

CS-C3100 Computer Graphics

Part 2 – Rendering

11.1 Introduction to Rendering



TAAV®
TALLER DE ARQUITECTURA VIRTUAL

Jaakko Lehtinen
with lots of slides from Frédo Durand

In This Video

- Overview of remaining course content
- What does *rendering* mean?
 1. Figure out what's visible in each pixel
 2. Compute what color those things appear to be
- Main approaches: *ray tracing* and *rasterization*

The Story So Far

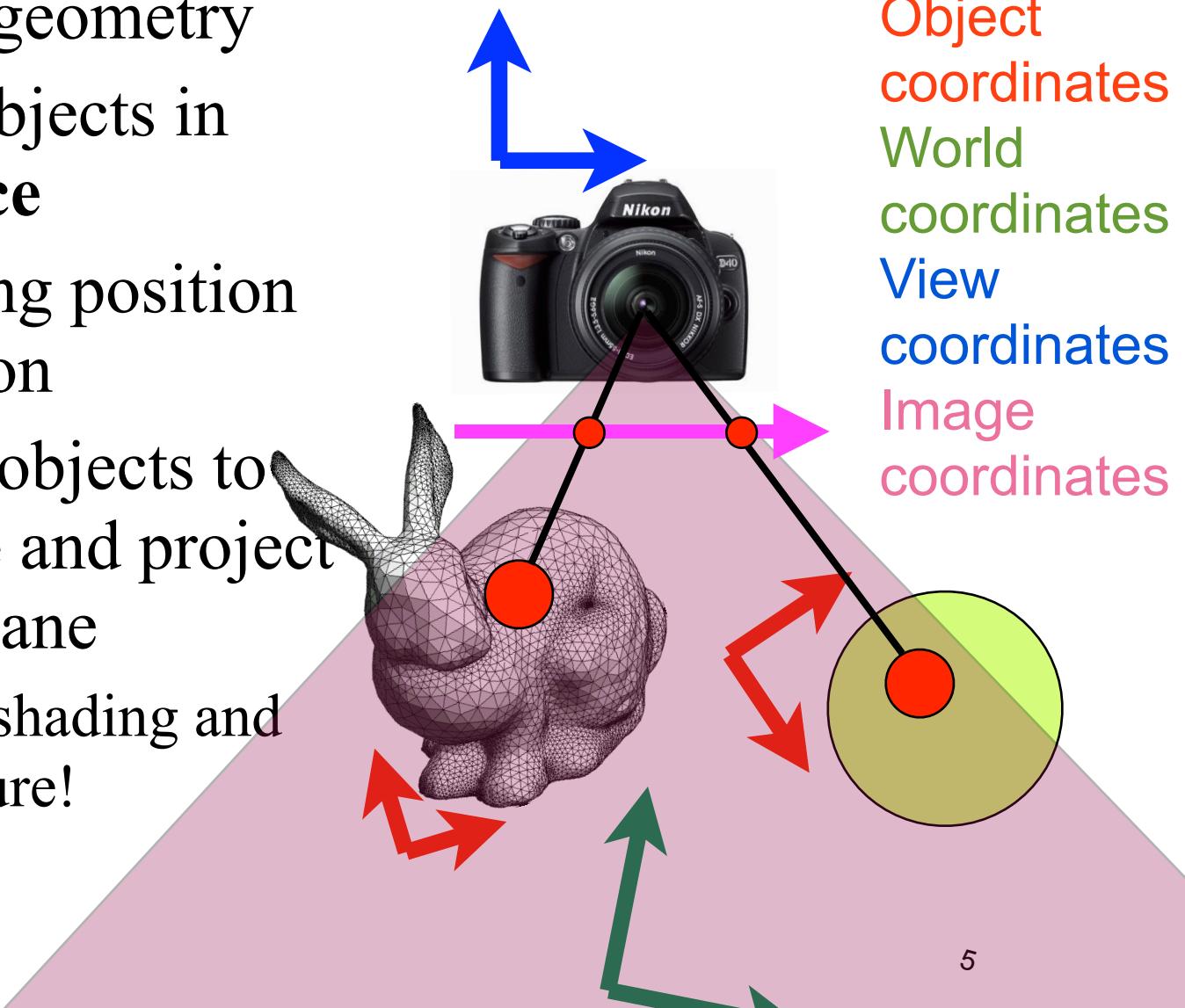
- Modeling
 - splines, hierarchies, transformations, meshes, etc.
- Animation
 - quaternions, skinning, ODEs, masses and springs
- **Now we'll see how to generate an image given a scene description!**

The Remainder of the Term

- Ray Tracing & its applications
- Rasterization, z-buffering
- Sampling and antialiasing
- Shading, texture mapping
 - What makes materials look like they do?
- Shadow techniques
- Intro to Global Illumination
 - Monte Carlo techniques, photon mapping, etc.

