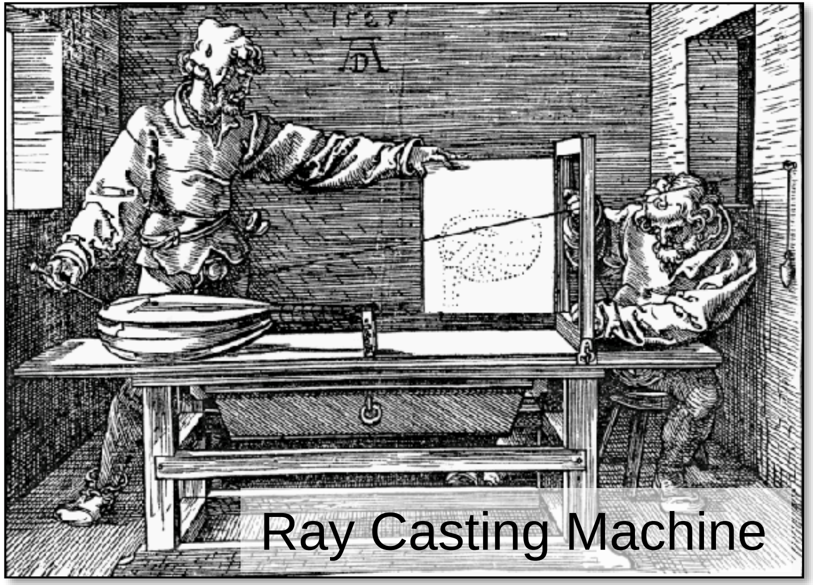
Ray Tracing

- Send rays into the scene
  to determine the color of a pixel



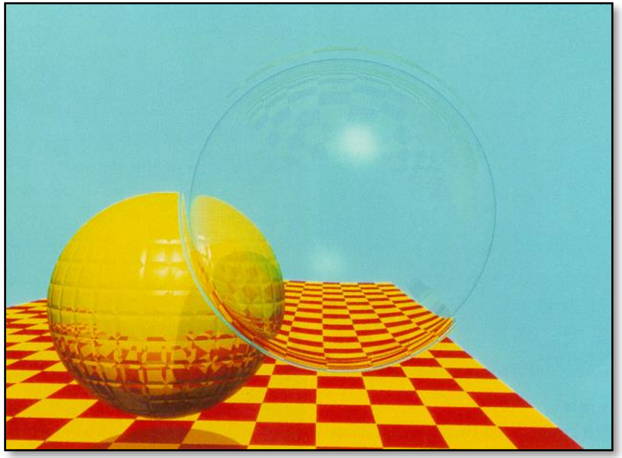
Ray Tracing

- Albrecht Dürer (1525)



Ray Tracing

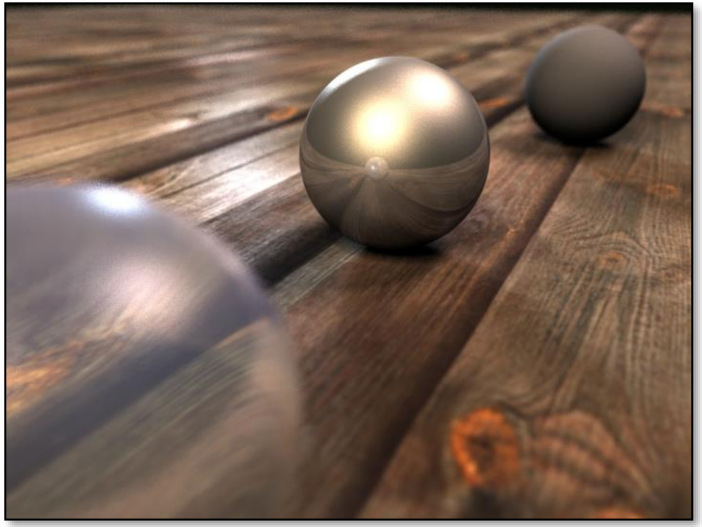
- Whitted (1979)



Recursive ray tracing (reflection & refraction)

