CS-C3100 Computer Graphics

Part 2 – Rendering

11.2 Basics of Ray Tracing

Jaakko Lehtinen
with lots of slides from Frédo Durand

In This Video

- Basics of ray tracing
  - Ray tracing loop
  - Types of rays:
    - primary
    - secondary
    - shadow
  - Ray representation
    - parametric line: O + t*D

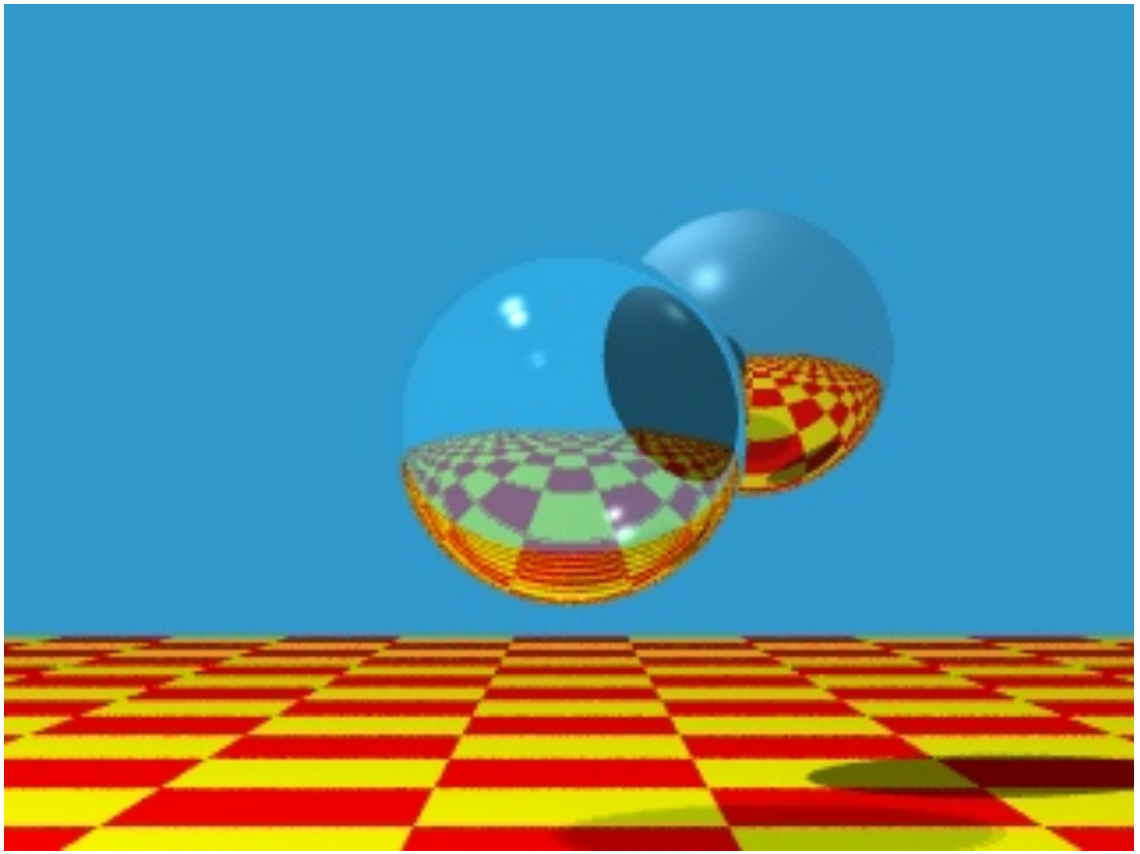
Origins of Ray Tracing (paper)