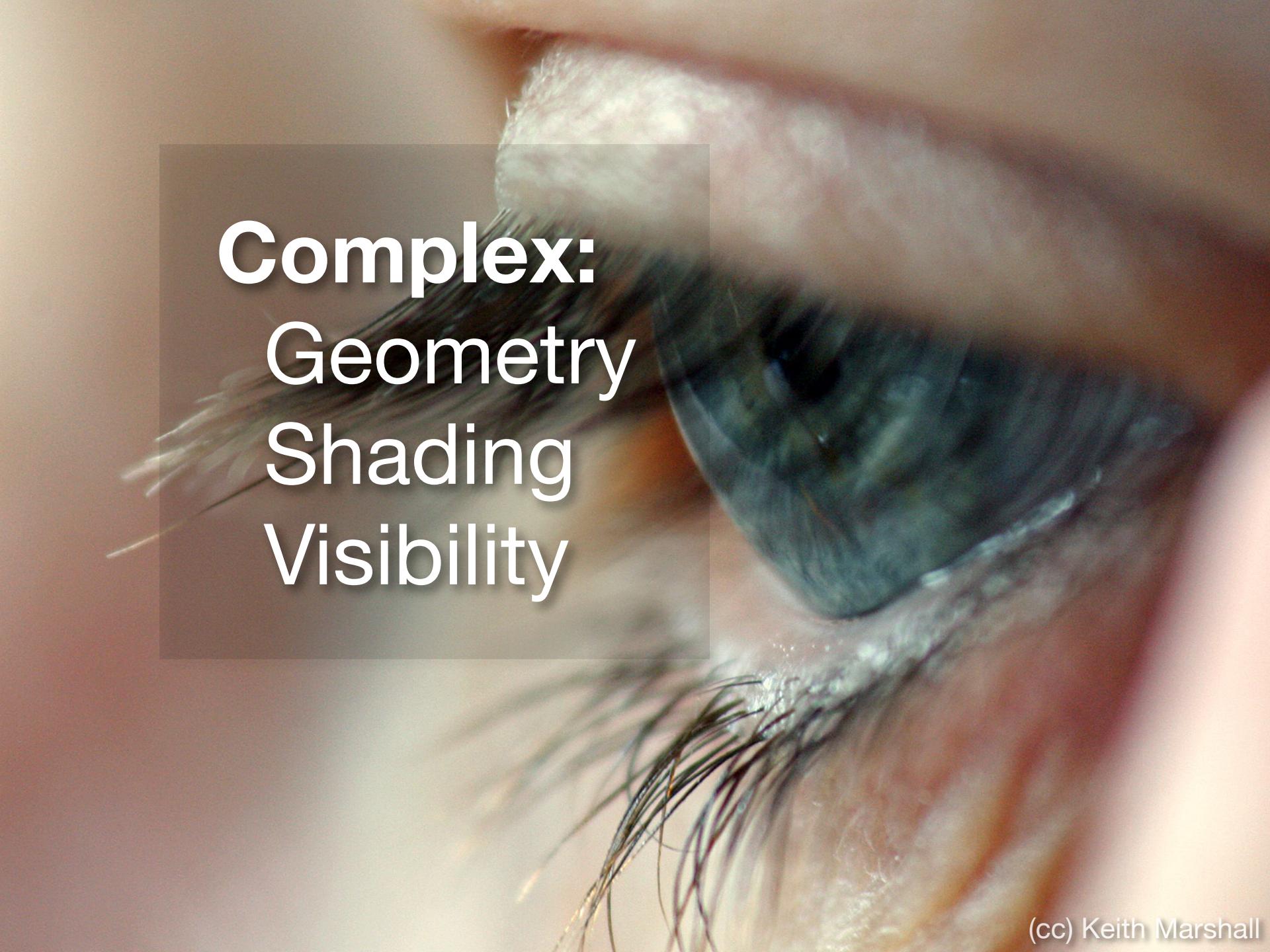
Modeling/Viewing Pipeline

1. Model the geometry
2. Place the objects in **world space**
3. Pick viewing position and direction
4. Transform objects to **view space** and project to image plane
 - Compute shading and draw picture!





(cc) Keith Marshall

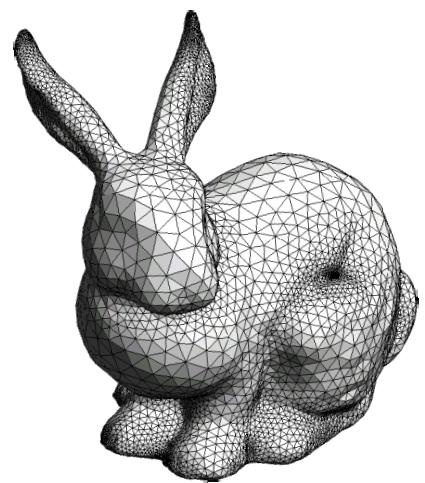


Complex: Geometry Shading Visibility

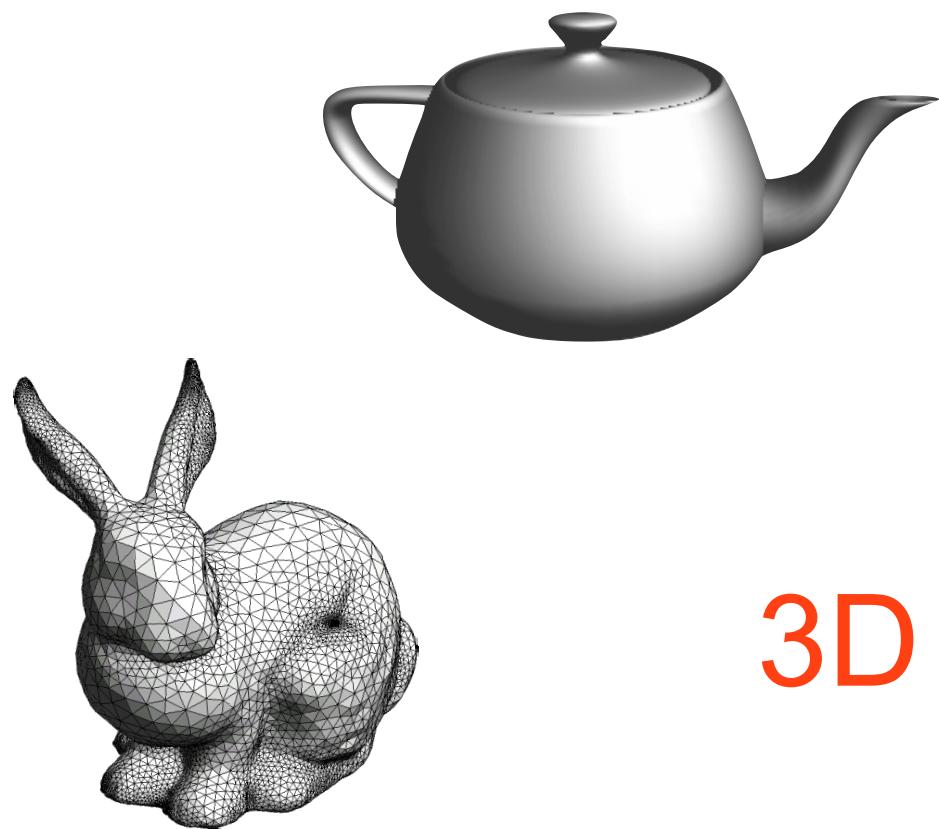
Rendering:

Compute what's visible.

Compute what color it is.