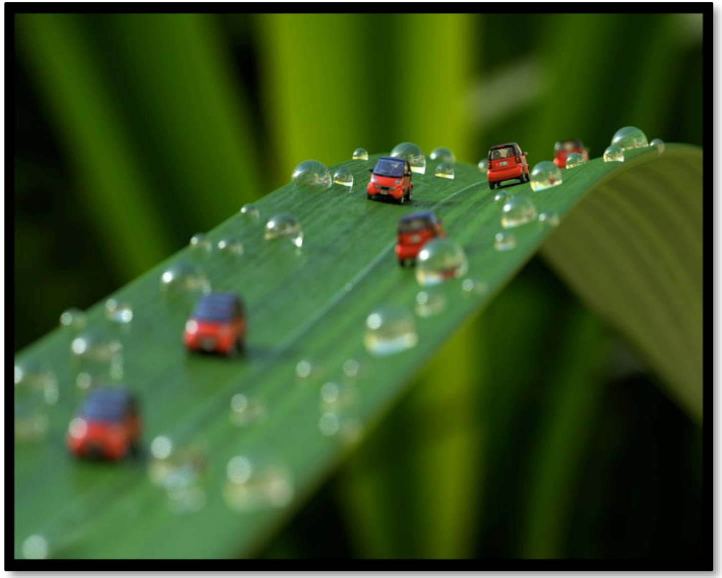
Ray Tracing

- Today – high-quality renderings



Ray Tracing

- Ray-traced images



Ray Tracing

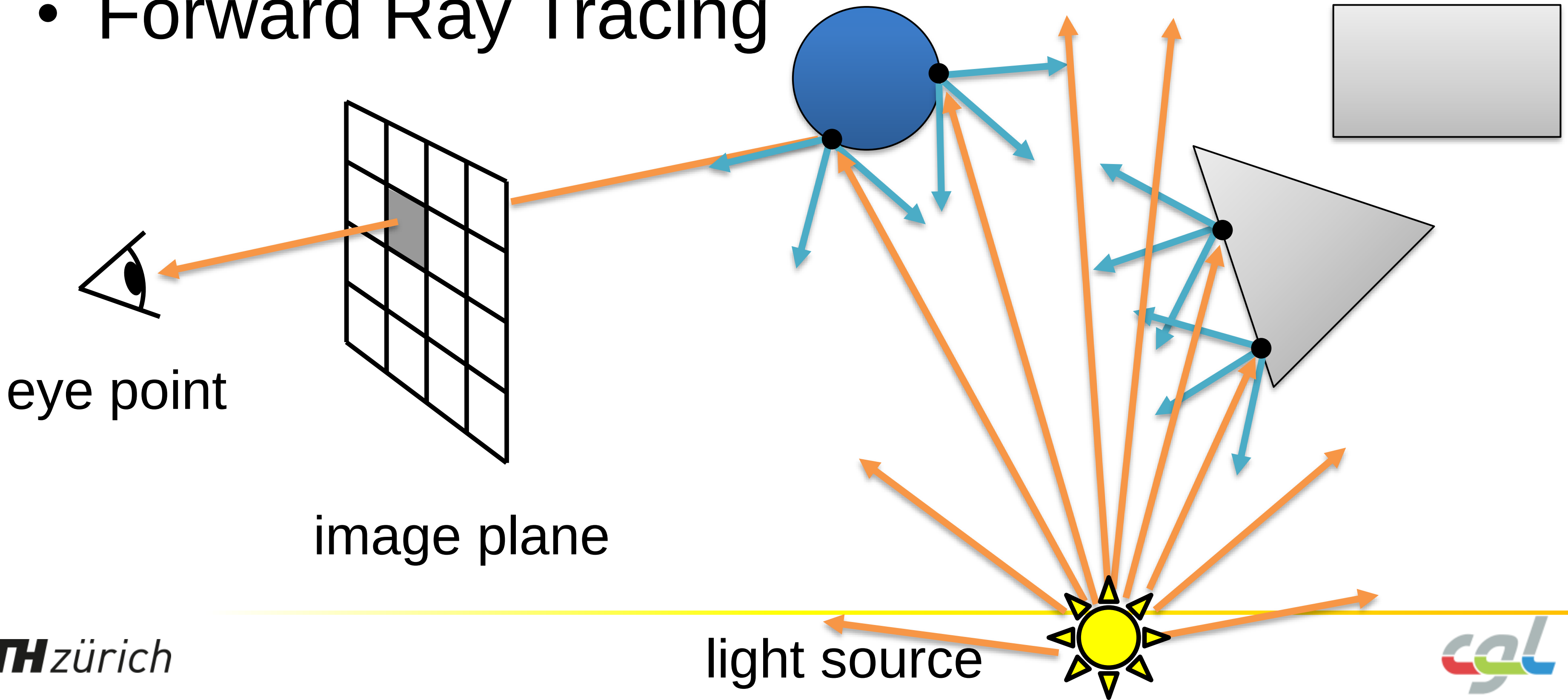
- Ray-traced images

Ray Tracing

- Ray-traced images

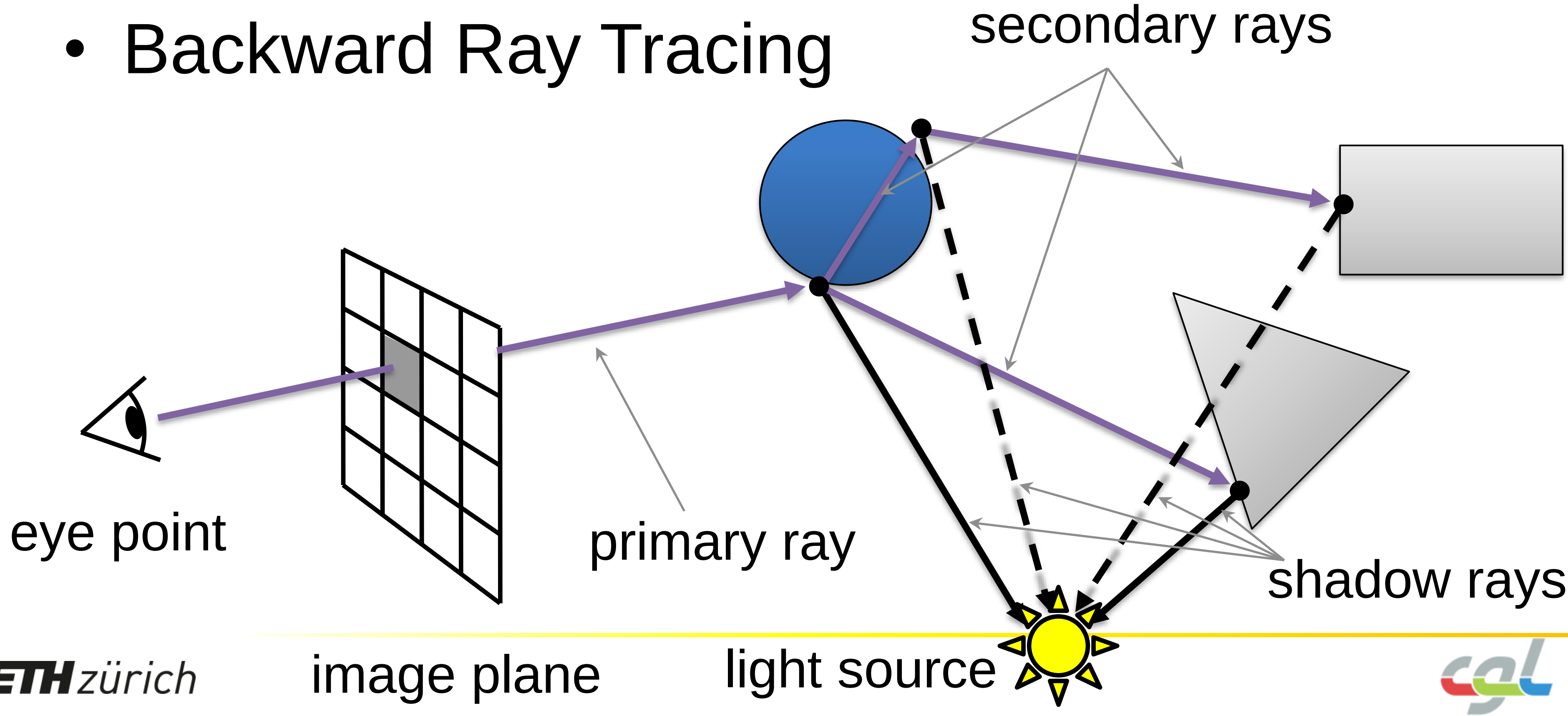
Ray Tracing

- Forward Ray Tracing



Ray Tracing

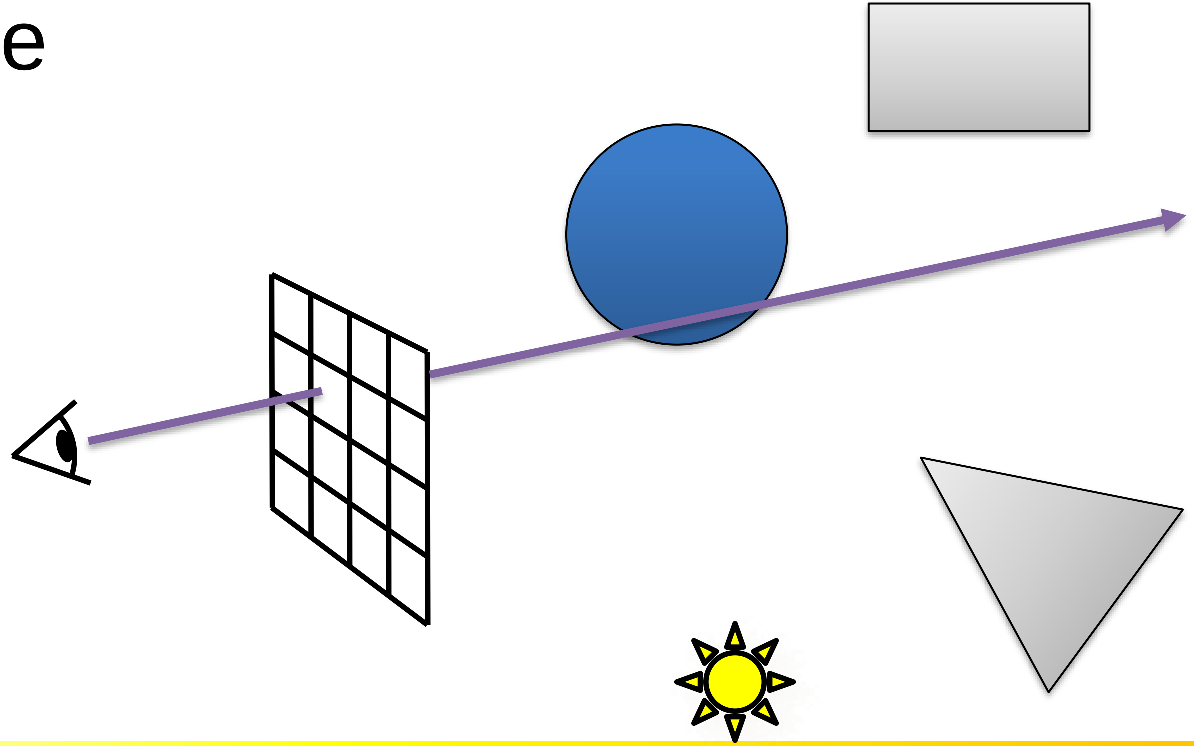
- Backward Ray Tracing



Ray Tracing

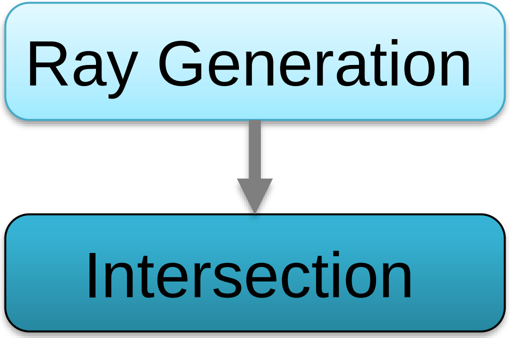
- Basic pipeline

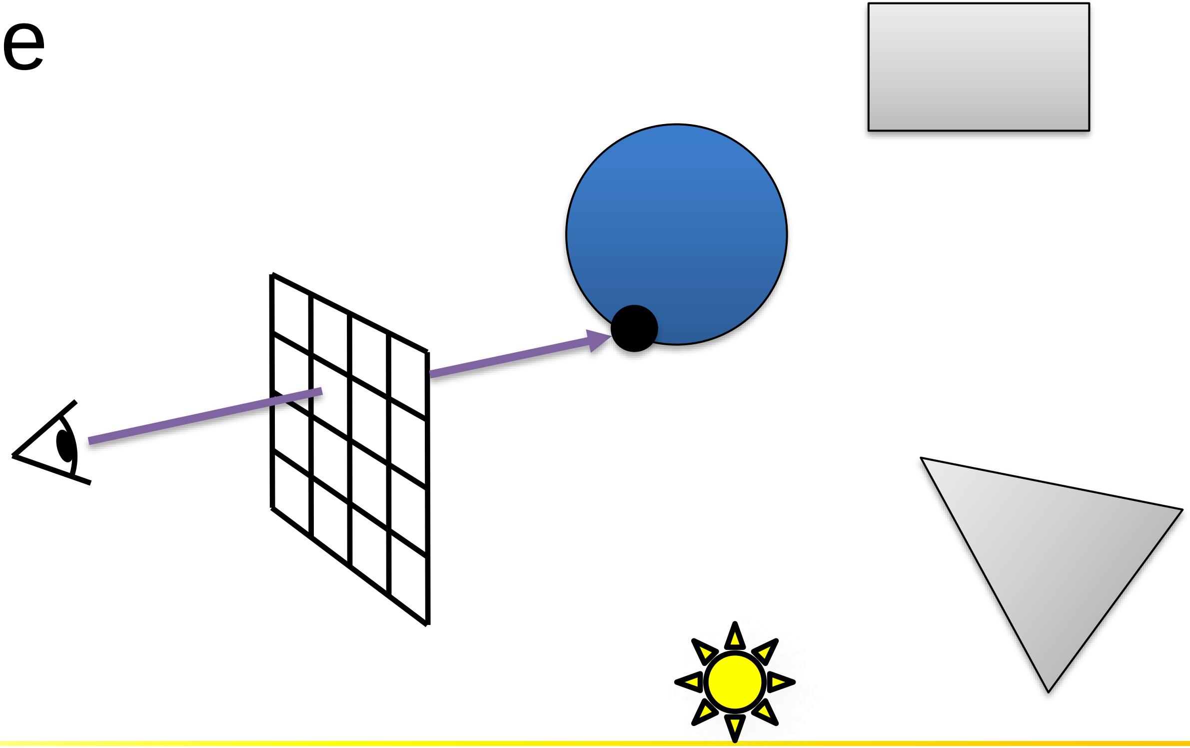
Ray Generation



Ray Tracing

- Basic pipeline





Ray Tracing

- Basic pipeline

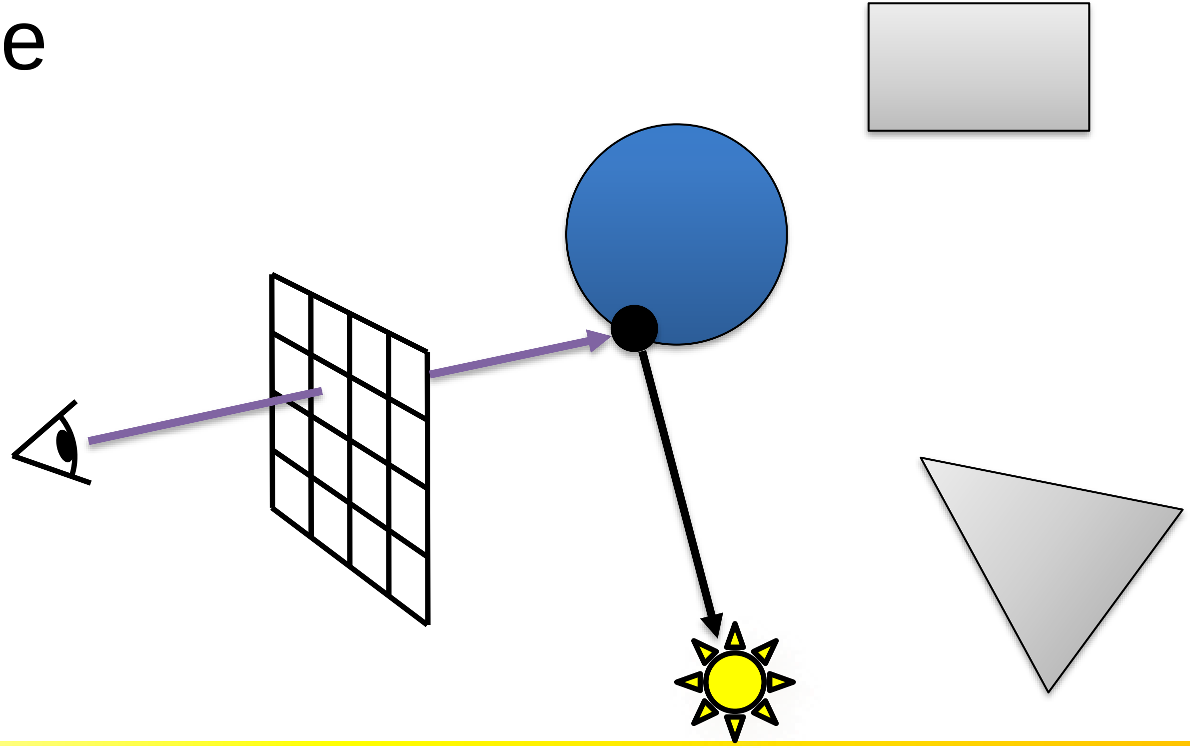
Ray Generation

↓

Intersection

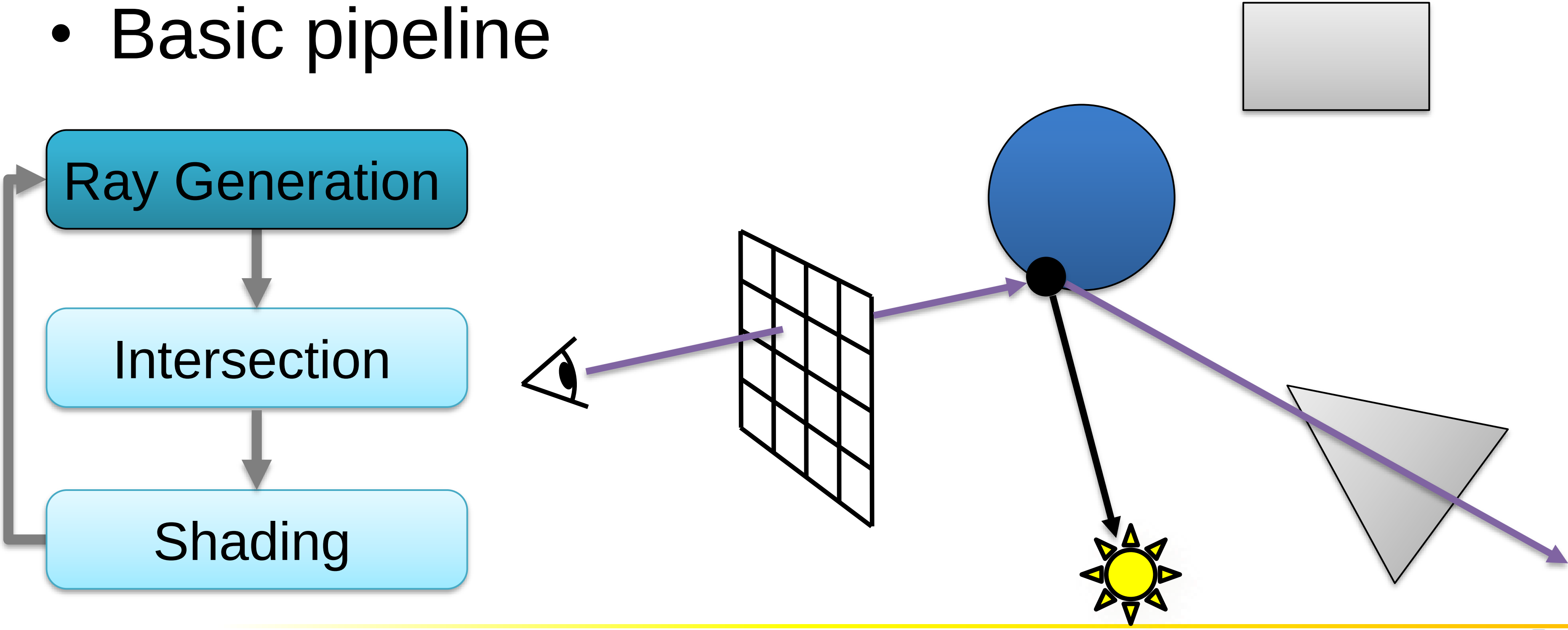
↓

Shading



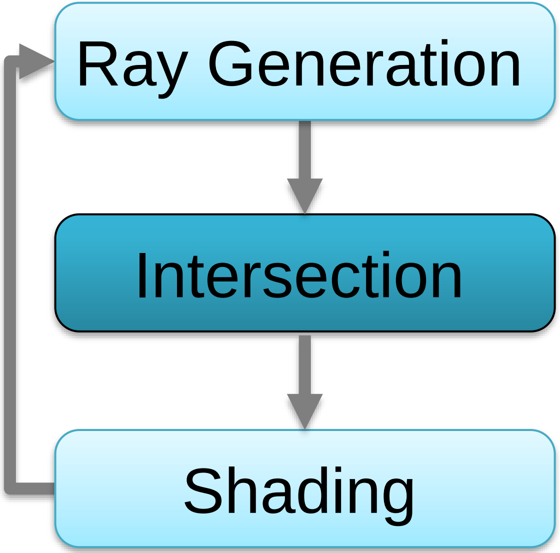