Graphics and Image Processing
J.D. Foley Editor

An Improved Illumination Model for Shaded Display

Turner Whitted
Bell Laboratories
Holmdel, New Jersey

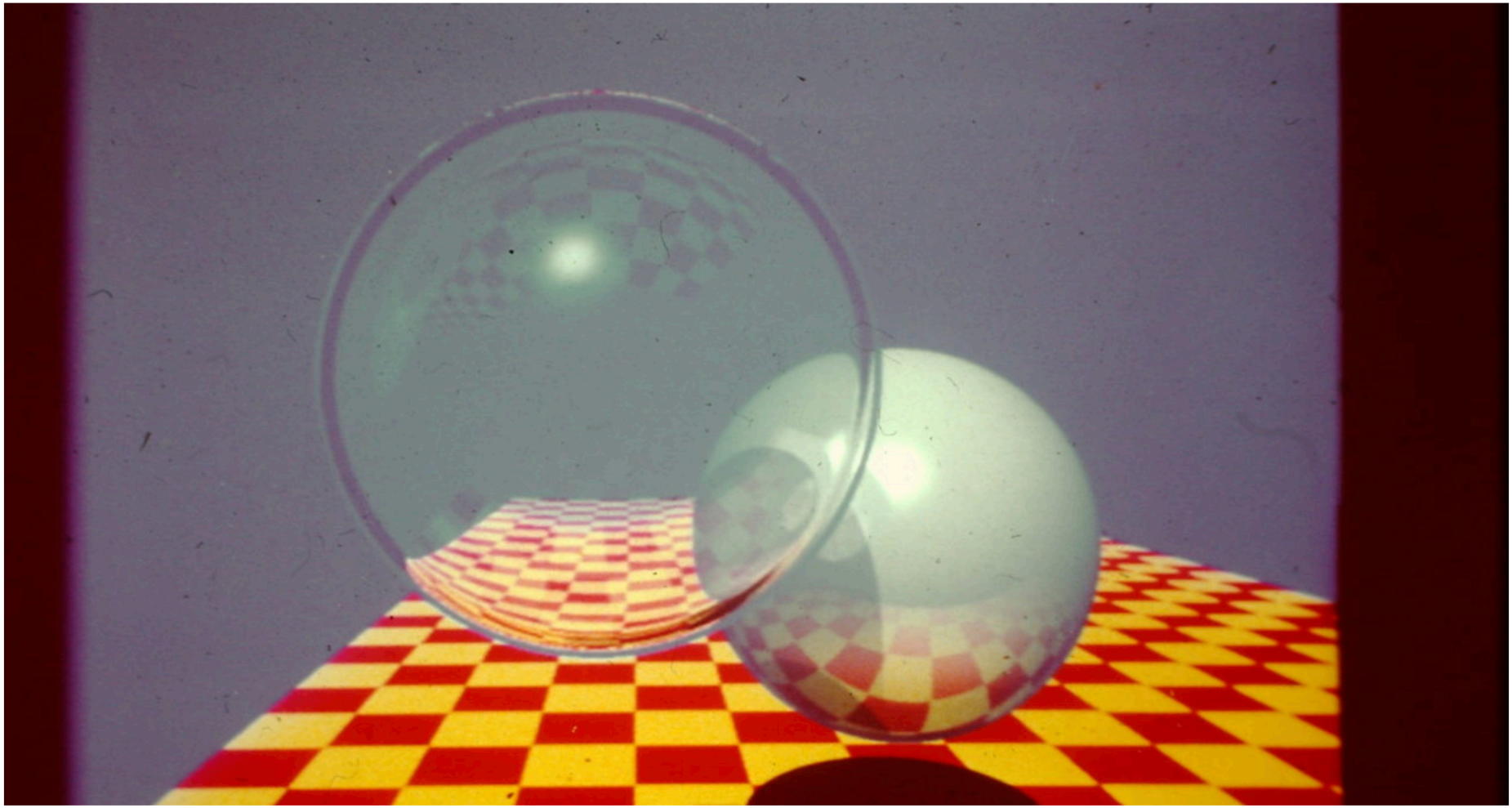
Communications of the ACM 23(6), 1980



Turner's blog post from summer '18

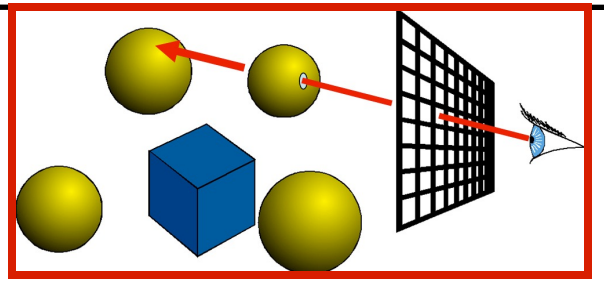
A Ray-Tracing Pioneer Explains How He Stumbled into Global Illumination

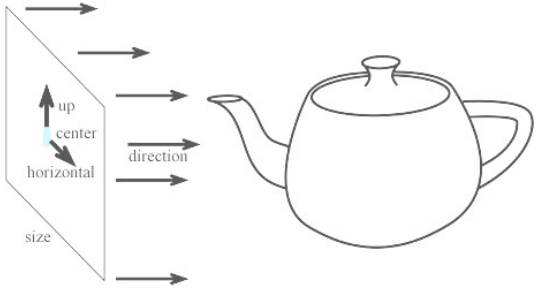
August 1, 2018 by J. TURNER WHITTED

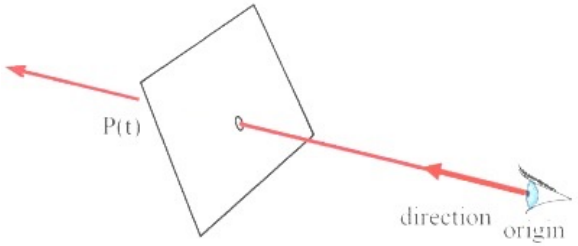


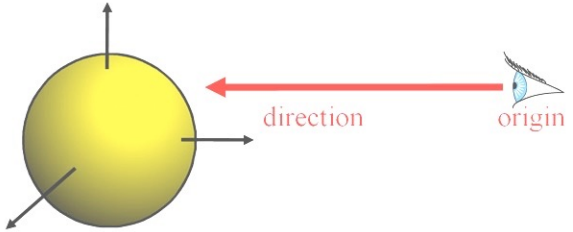
Ray Tracing

- Ray Tracing Basics
- Camera and Ray Generation
- Ray-Plane Intersection
- Ray-Sphere Intersection