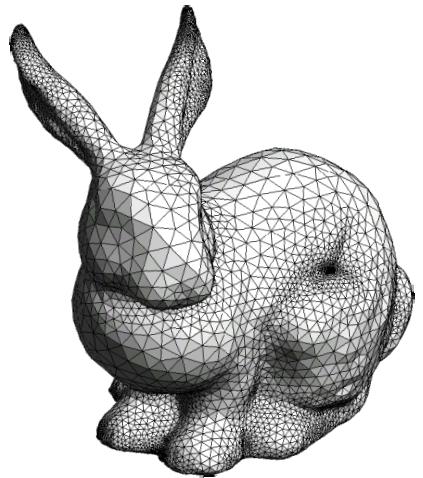
Scene



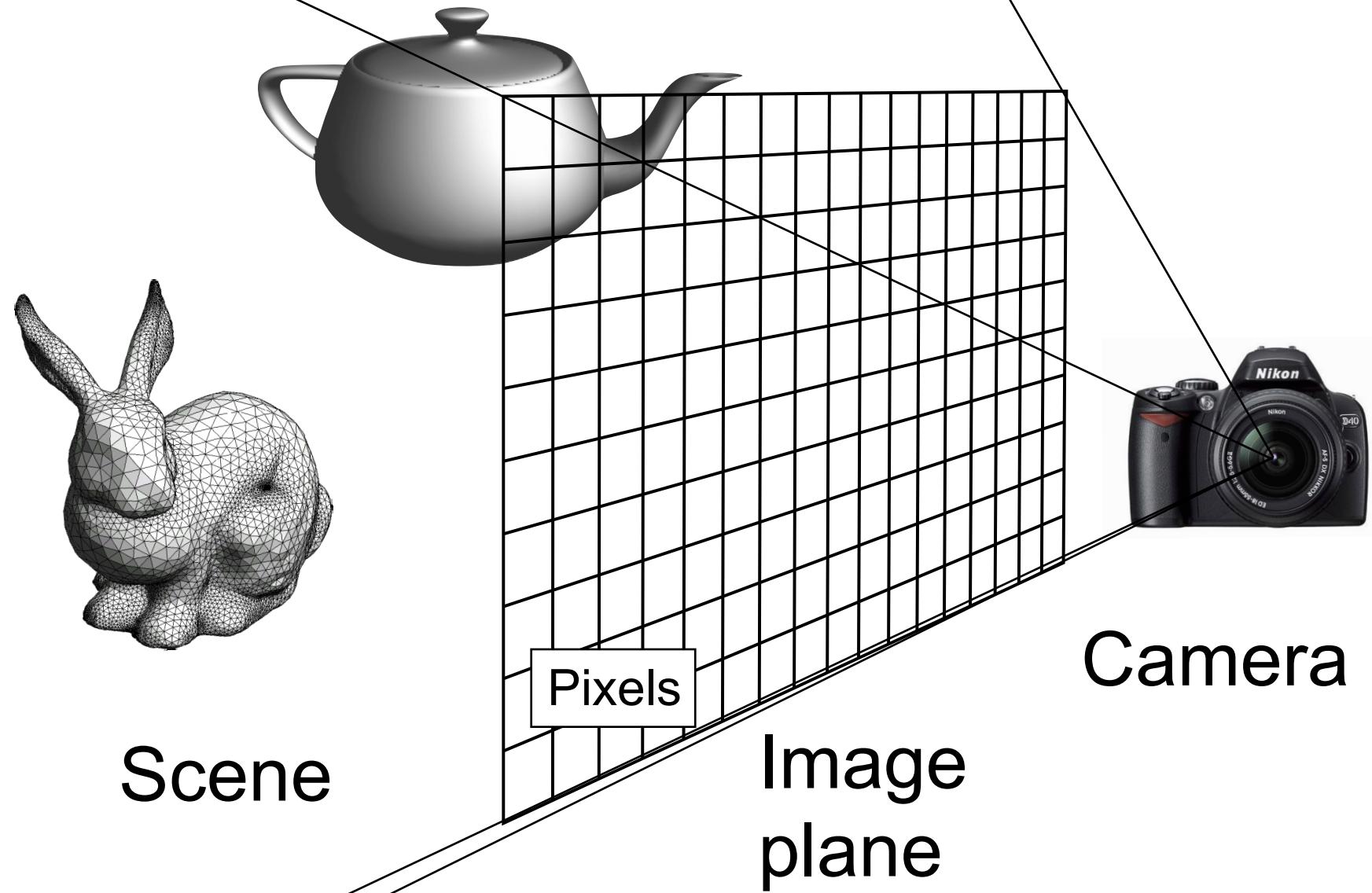
3D + time = 4D

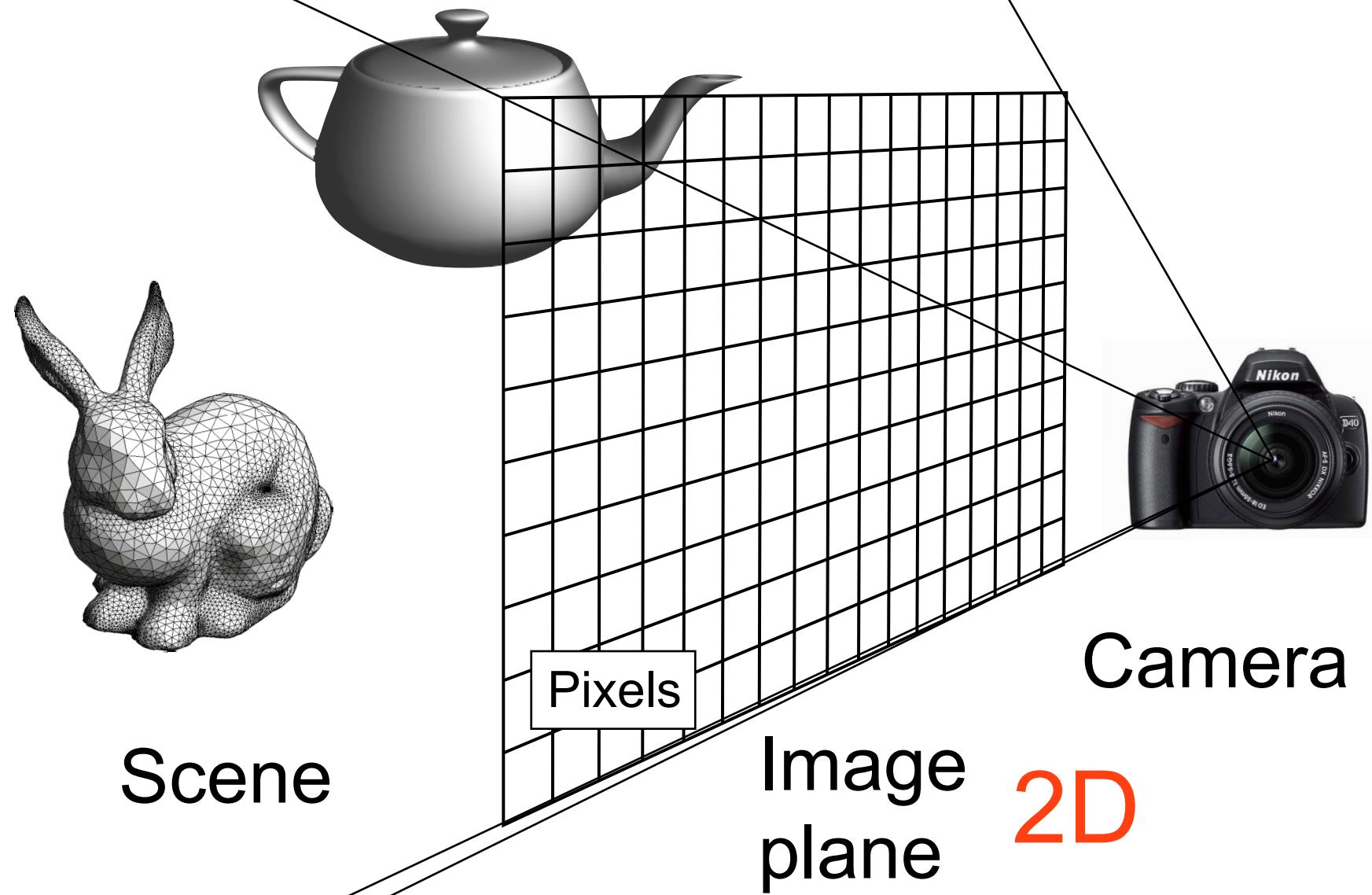