Ray Tracing

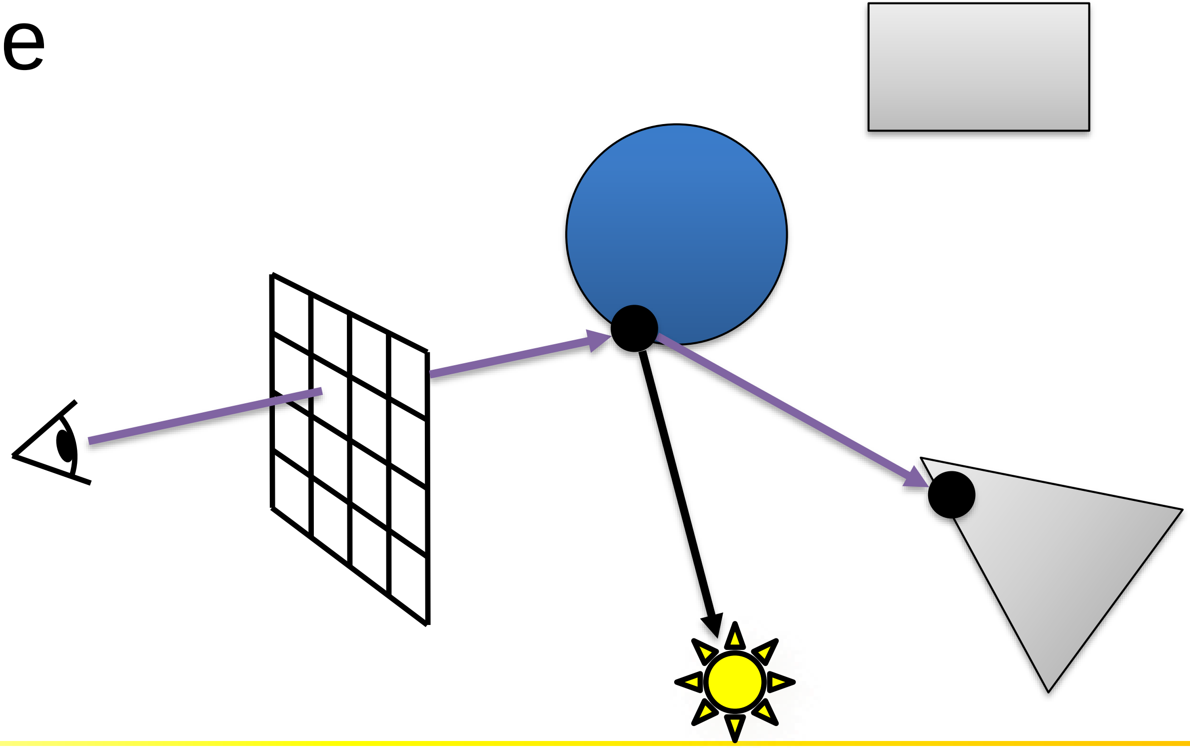
- Basic pipeline



Ray Tracing

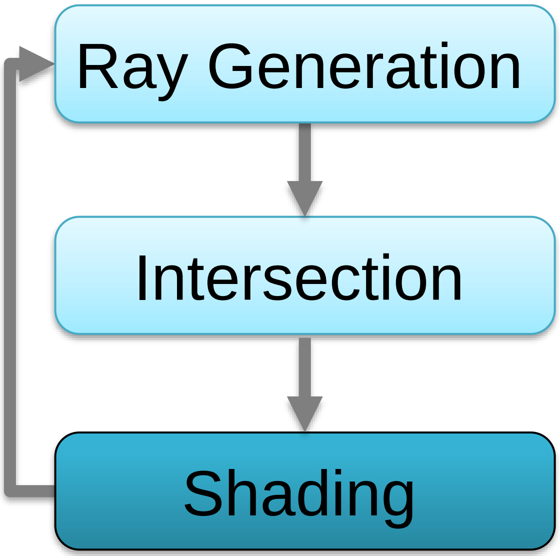
- Basic pipeline

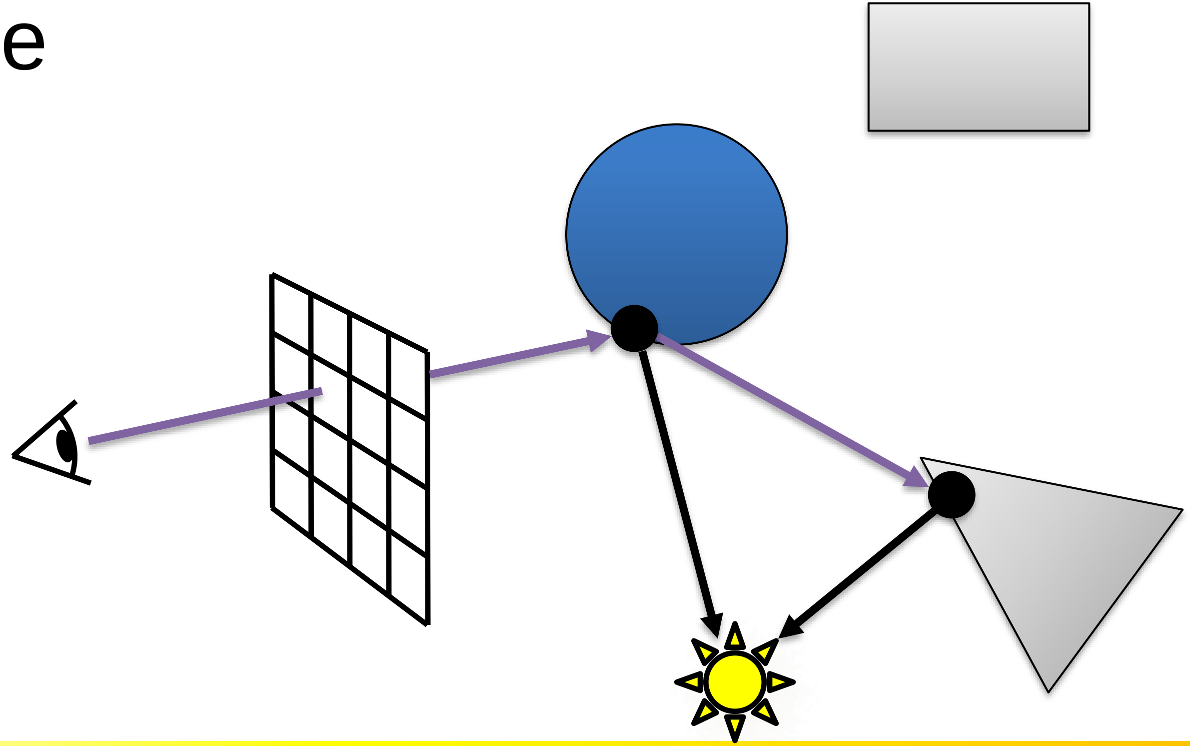




Ray Tracing

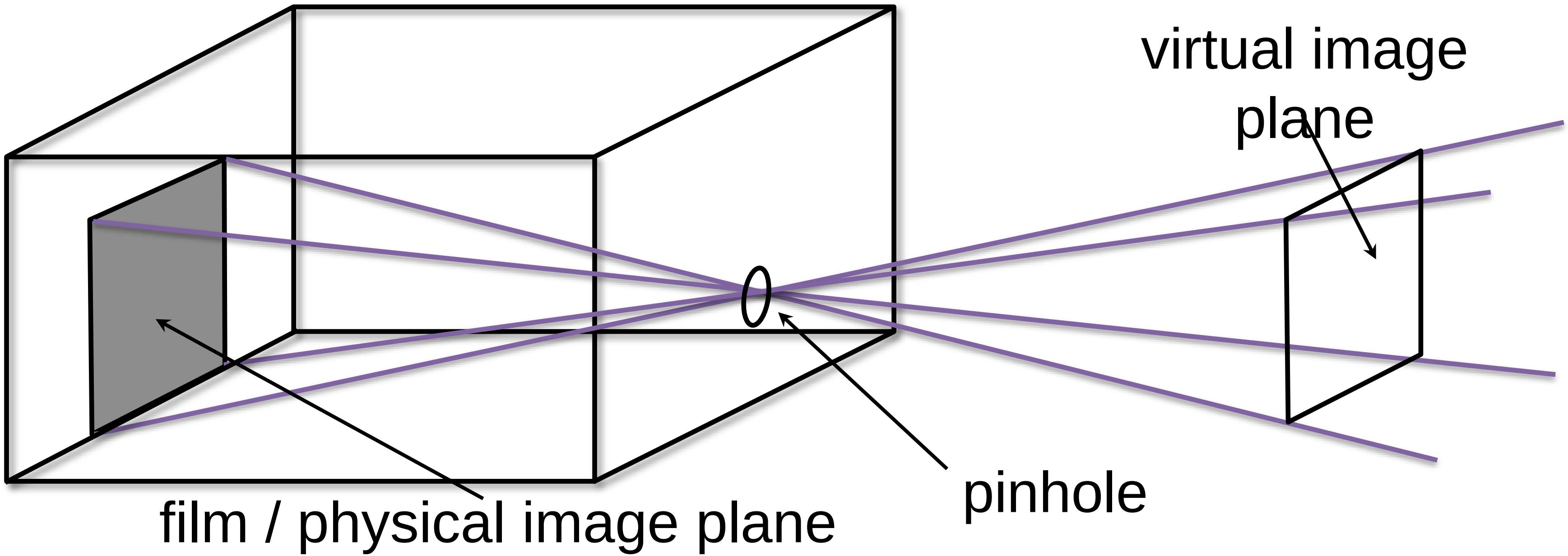
- Basic pipeline





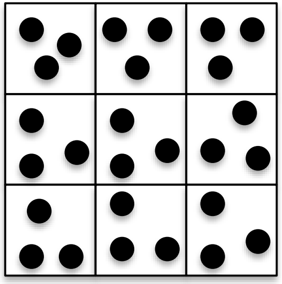
Ray Generation

- Pinhole camera



Ray Generation

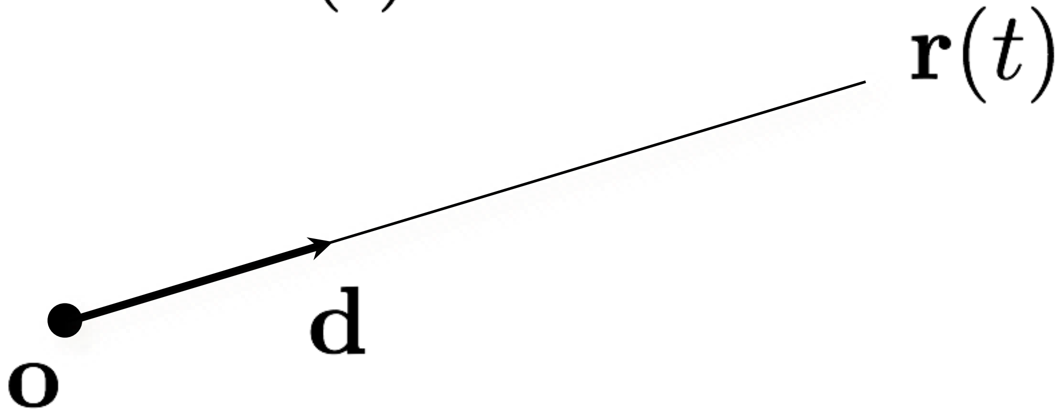
- Extension – Anti-aliasing
  - Multiple rays per pixel - supersampling



Ray-Surface Intersections

- Ray equation

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



Ray-Surface Intersections

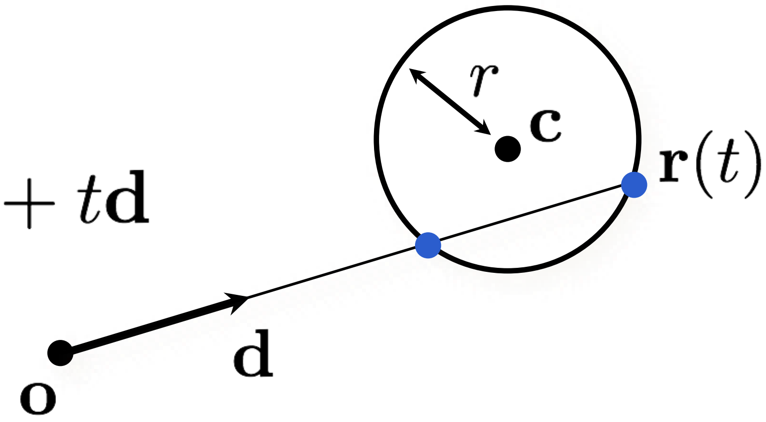
- Sphere
  - Equation

$$\|\mathbf{x} - \mathbf{c}\|^2 - r^2 = 0$$