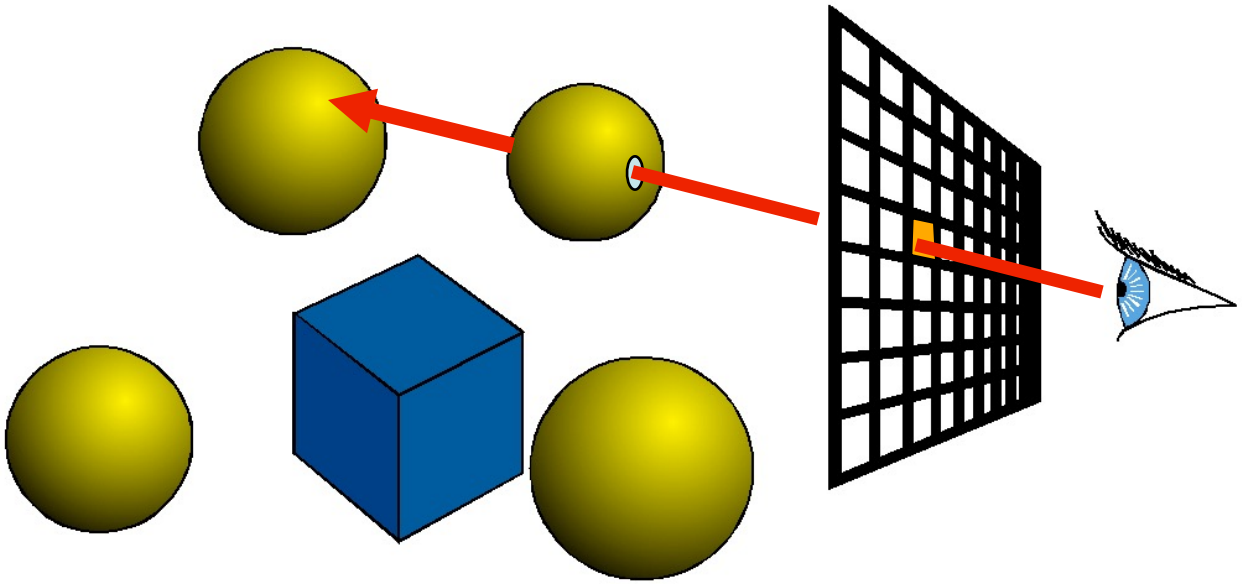






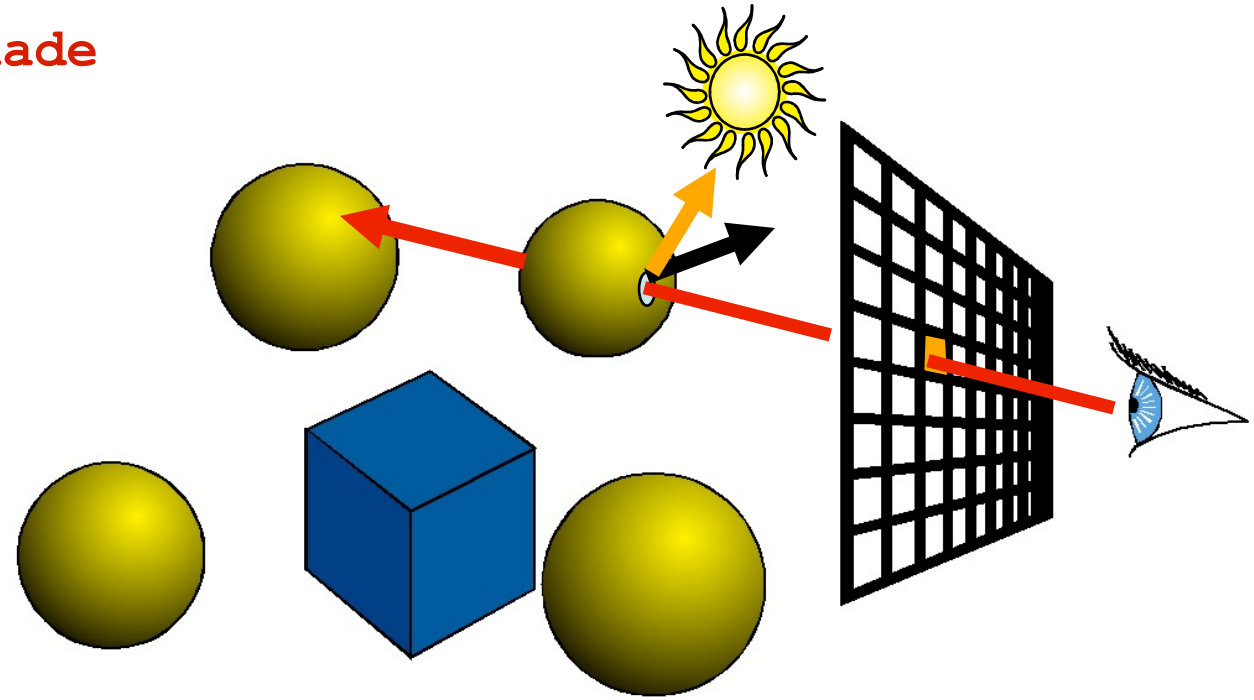
Simple Ray Tracing

```
For every pixel
   Construct a ray from the eye
   For every object in the scene
       Find intersection with the ray
       Keep if closest
```



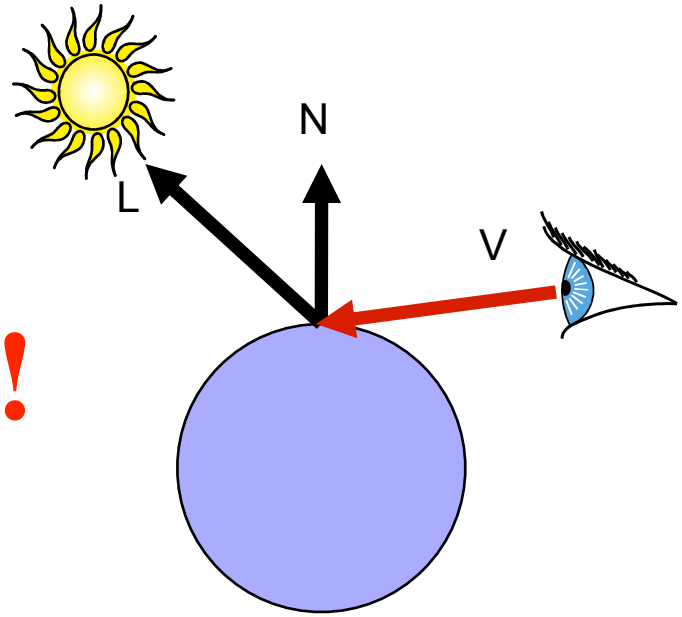
Shading

```
For every pixel
   Construct a ray from the eye
   For every object in the scene
       Find intersection with the ray
       Keep if closest
   Shade
```



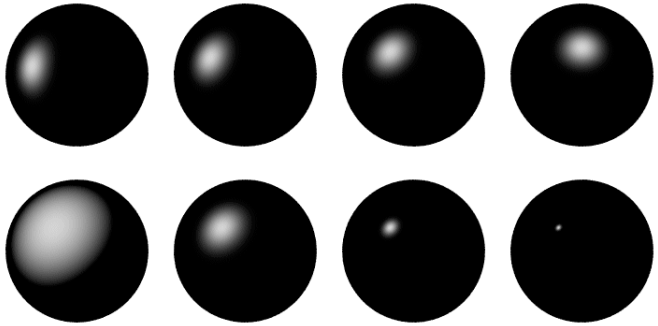
Shading = What Surfaces Look Like

- Surface/Scene Properties
  - surface normal
  - direction to light
  - viewpoint
- Material Properties
  - Diffuse (matte)
  - Specular (shiny)
  - …
- Light properties
  - Position
  - Intensity, ...
- Much more!