Scene

Scene

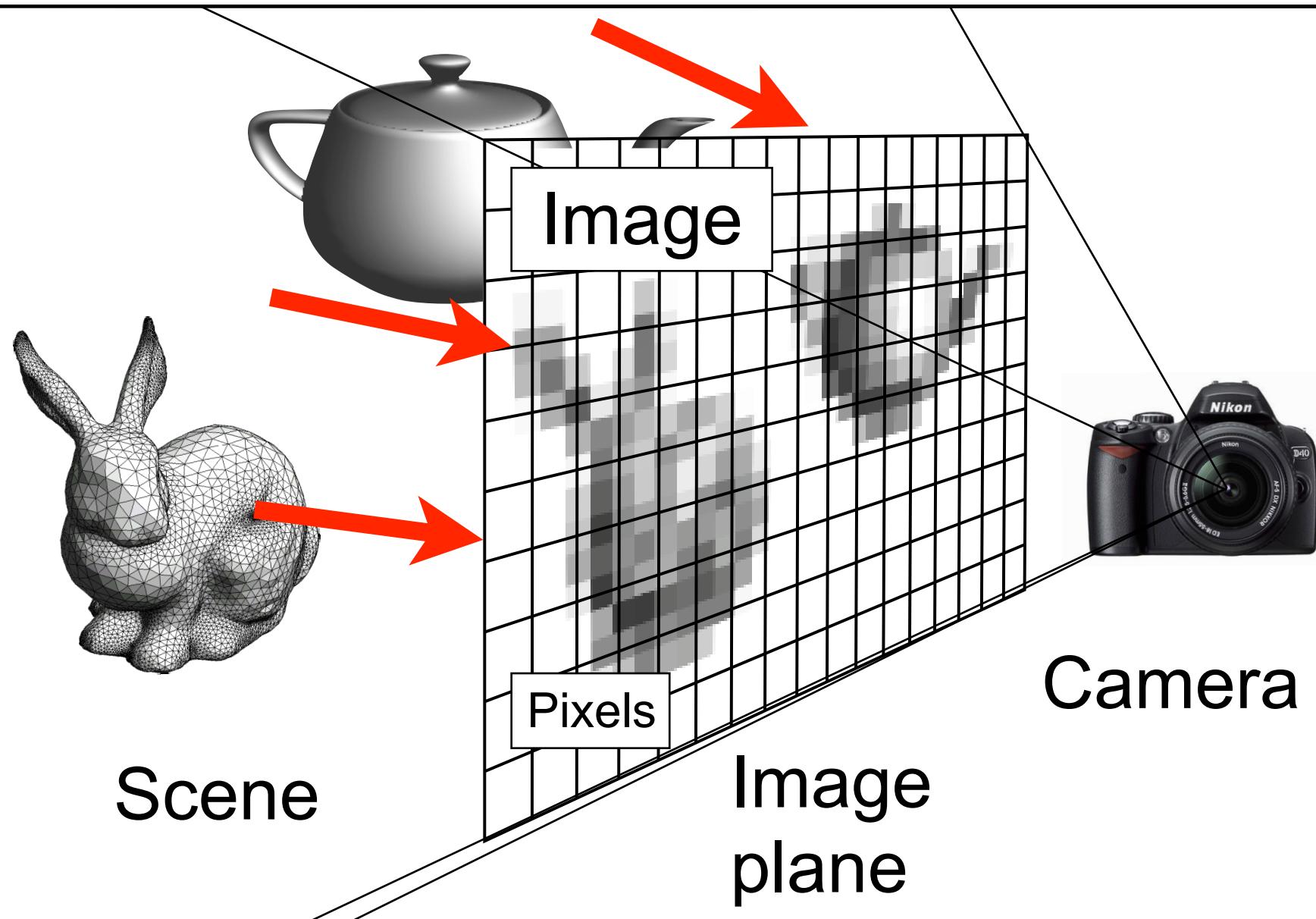


Camera

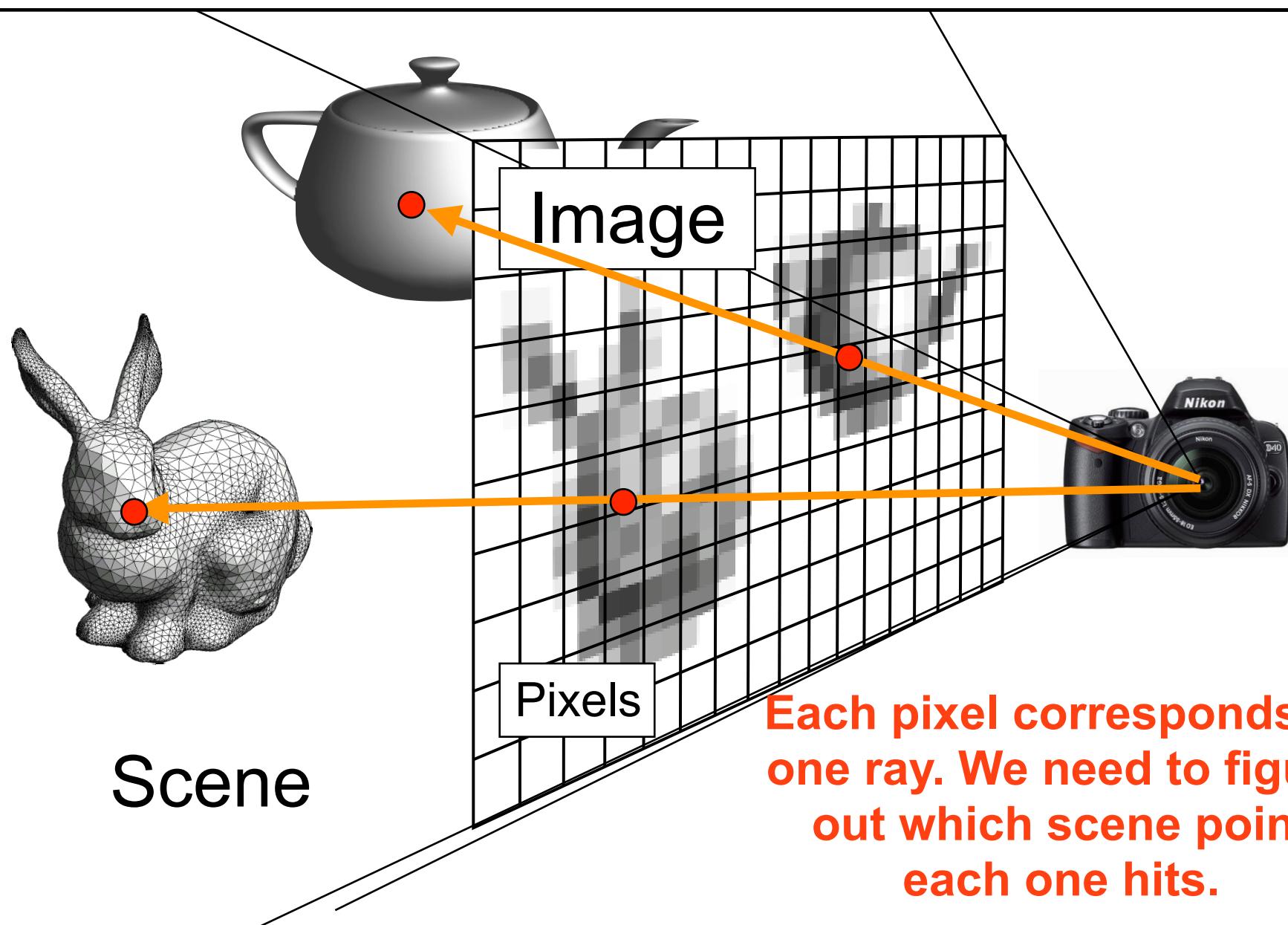




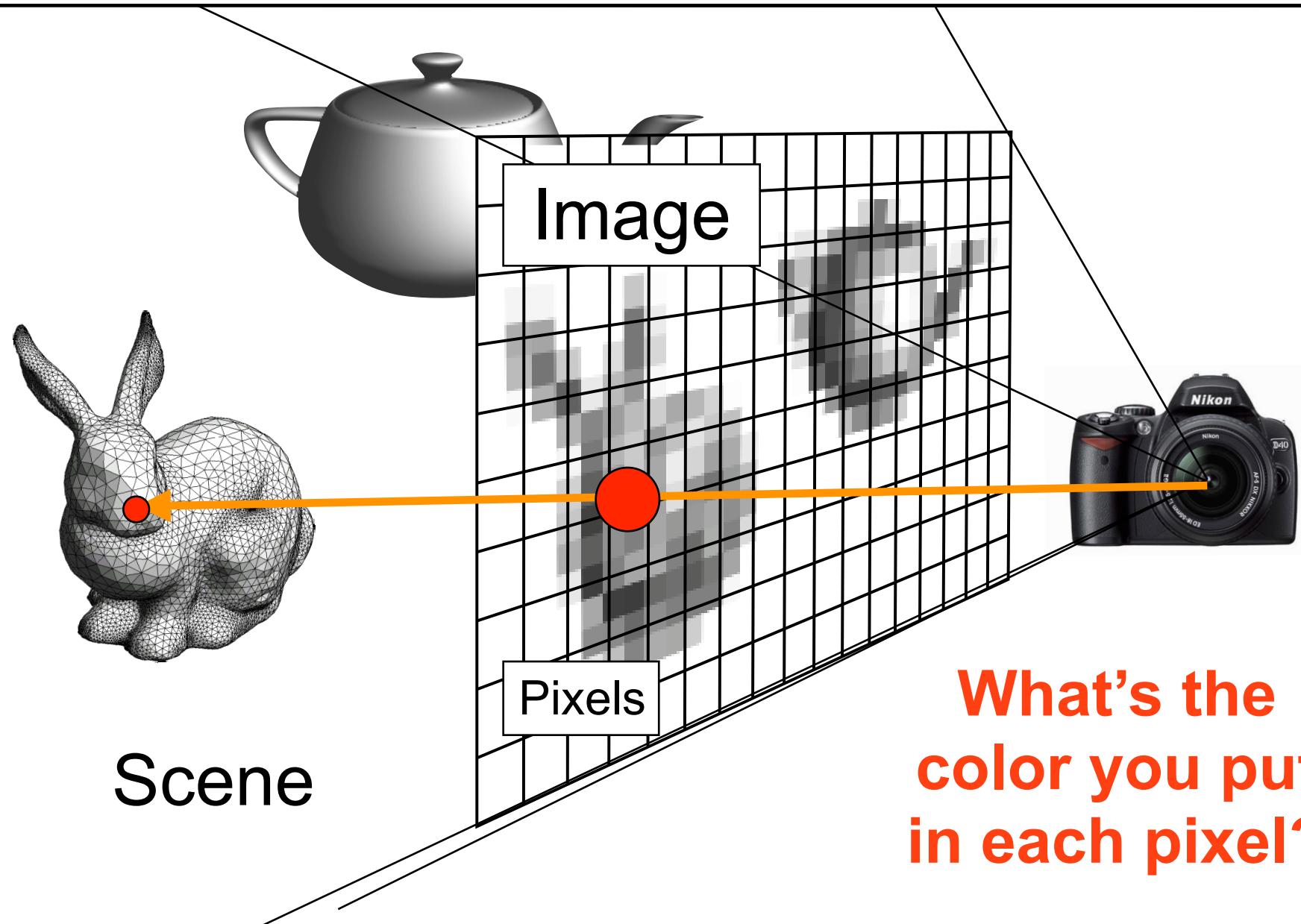