point of interest ($\mathbf{x}$), center ($\mathbf{c}$), radius (r)

  - Insert ray equation $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$
  - Solve for t

$$\|\mathbf{o} + t\mathbf{d} - \mathbf{c}\|^2 - r^2 = 0$$

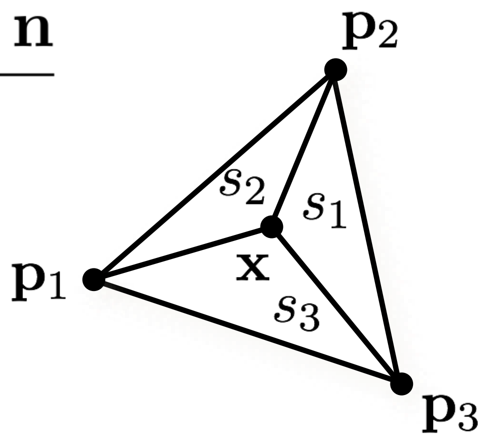


Ray-Surface Intersections

- Triangle
  - Barycentric coordinates $\mathbf{x} = s_1\mathbf{p}_1 + s_2\mathbf{p}_2 + s_3\mathbf{p}_3$
  - Intersect with triangle's plane

$$(\mathbf{o} + t\mathbf{d} - \mathbf{p_1}) \cdot \mathbf{n} = 0 \qquad t = -\frac{(\mathbf{o} - \mathbf{p_1}) \cdot \mathbf{n}}{\mathbf{d} \cdot \mathbf{n}}$$

  - where $\mathbf{n} = (\mathbf{p_2} - \mathbf{p_1}) \times (\mathbf{p_3} - \mathbf{p_1})$
  - Compute s_i
  - Test $s_1 + s_2 + s_3 = 1 \quad 0 \leq s_i \leq 1$