Later!

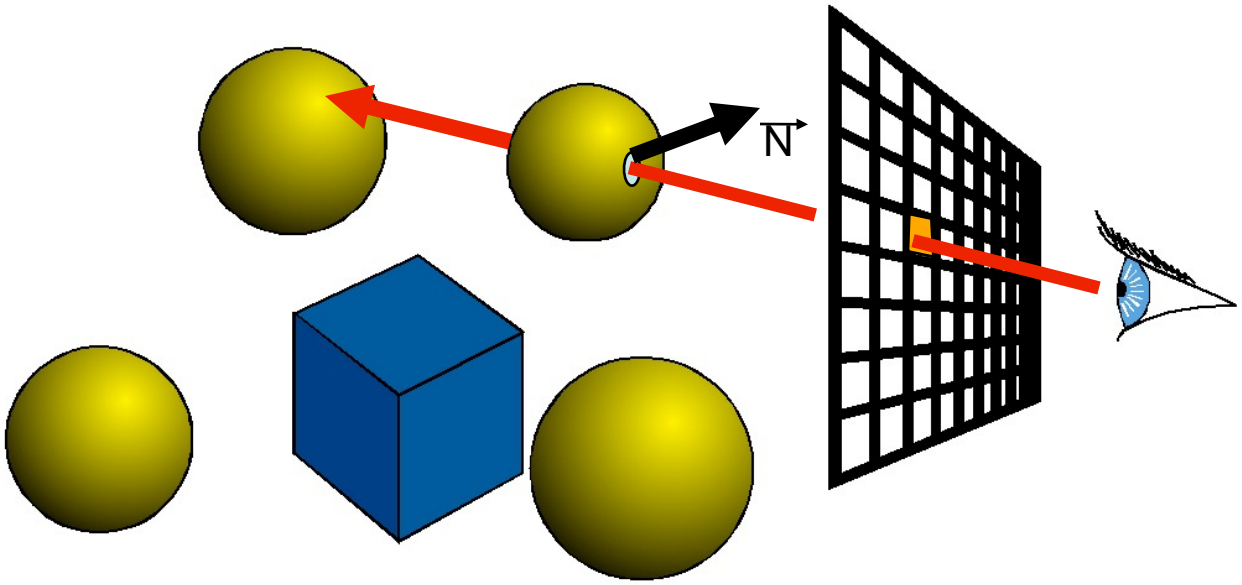
Diffuse sphere



Specular spheres

Simple Ray Tracing

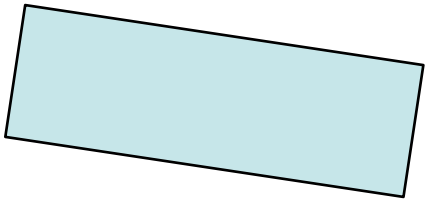
```
For every pixel
   Construct a ray from the eye
   For every object in the scene
       Find intersection with the ray
       Keep if closest
    Shade depending on light and normal vector
```

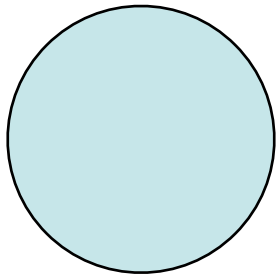


Finding the intersection point and normal is the central part of ray tracing

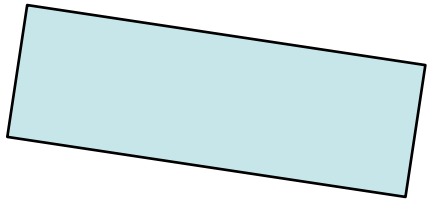
Types of Rays

- Let's think about shadows...

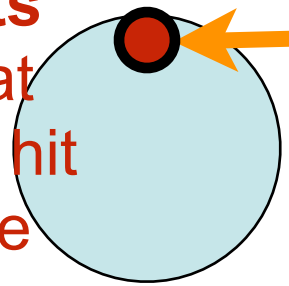




Types of Rays: Primary



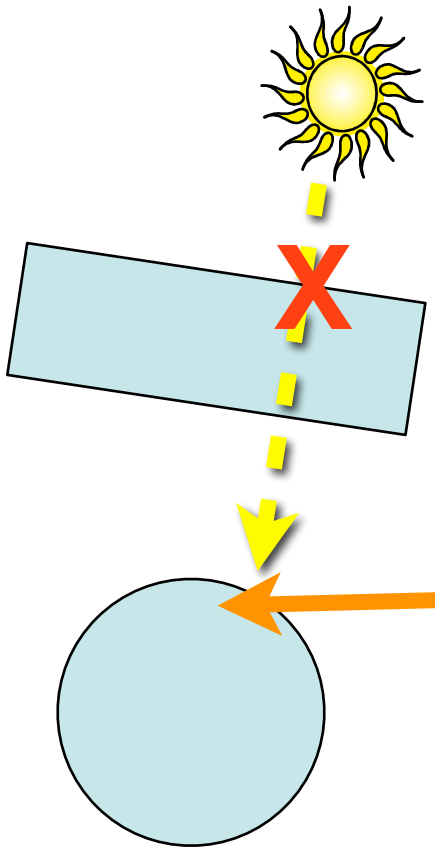
?