Rendering = Scene to Image



Rendering – Pinhole Camera

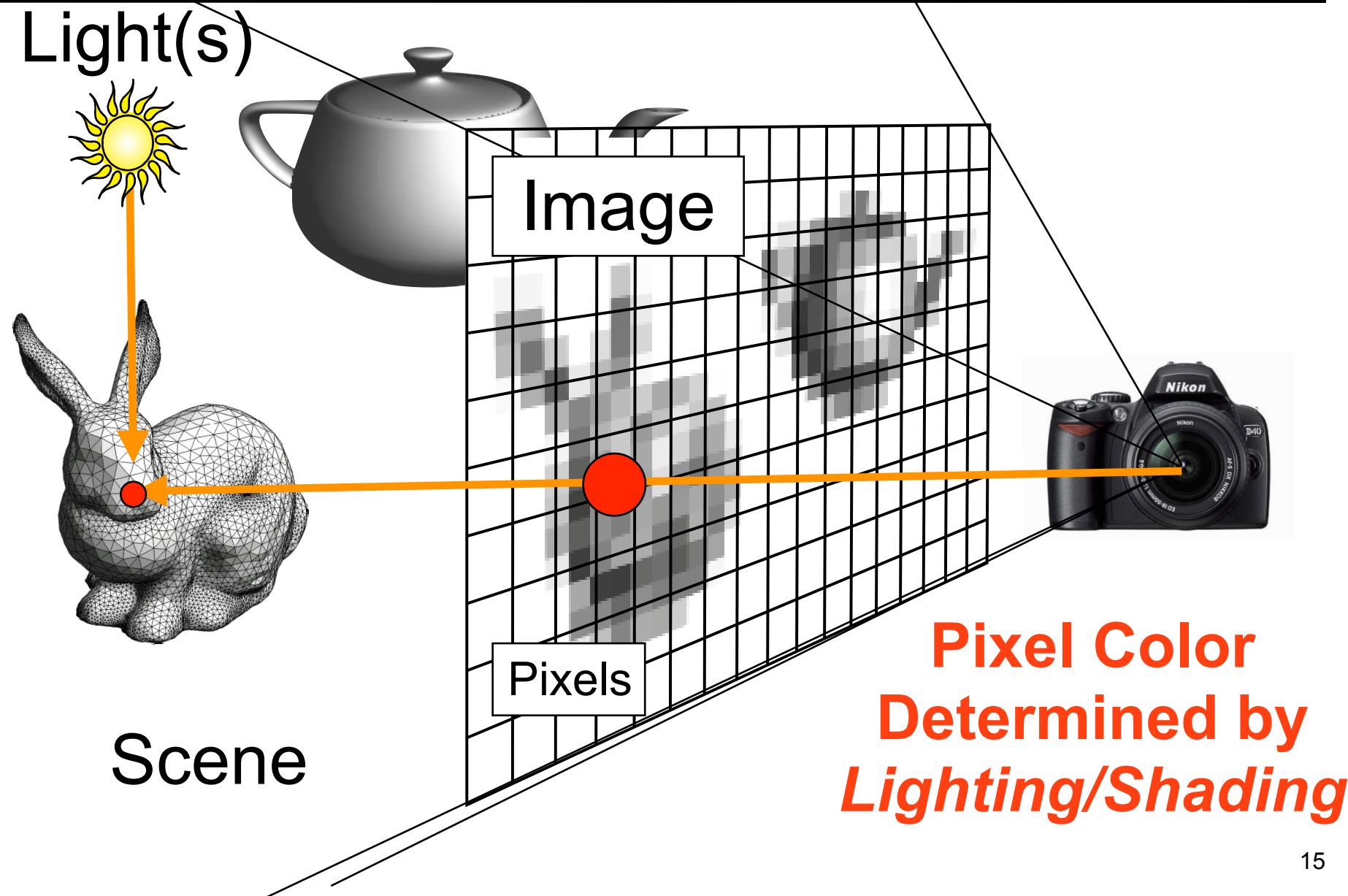


Rendering



What's the
color you put
in each pixel?