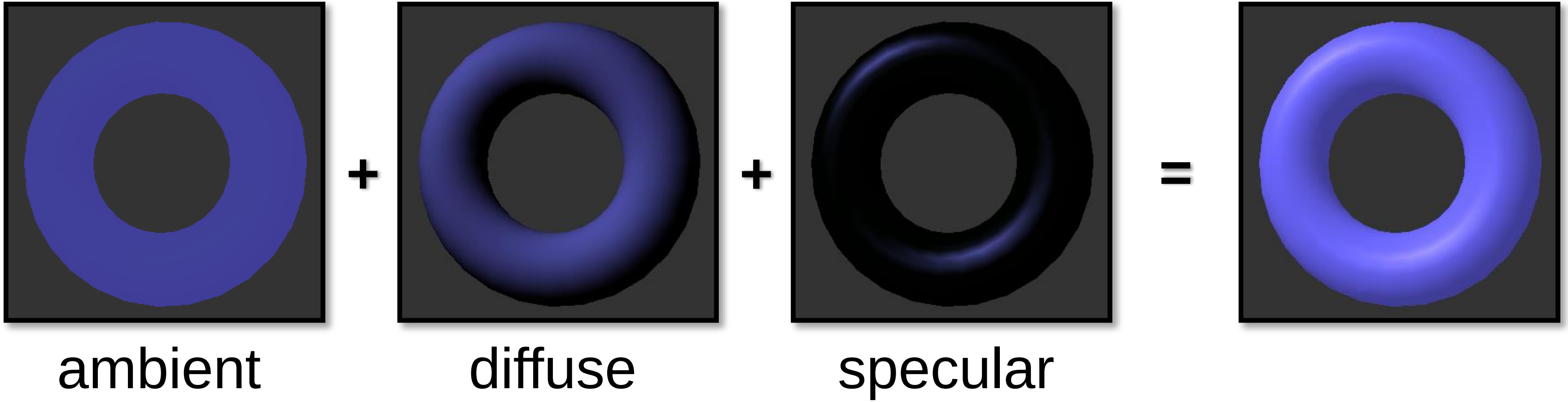


Shading

- Physically exact shading too costly
- Simplifying assumptions
  - **Surface reflectance**: diffuse, specular, ambient, transparency terms
  - **Shadows**: shadow rays to determine if a hit point is in shadow or not

Shading

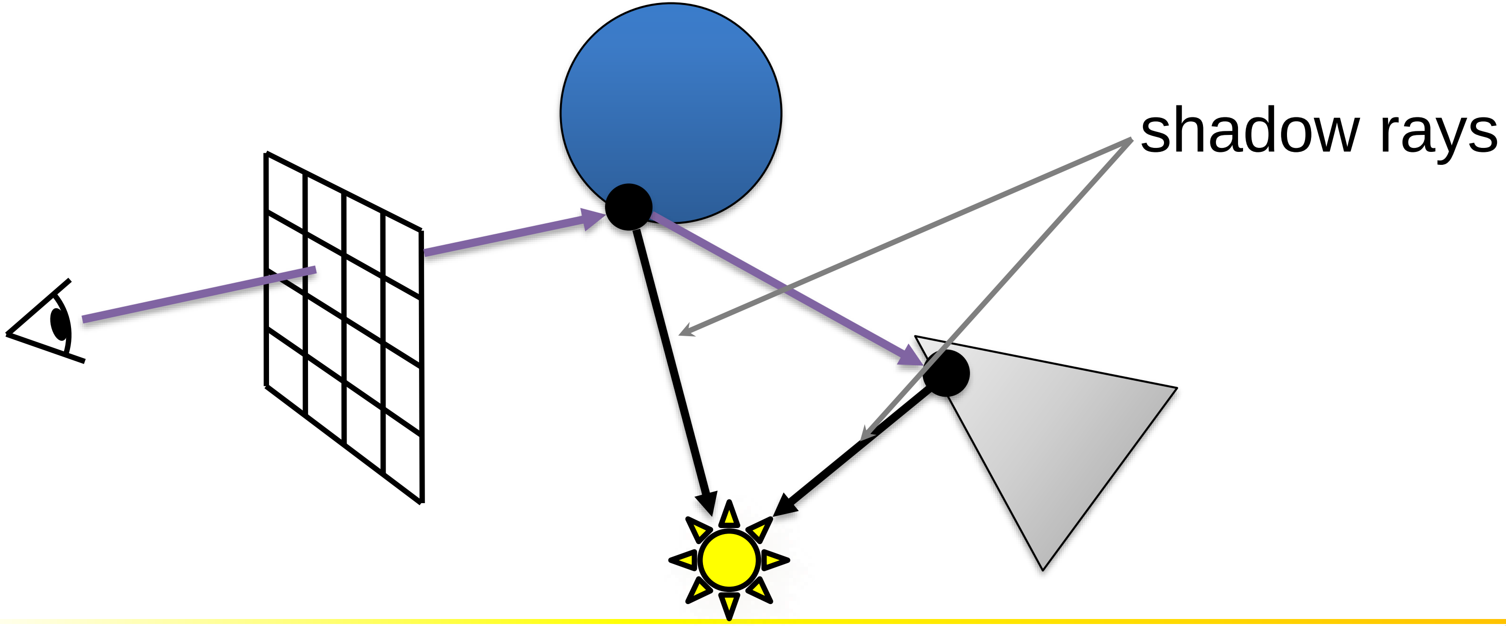
- Simplifying assumptions – surface reflectance



(see lighting and shading slides)