“**primary hits**”
= points that primary rays hit in the scene

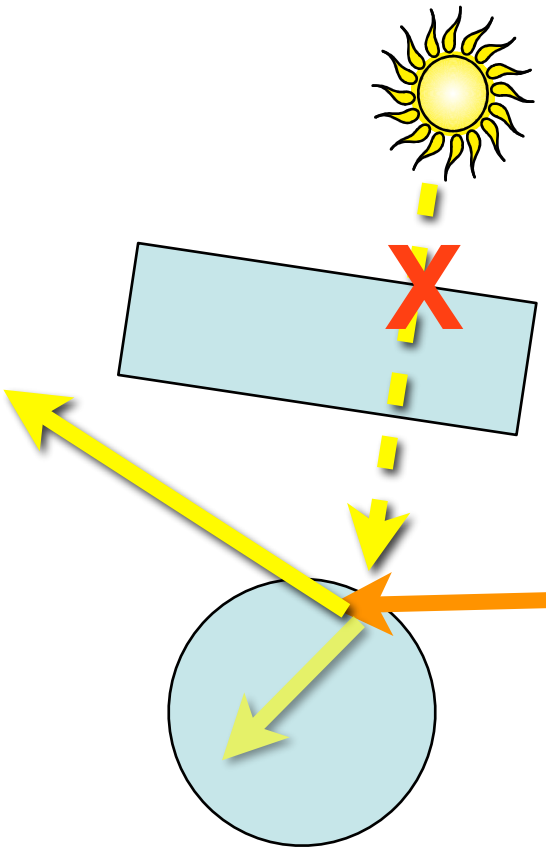
“**primary/camera/eye rays**” are rays from the camera to the scene

Types of Rays: Shadow



ray from light to hit point is blocked, i.e., **point is in shadow**

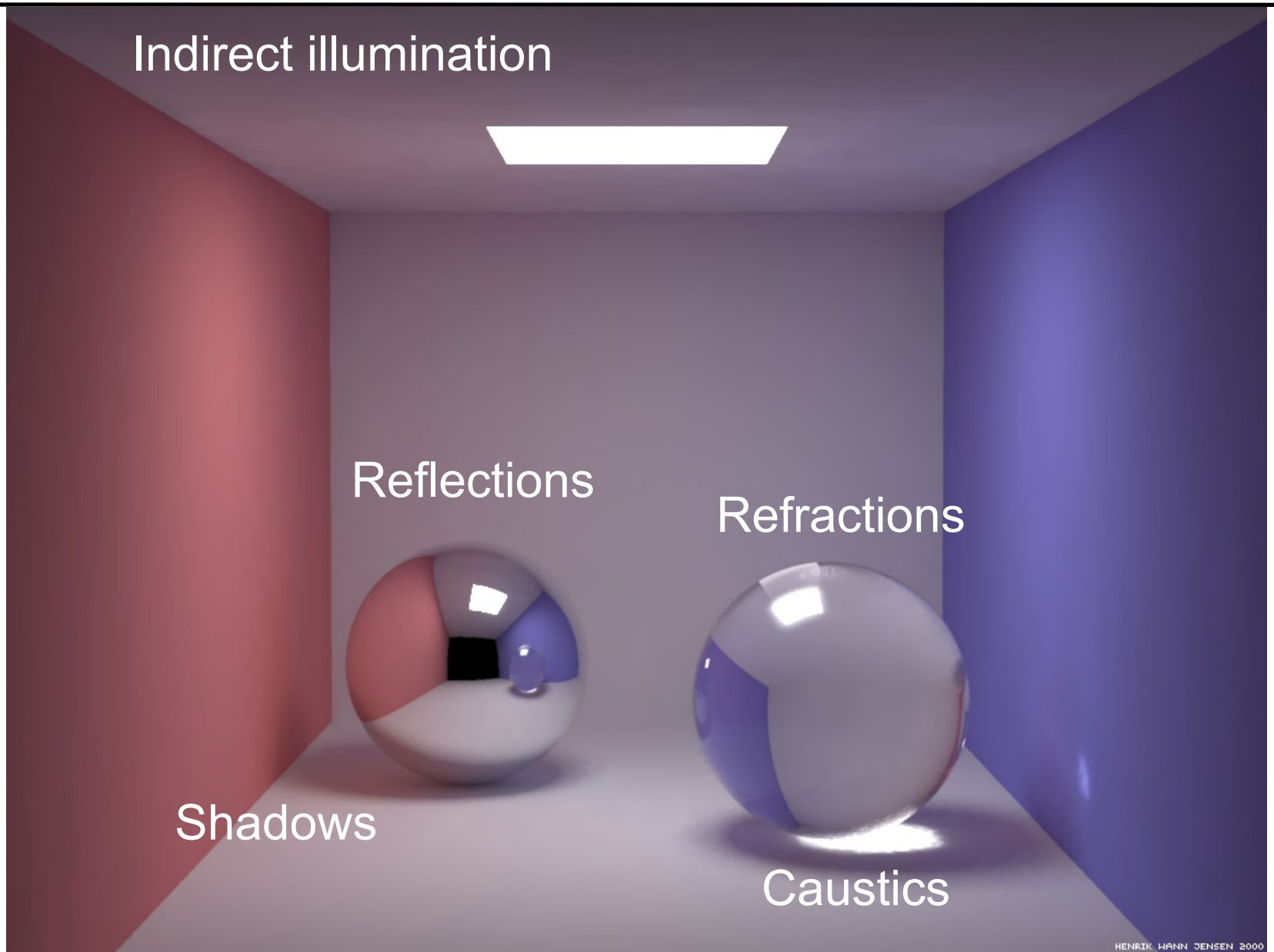
Types of Rays: Secondary



Primary rays start from camera
Secondary rays are used for testing shadows, doing reflections, refractions, etc.