Rendering



Rendering

- “Rendering” refers to the entire process that produces colors for pixels
- Pixels correspond to rays; need to figure out the **closest** scene point along each ray
 - Called “hidden surface problem” in older texts
 - “Visibility” is a more modern term
 - Also, we assume (for now) a single ray per pixel

It Works



Rendered using NVIDIA tray

It Works



Star Wars Battlefront 2 / EA / DICE

Rendering

- “Rendering” refers to the entire process that produces color values for pixels
- Pixels correspond to rays; need to figure out the **closest** scene point along each ray
 - Called “hidden surface problem” in older texts
 - “Visibility” is a more modern term
 - Also, we assume (for now) a single ray per pixel
- Major algorithms: **Ray tracing and rasterization**
- Note: We are assuming a pinhole camera (for now)

Ray Tracing vs. Rasterization

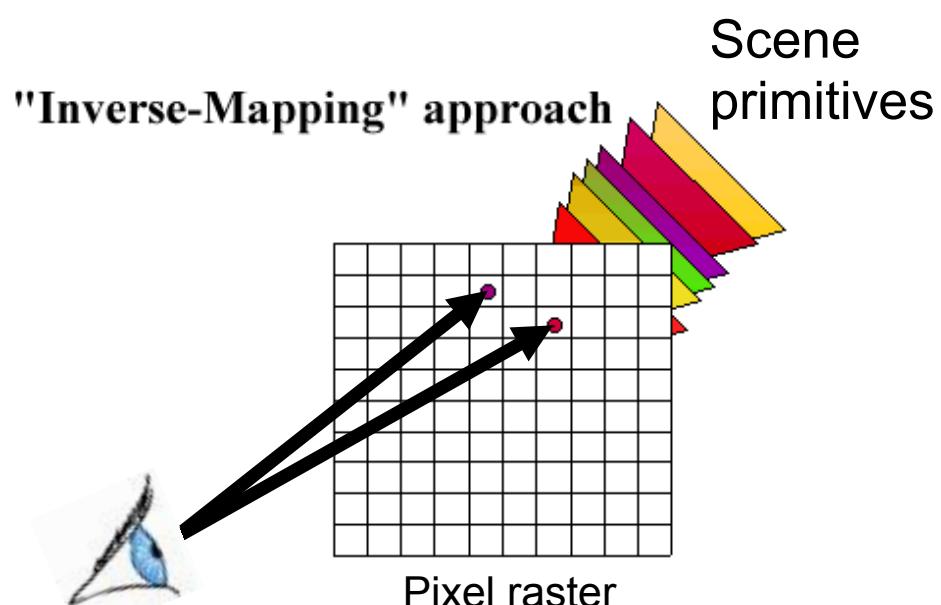
Ray Tracing

For each pixel (ray)

For each object

Does ray hit object?

Keep closest hit



Ray Tracing vs. Rasterization

Ray Tracing

For each pixel (ray)

For each object

Does ray hit object?

Keep closest hit

Rasterization

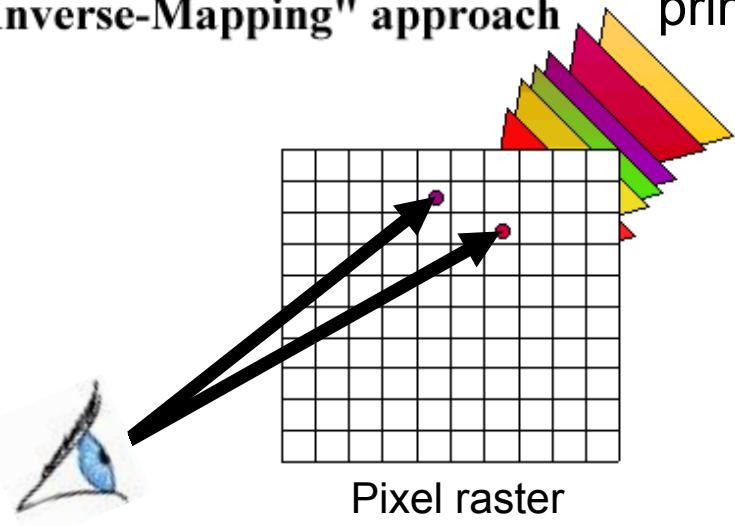
For each triangle

For each pixel

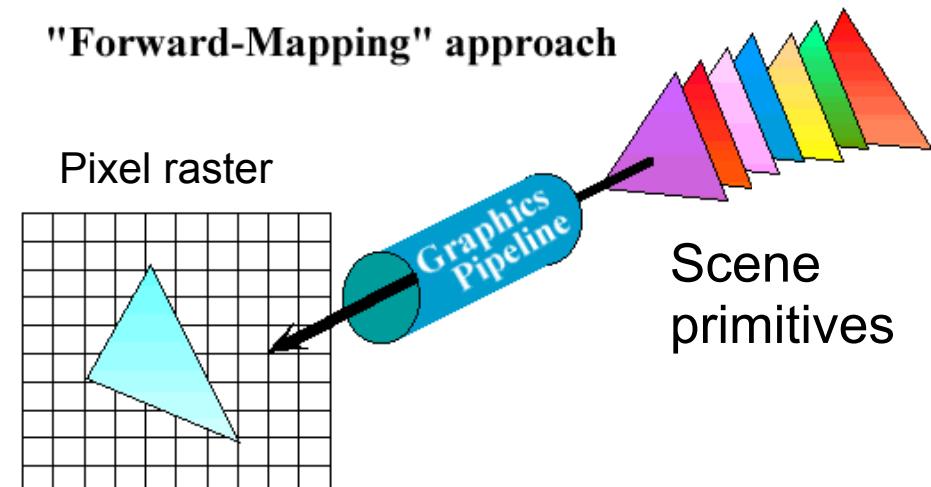
Does triangle cover pixel?

Keep closest hit

"Inverse-Mapping" approach



"Forward-Mapping" approach



Ray Tracing vs. Rasterization

Ray Tracing

For each pixel (ray)

For each object

Does ray hit object?

Keep closest hit

Rasterization

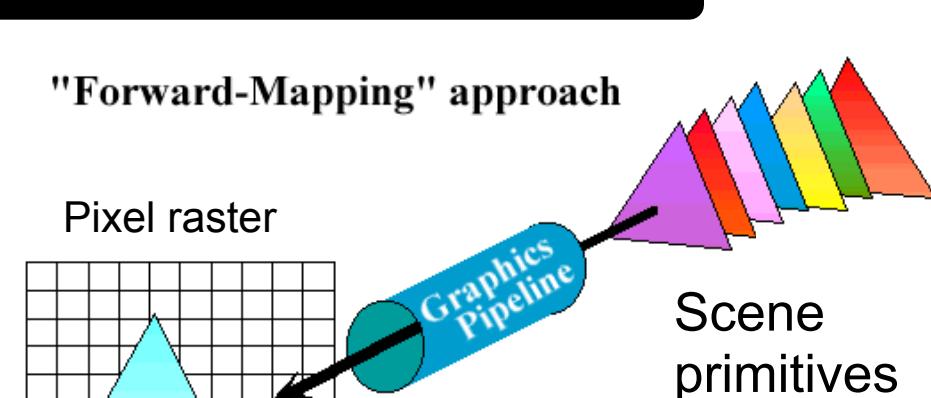
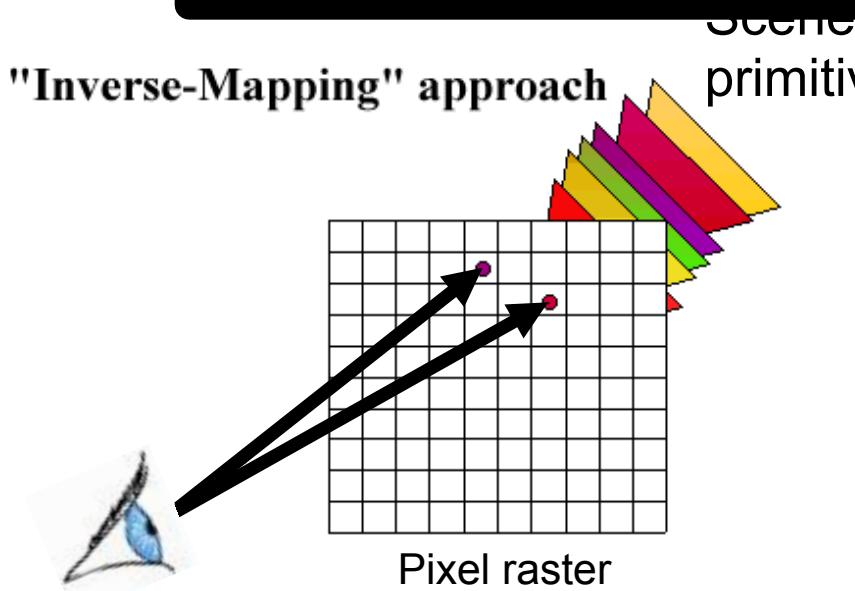
For each triangle

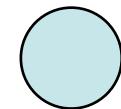
For each pixel

Does triangle cover pixel?