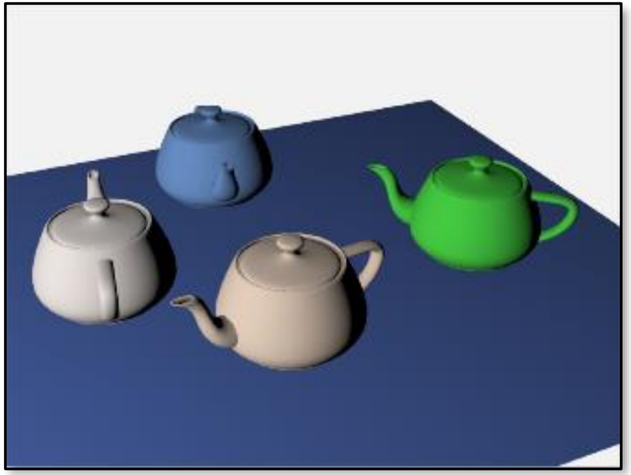
Shading

- Simplifying assumptions – shadows

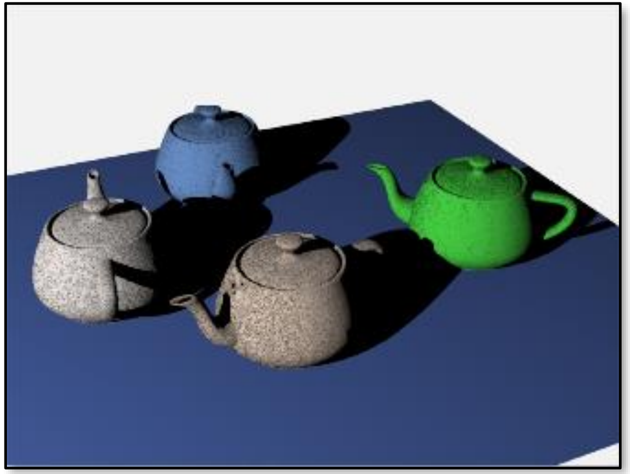


Shading

- Simplifying assumptions – shadows



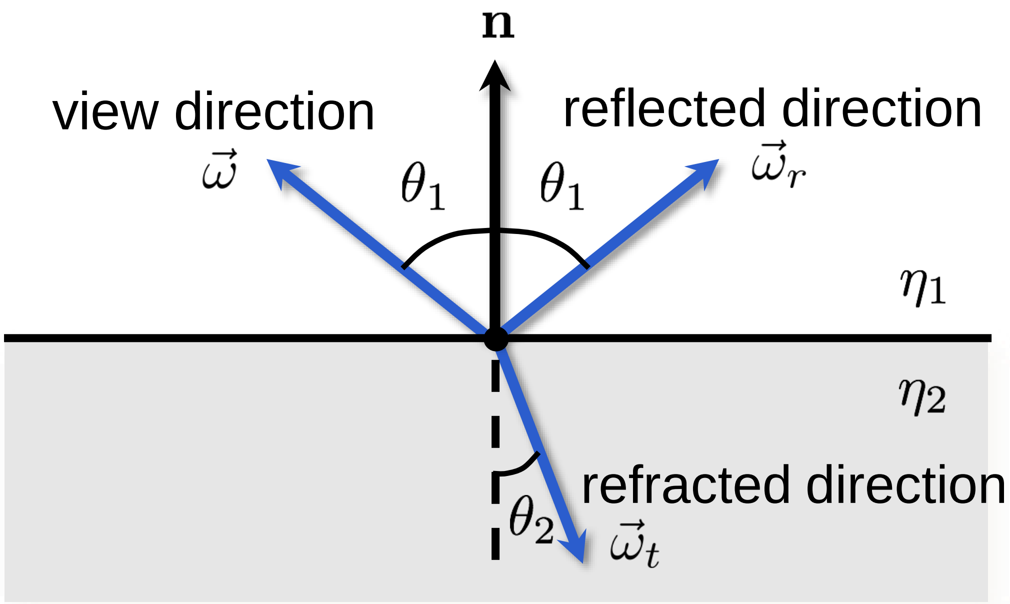
Diffuse shading



With shadows

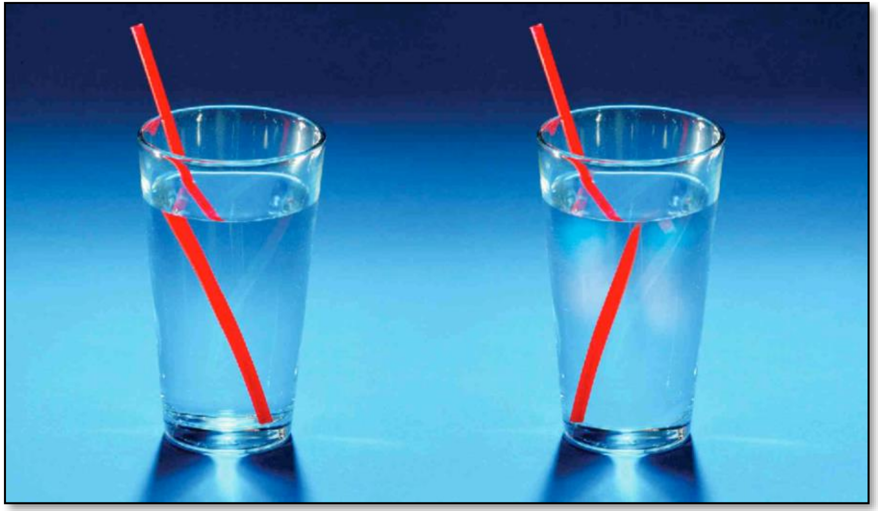
Shading

- Extensions – Refraction



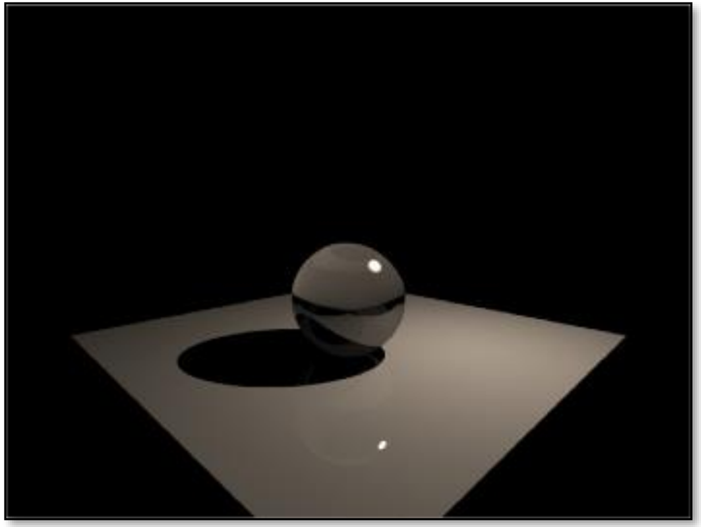
Shading

- Extensions – Refraction

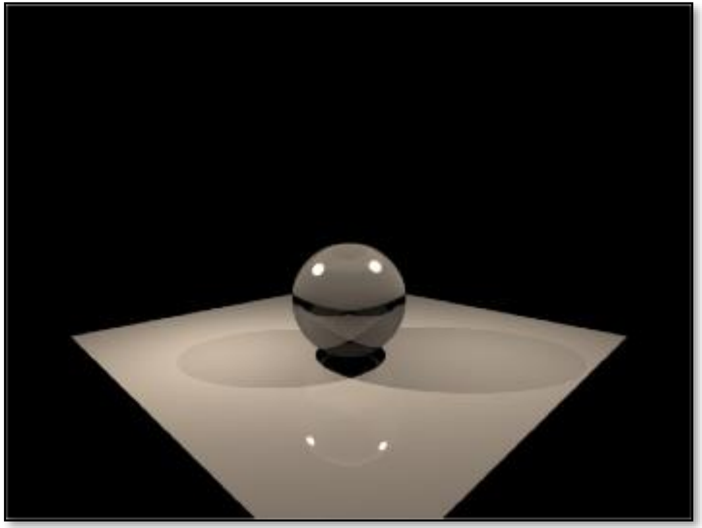


Shading

- Extensions – Multiple lights



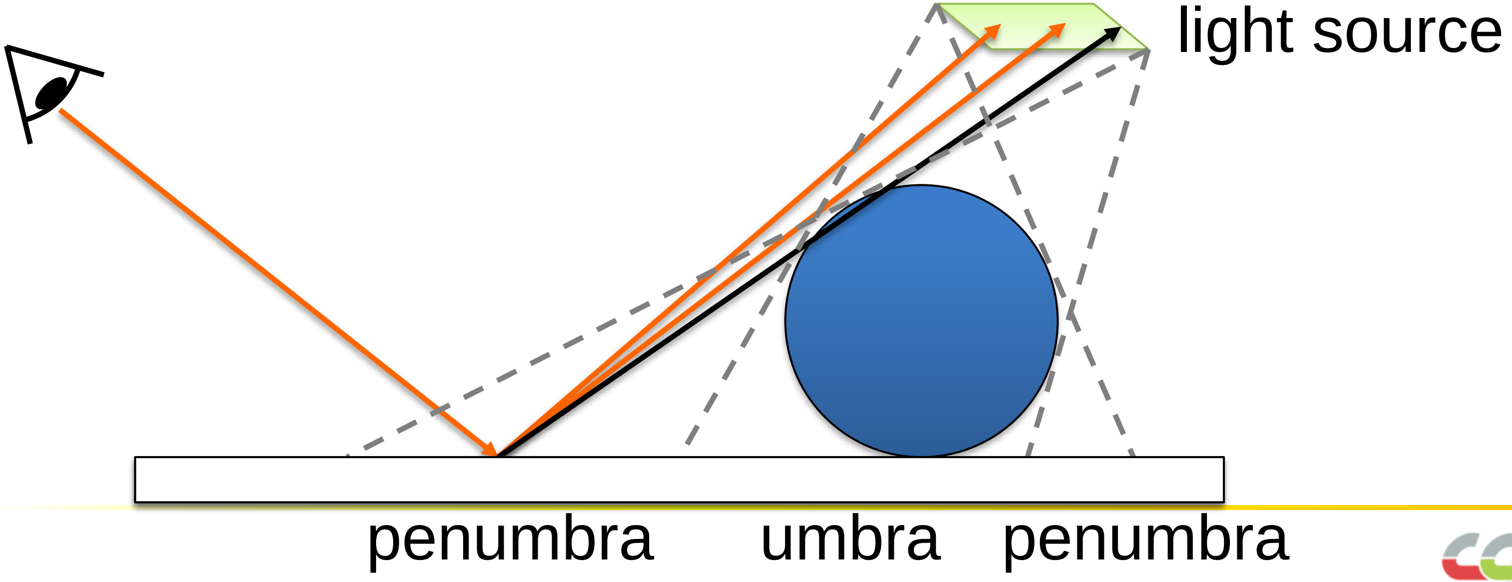
One light



Two lights

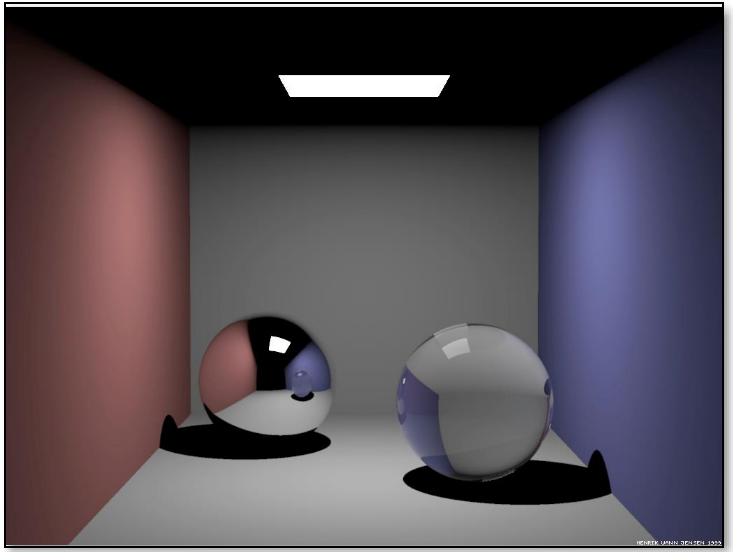
Shading

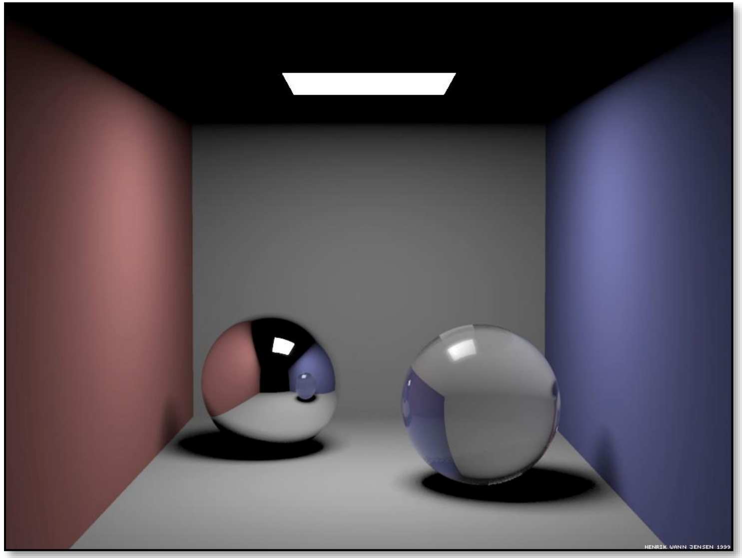
- Extensions – Area lights for soft shadows
  - Multiple shadow rays to sample area light source



Shading

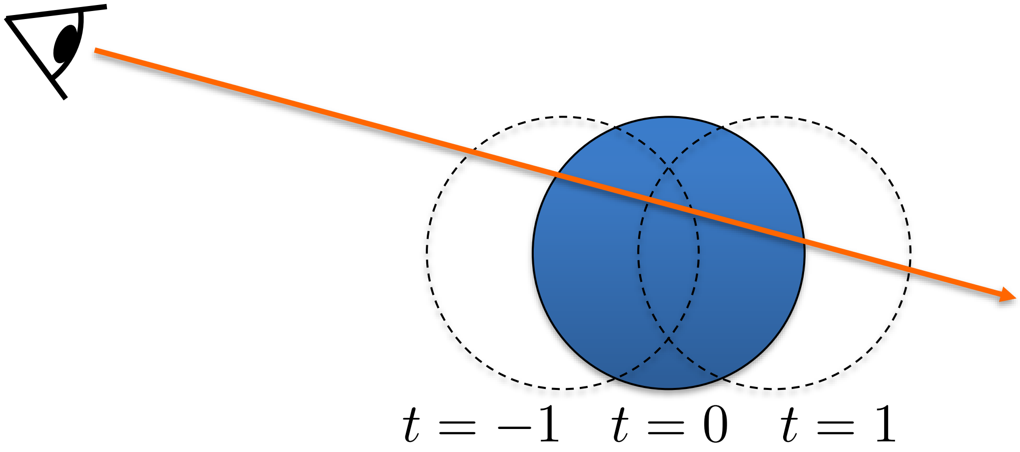
- Extensions – Area lights for soft shadows