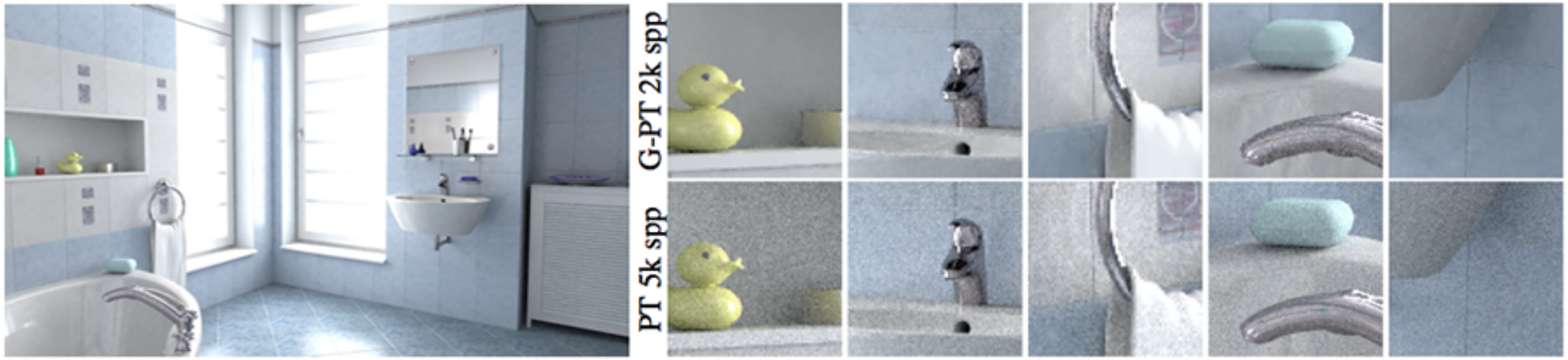
We'll do all this a little later!

Secondary Rays



(We Do Research in This)

- Several recent state-of-the art Monte Carlo rendering techniques
  - This one is Gradient-domain Path Tracing



Another paper here

Temporal Gradient-Domain Path Tracing

Marco Manzi[1,⋆] Markus Kettunen[2,⋆] Frédo Durand[4] Matthias Zwicker[1] Jaakko Lehtinen[2,3]

[1]University of Bern [2]Aalto University [3]NVIDIA [4] MIT CSAIL

⋆ *Both authors contributed equally to this work.*

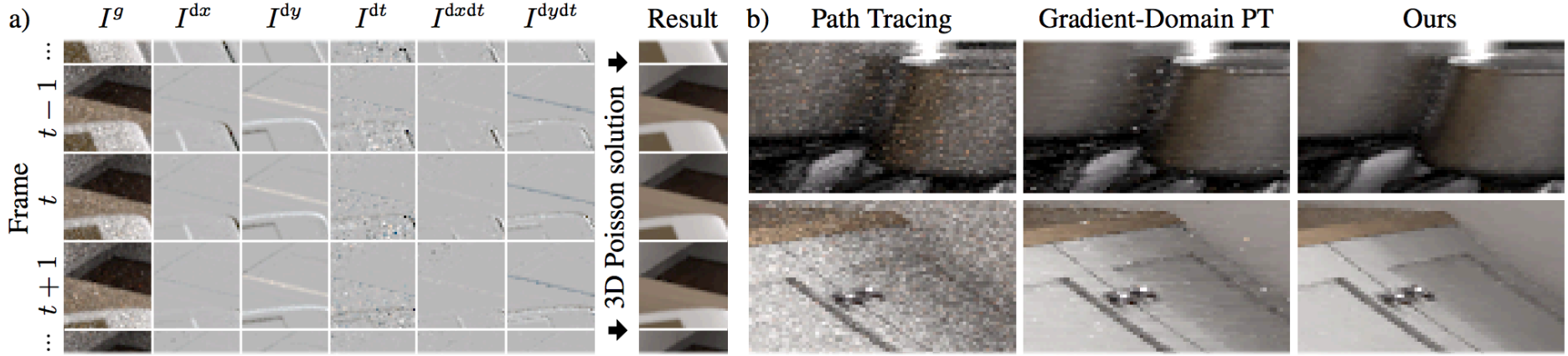
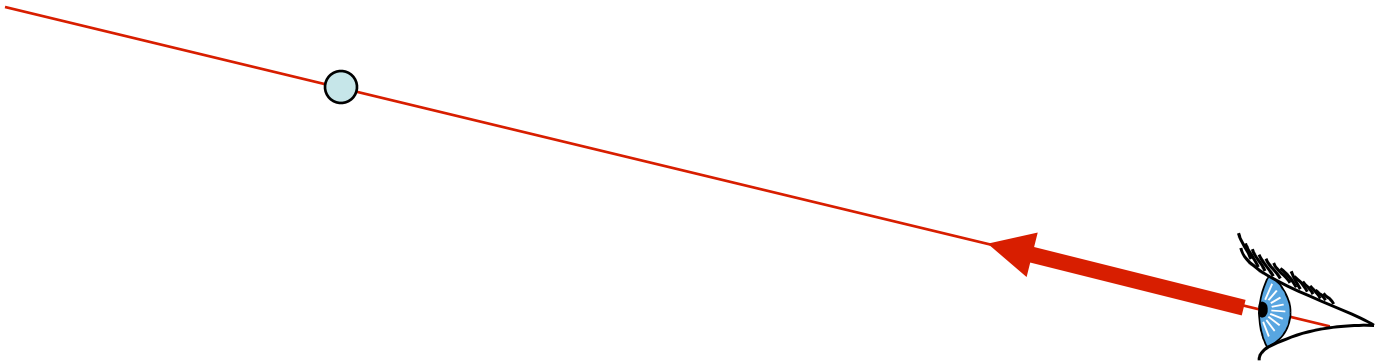