Keep closest hit

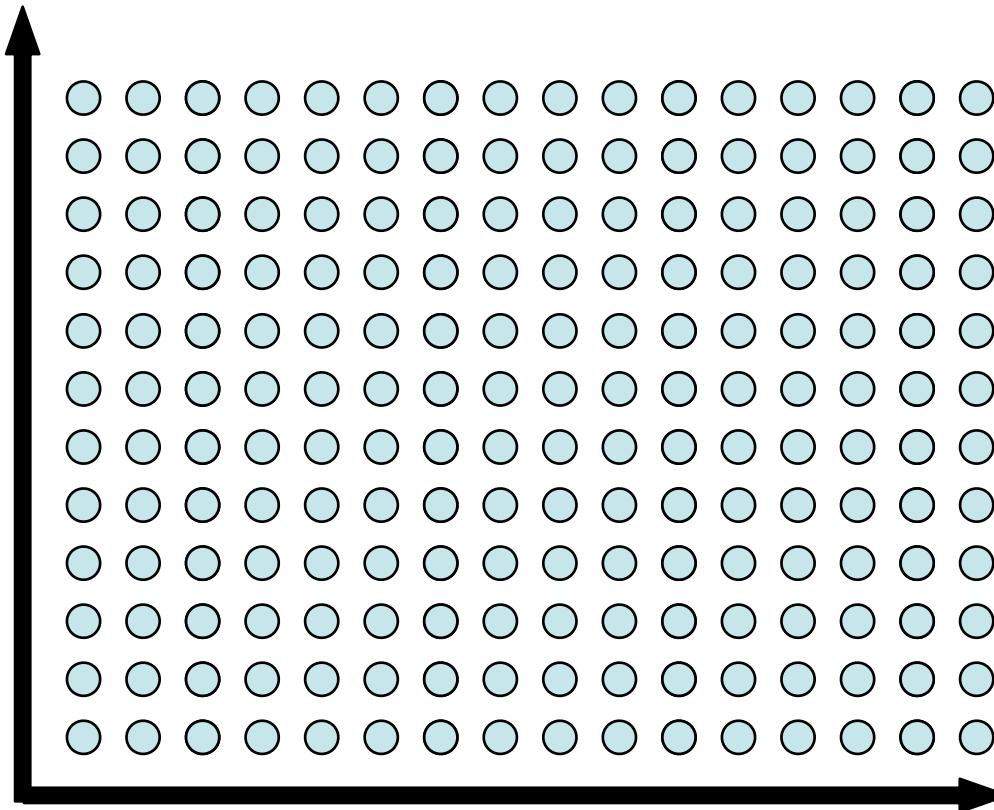
It's just a different order of the loops!





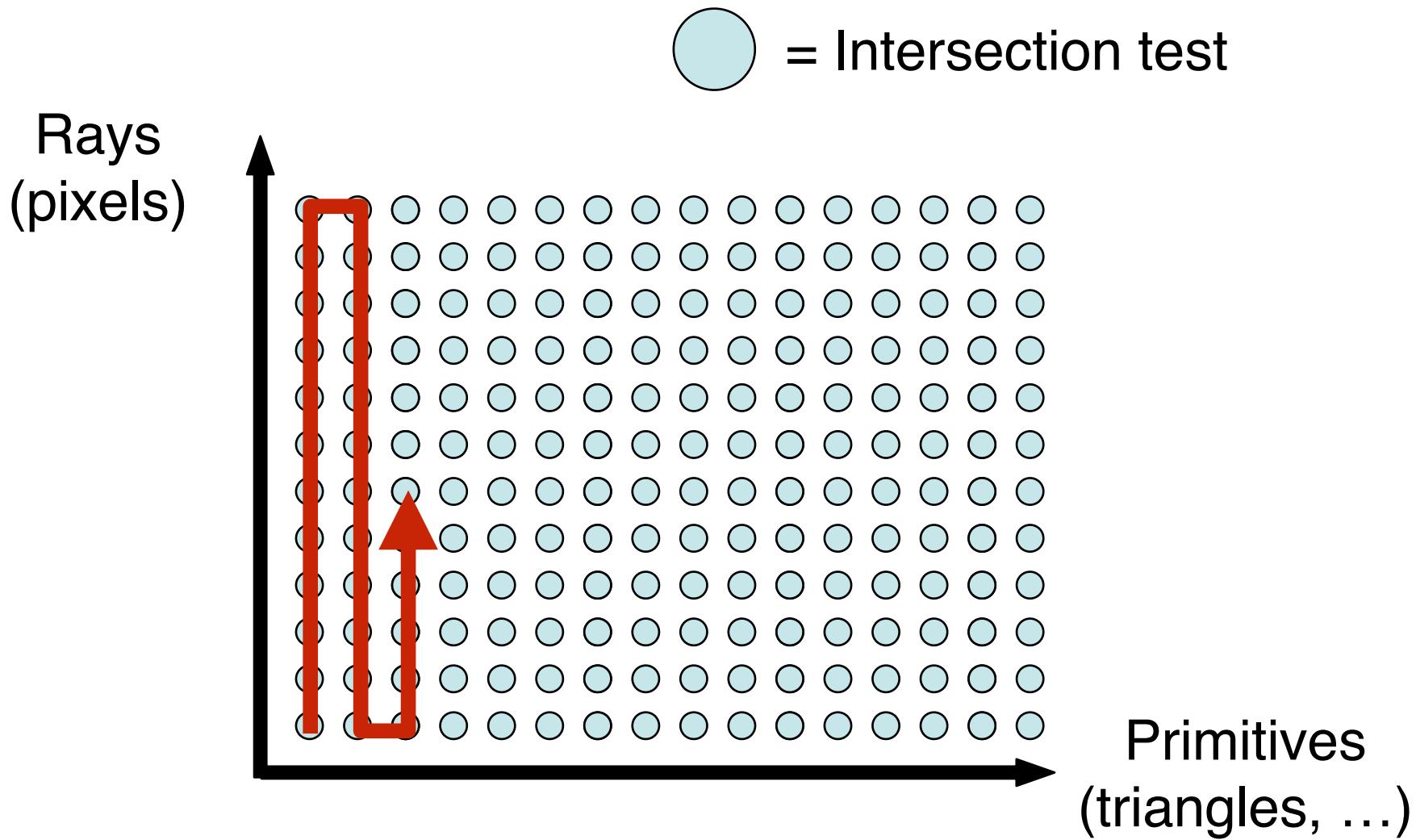
= Intersection test

Rays
(pixels)