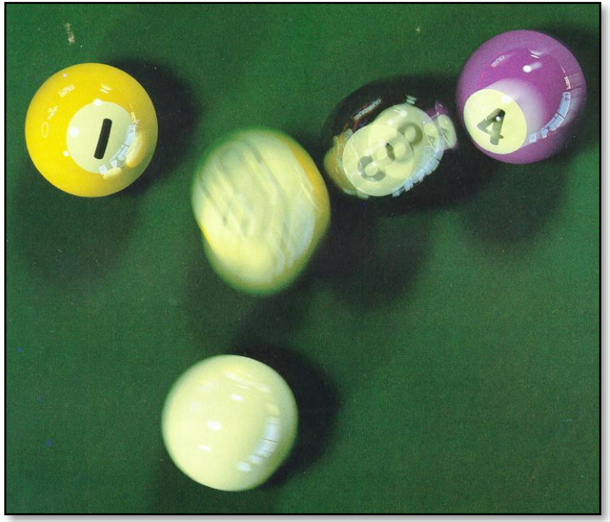
Shading

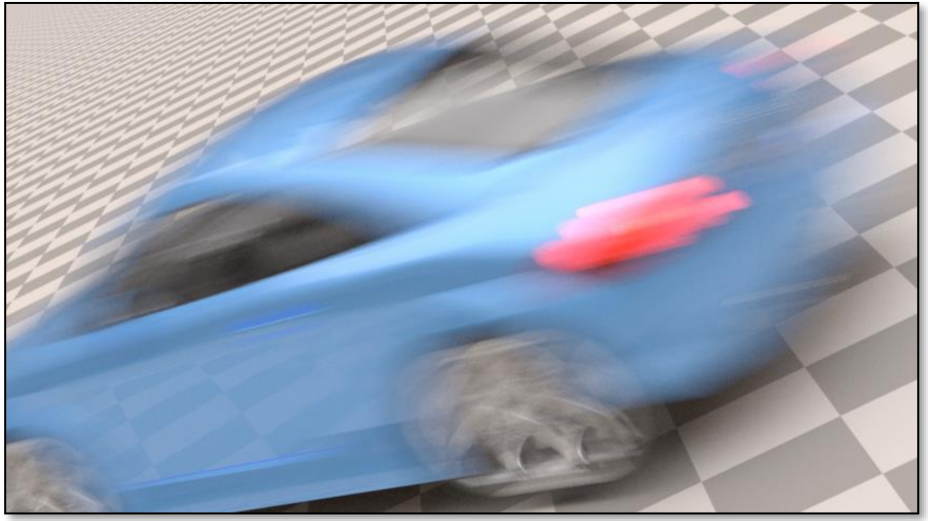
- Extensions – Motion blur
  - Sample objects and intersect in time