Figure 1: a) *In addition to standard path sampling, our method also estimates spatial, temporal and mixed finite differences for the frames of an animation. We then solve a 3D screened Poisson problem to reconstruct the animation whose frames best match the sampled data.* b) *Equal-time rolling shutter crops of the animation* KITCHEN 2. *The rows are extracted from sequential animation frames. Our method often reduces both spatial variance, seen as horizontal noise, and flickering, seen as vertical noise.*

Ray Representation

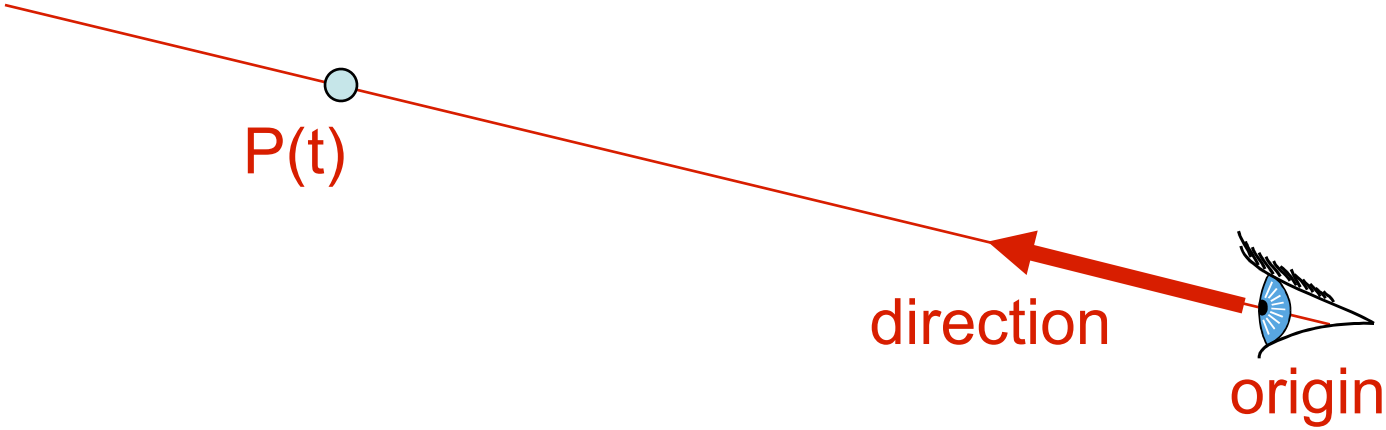
How would you represent a ray?



Ray Representation

- Two vectors:
  - Origin
  - Direction
- Parametric line
  - P(t) = origin + t * direction

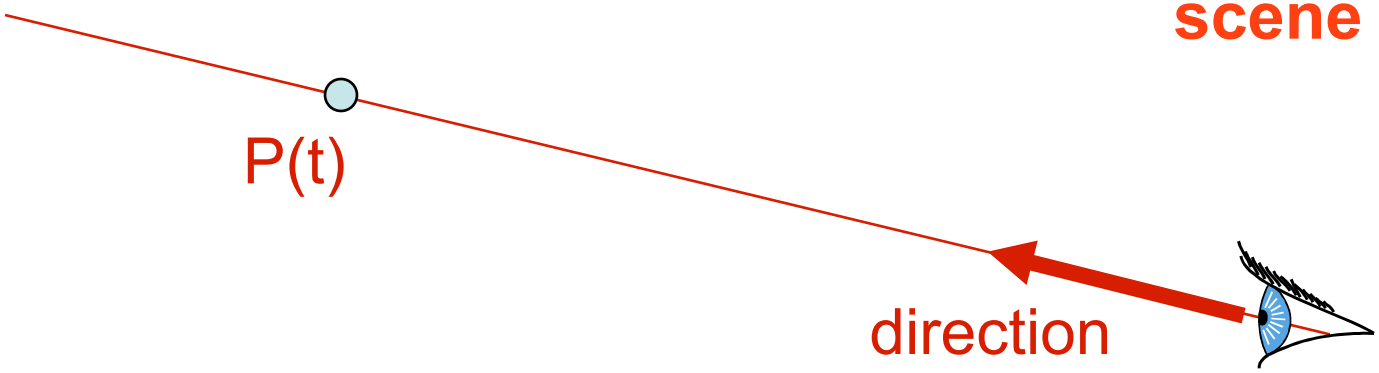
How would you represent a ray?



Ray Representation

- Two vectors:
  - Origin
  - Direction
- Parametric line
  - P(t) = origin + t * direction

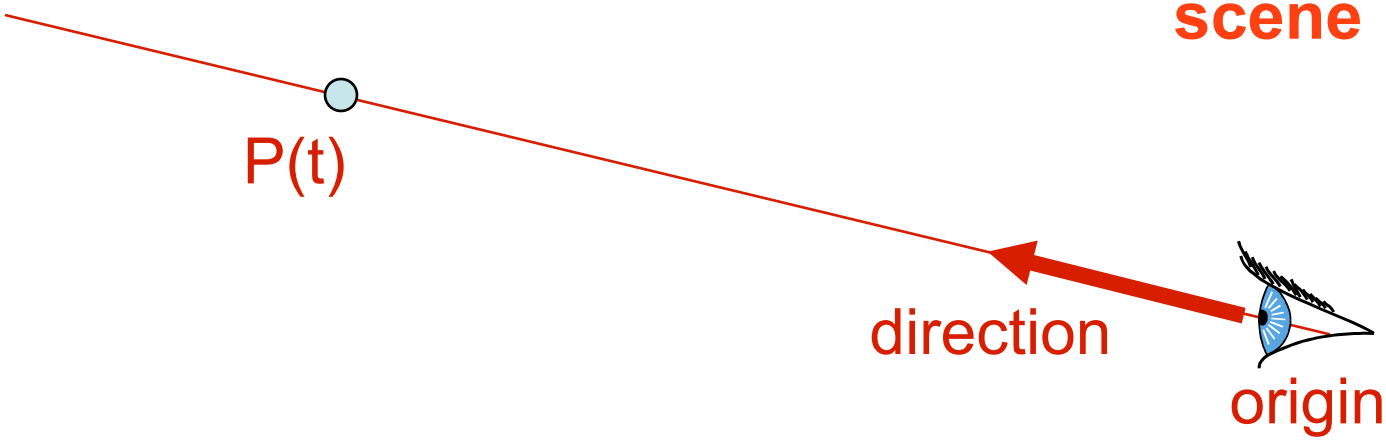
Another way to put the ray casting problem statement:
Find smallest t > 0 such that P(t) lies on a surface in the scene



