Ray-tracing
Ray-tracing

- Ray casting, ray tracing: rays sent from viewpoint towards the scene.
- One ray for every pixel.
- Pixel color depends on illumination at the first surface intersected by the ray.
- Using local illumination models.
Three new rays are generated: refracted ray, reflected ray, shadow ray.
Ray–tree

Viewpoint

Object 1

- reflected
- refracted

Object 2

- reflected
- refracted

Object 3

- reflected
- refracted

Light source
Even more rays

- Soft shadows
  - Several shadow rays for each extended light source
Even more rays

- **Soft shadows**
  - Several shadow rays for each extended light source
- **Anti-aliasing**
  - Several rays per pixel

1 rayon  
2 rayons  
3 rayons
Even more rays

- Soft shadows
  - Several shadow rays for each extended light source
- Anti-aliasing
  - Several rays per pixel
- Glossy reflections
  - Several reflected rays
Even more rays

- Soft shadows
  - Several shadow rays for each extended light source
- Anti-aliasing
  - Several rays per pixel
- Glossy reflections
  - Several reflected rays
- Motion blur
  - Several rays through time
Even more rays

- Soft shadows
  - Several shadow rays for each extended light source
- Anti-aliasing
  - Several rays per pixel
- Glossy reflections
  - Several reflected rays
- Motion blur
  - Several rays through time
- Depth of field
  - Several rays per pixel through the lens
Even more rays

- Soft shadows
  - Several shadow rays for each extended light source
- Anti-aliasing
  - Several rays per pixel
- Glossy reflections
  - Several reflected rays
- Motion blur
  - Several rays through time
- Depth of field
  - Several rays per pixel through the lens
Ray–scene intersection

- Ray–plane: line–plane intersection
- Ray–polygon:
  - line–plane intersection.
  - test whether intersection point is in polygon:
    - project onto xy plane, check inside 2D polygon.
Ray–scene intersection

- 99% of the time is spent doing intersections.
- Need for accelerations:
  - bounding volumes,
  - uniform grids (voxels),
  - octrees,
  - BSP–trees,
  - problem specific accelerations;
Bounding volumes

- Intersection with a bounding volume
- Early rejection
Bounding volumes

Not-axis-aligned bounding box

Bounding sphere

Arbitrary convex region

Axis-aligned bounding box
BVH: Bounding Volume Hierarchy
Uniform grid
Adaptive grid: Octree
Question. 3 mn with your neighbors

- Compare 3 accelerations structures:
  - Bounding volumes
  - Uniform grid
  - Octree
Comparison

- **Bounding volume:**
  - long initial step, fast requests.

- **Uniform Grid:**
  - fast initial step, fast requests... if proper resolution.

- **Octrees:**
  - fast and simple initial step, longer requests.
Ray-tracing: advantages

- Slow, but no extra charge for:
  - hidden surface removal,
  - shadows,
  - transparency,
  - texture-mapping (including procedural).
- Inter-reflexions between objects,
- Any graphics primitives,
- Global illumination model.
Ray-tracing: issues

- Limited to Snell–Descartes:
  - all objects are metallic.

- Tree limited to a certain depth:
  - complex objects may be a problem (diamonds, cristal glass)

- Extension: Monte–Carlo Ray–Tracing
  - shoots several rays. slow, but nice.