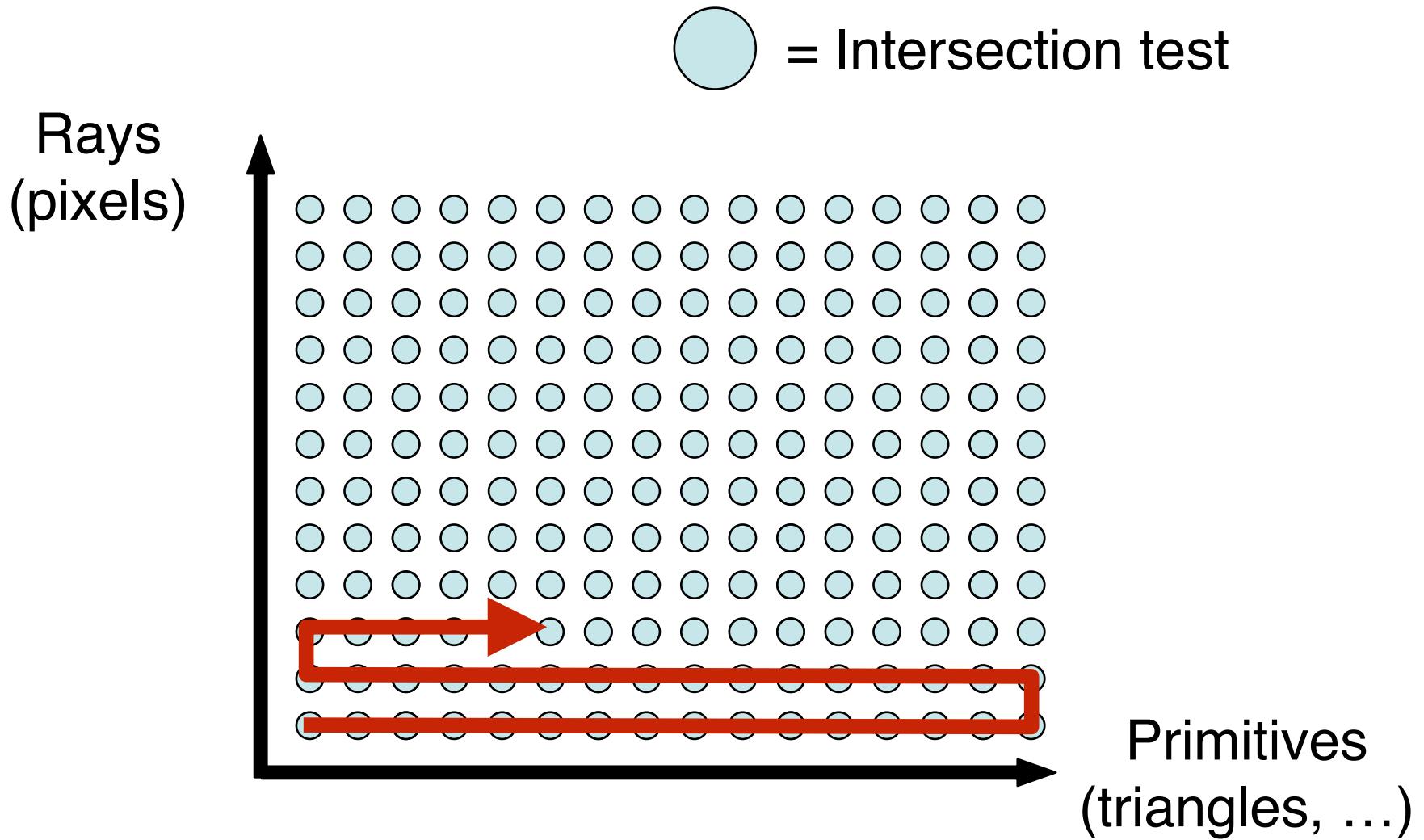


Primitives
(triangles, ...)

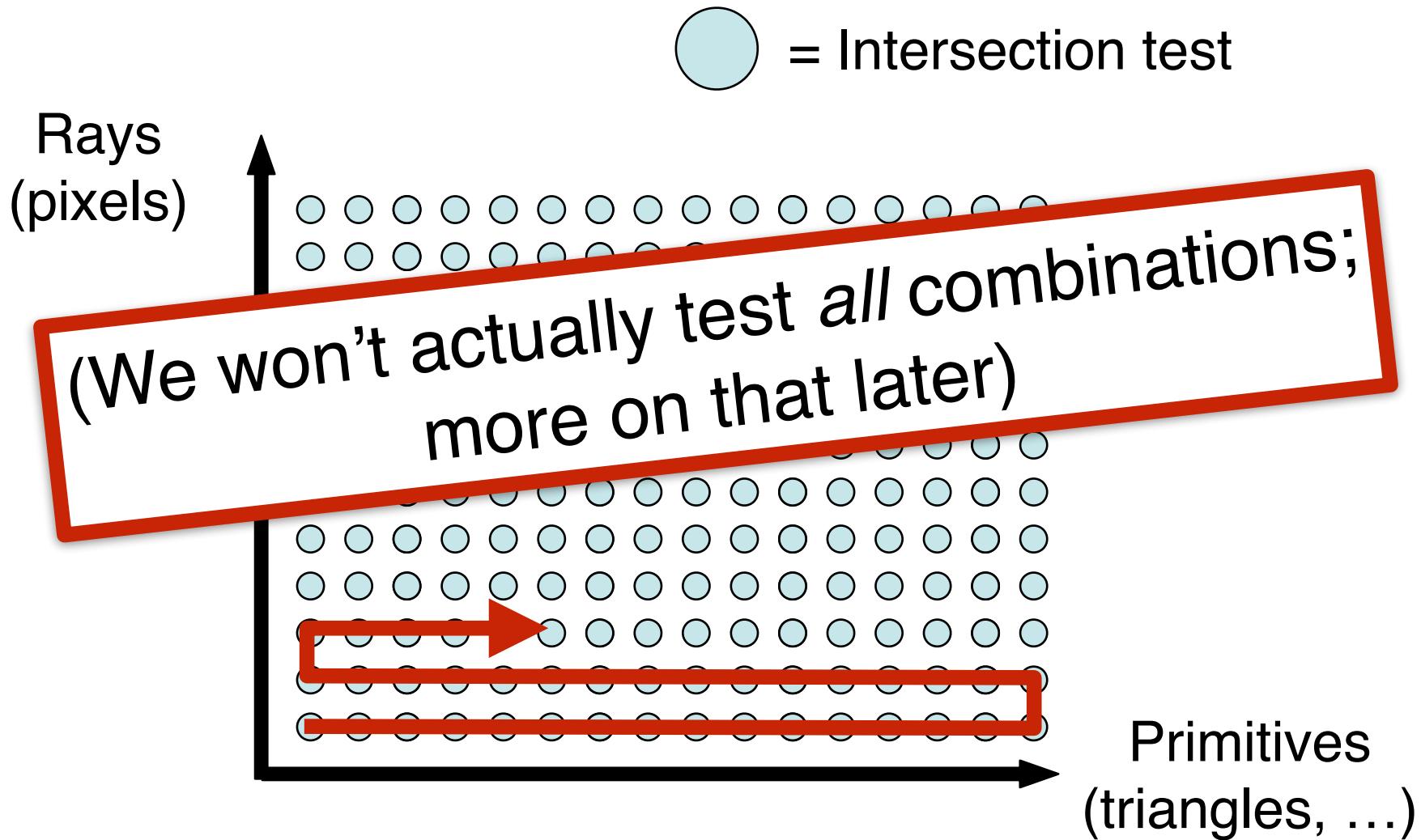
Rasterization = triangles first



Ray tracing = rays first



Ray tracing = rays first



That's All!

Up next: basics of ray tracing