Ray Representation

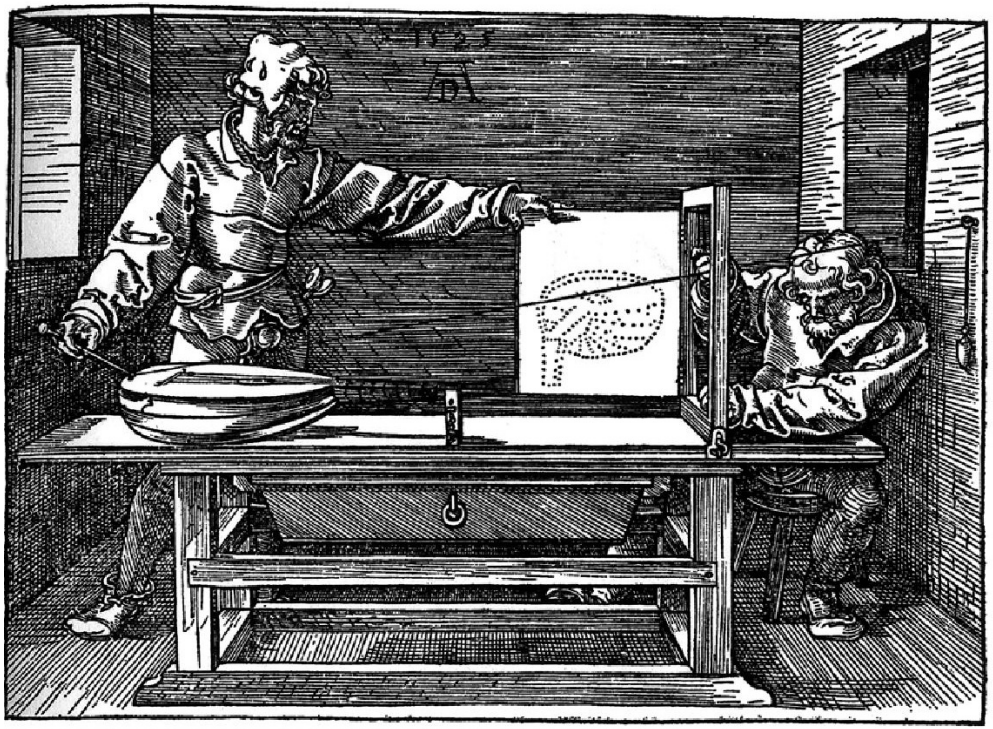
- Two vectors:
  - Origin
  - Direction
- Parametric line
  - P(t) = origin + t * direction

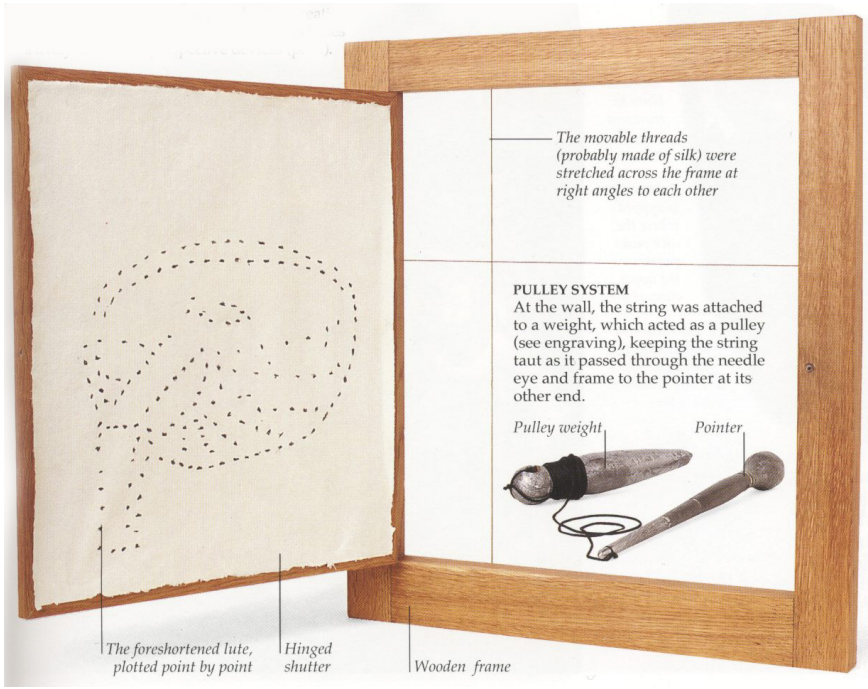
Another way to put the ray casting problem statement:
Find smallest t > 0 such that P(t) lies on a surface in the scene



Dürer's Ray Casting Machine

- Albrecht Dürer, 16th century





Dürer's Ray Casting Machine

- Albrecht Dürer, 16th century

That's All!

Up next: cameras and generating rays