Shading

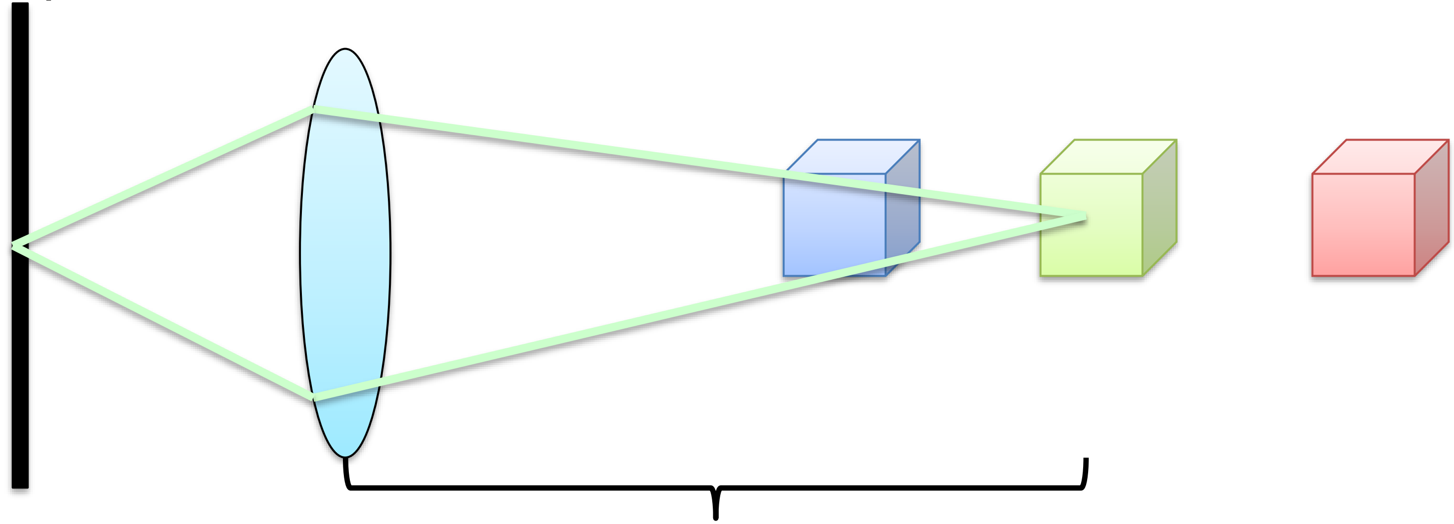
- Extensions – Motion blur





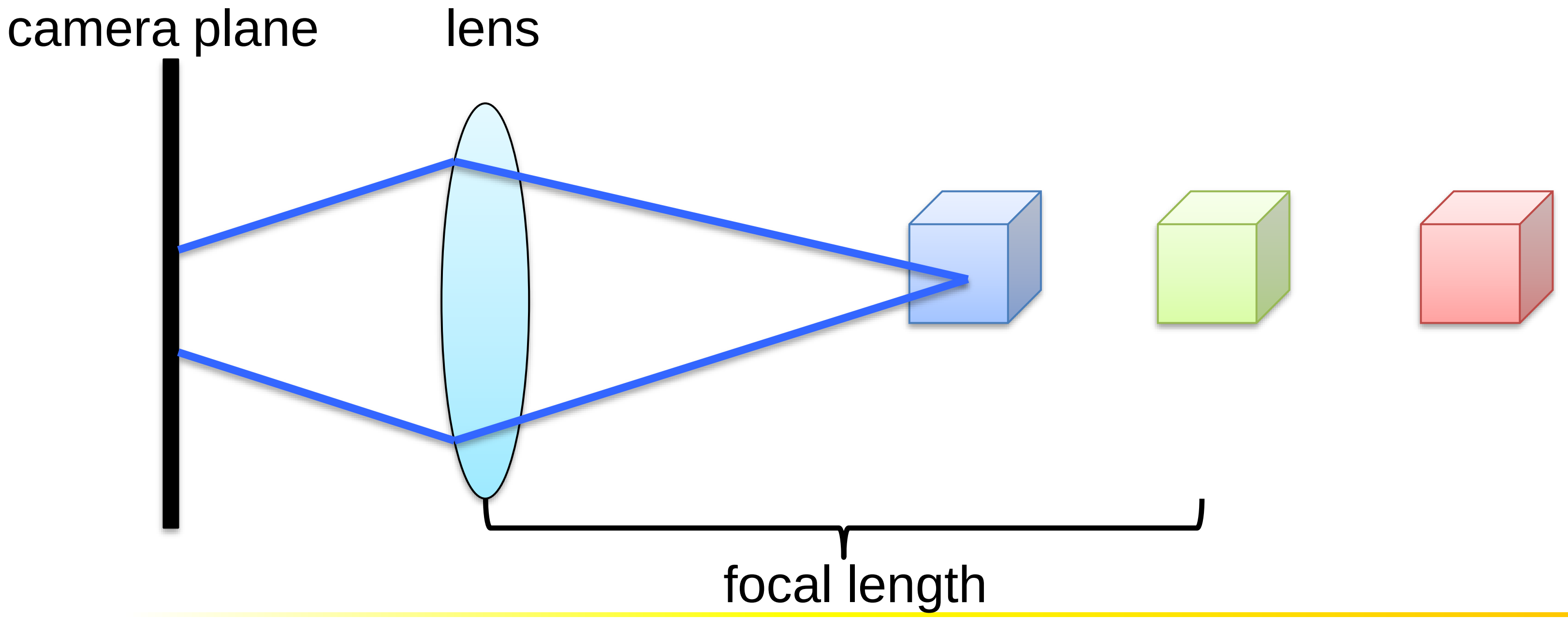
Shading

- Extensions – Depth of field



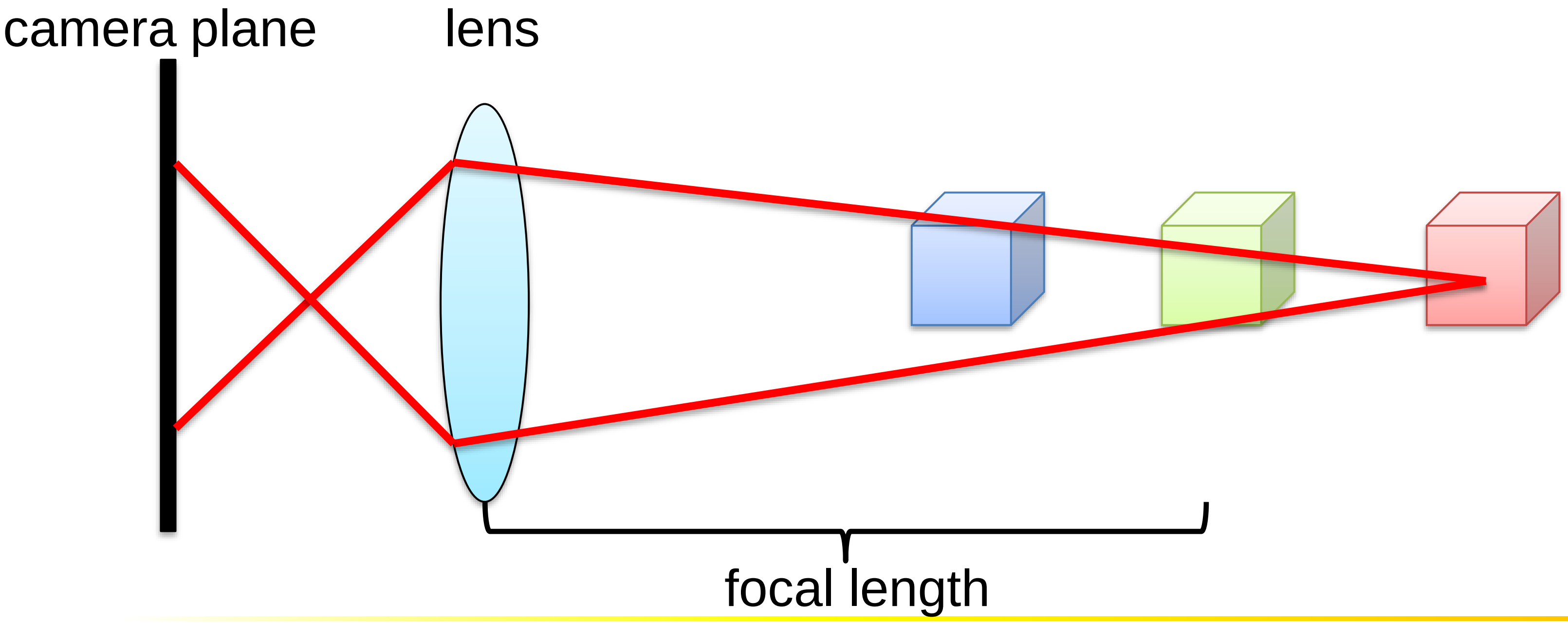
Shading

- Extensions – Depth of field



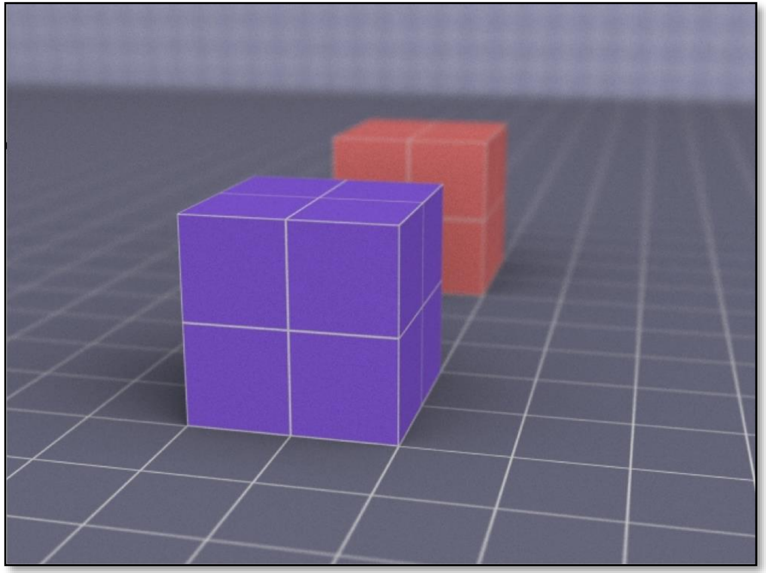
Shading

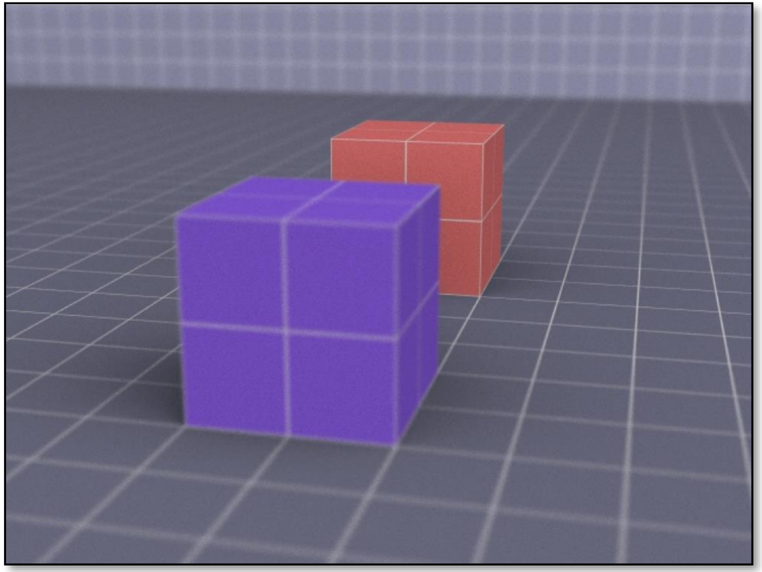
- Extensions – Depth of field



Shading

- Extensions – Depth of field





Acceleration

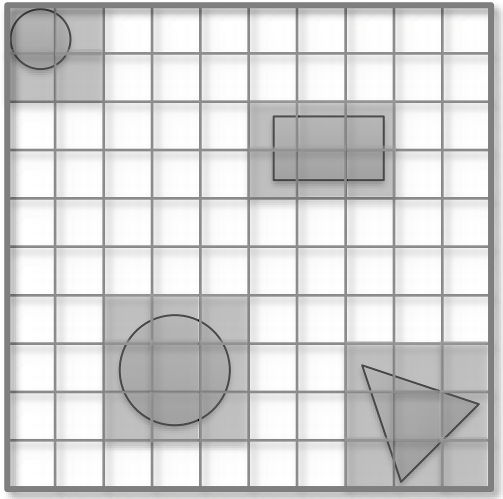
- Complex scenes
  - Cost = *O(#rays* x *#objects)*
  - 50K trees each with 1M polygons = 50B polygons
    → **594 years!**

Acceleration

- Solution: fewer intersections
  - Uniform grids
  - Space partitioning trees
- 11 minutes – 300,000,000x speedup

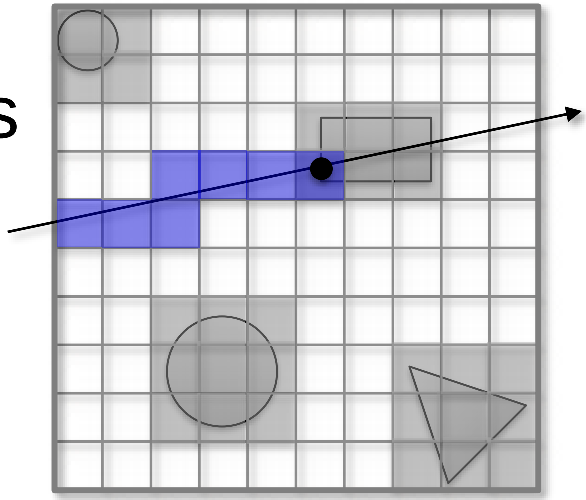
Acceleration

- Uniform grids – Preprocessing
  - Compute bounding box
  - Set grid resolution
  - Rasterize objects
  - Store references to objects



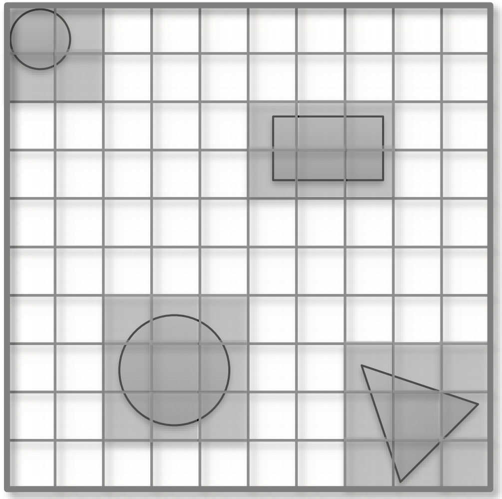
Acceleration

- Uniform grids – Traversal
  - Incrementally rasterize ray
  - Stop when intersection happens



Acceleration

- Uniform grids
- Advantages
  - Fast to build
  - Easy to code
- Disadvantages
  - Non-adaptive to scene geometry



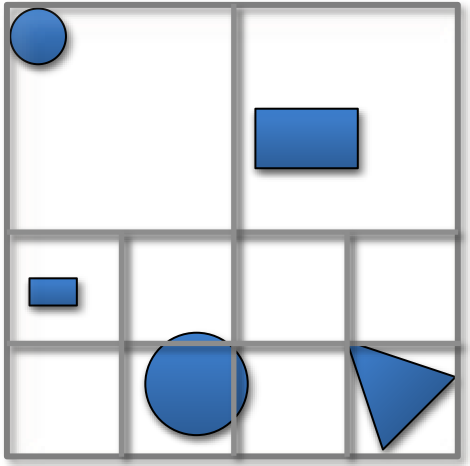
Acceleration

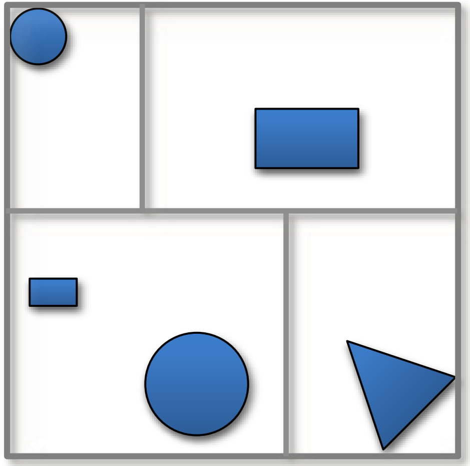
- Uniform grids

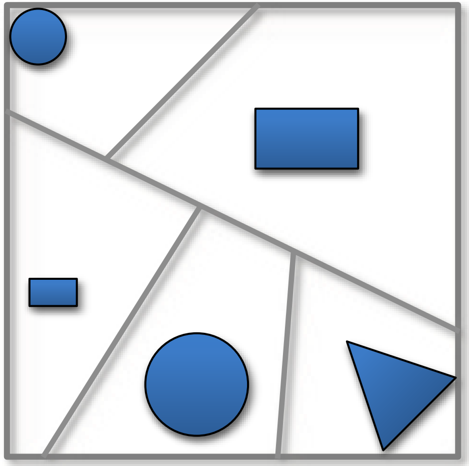
Brute force: 6321 intersection tests per ray
Uniform grid: 44.86 intersection tests per ray

Acceleration

- Space partitioning trees


octree


kd-tree


bsp-tree