Computer Graphics

Prof. Jaakko Lehtinen

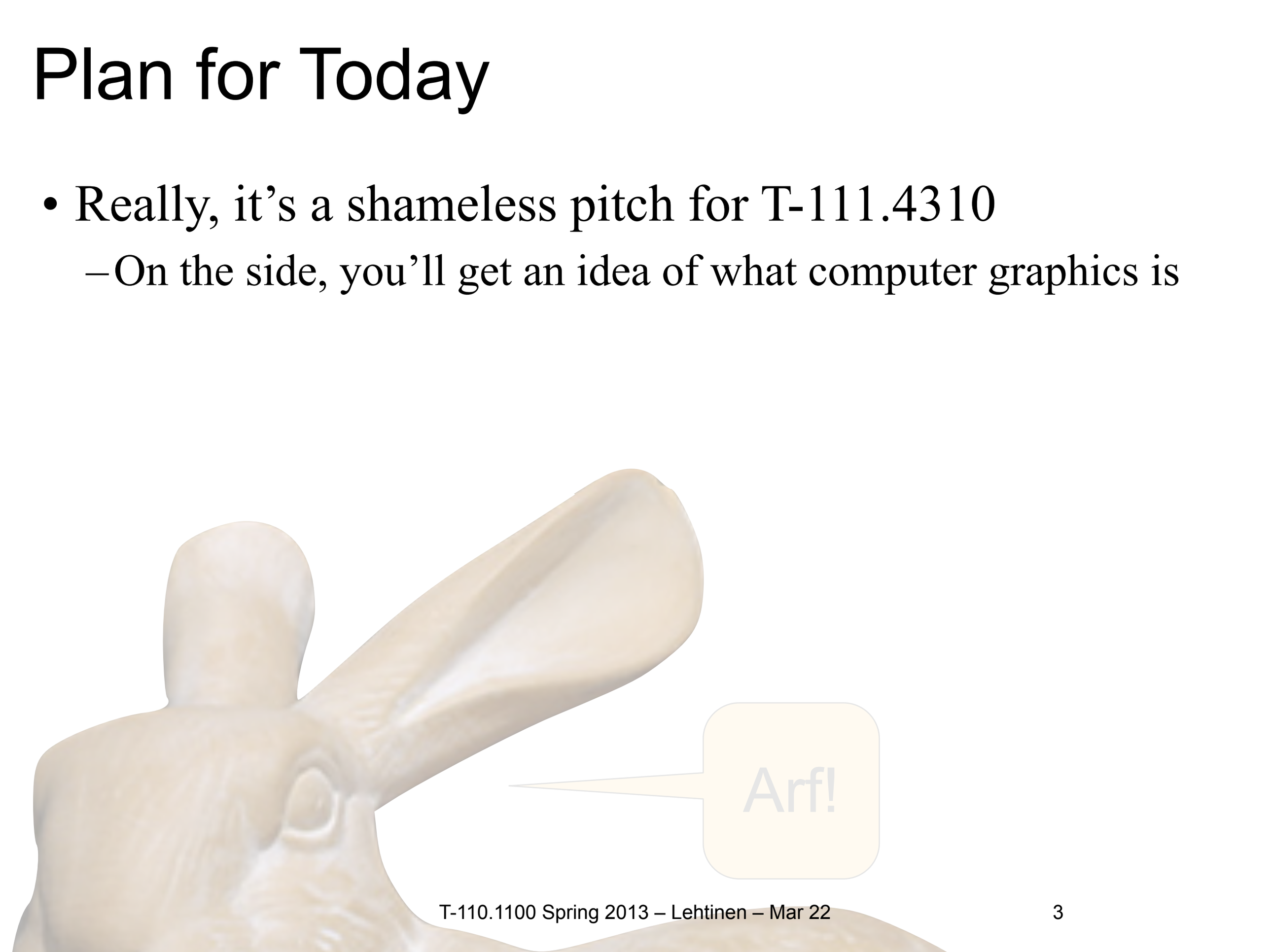
with lots of material from Frédo Durand

Picture: Alexis Rufatt

Luxo Jr. (Pixar, 1986)

Plan for Today

- Really, it's a shameless pitch for T-111.4310
  - On the side, you'll get an idea of what computer graphics is

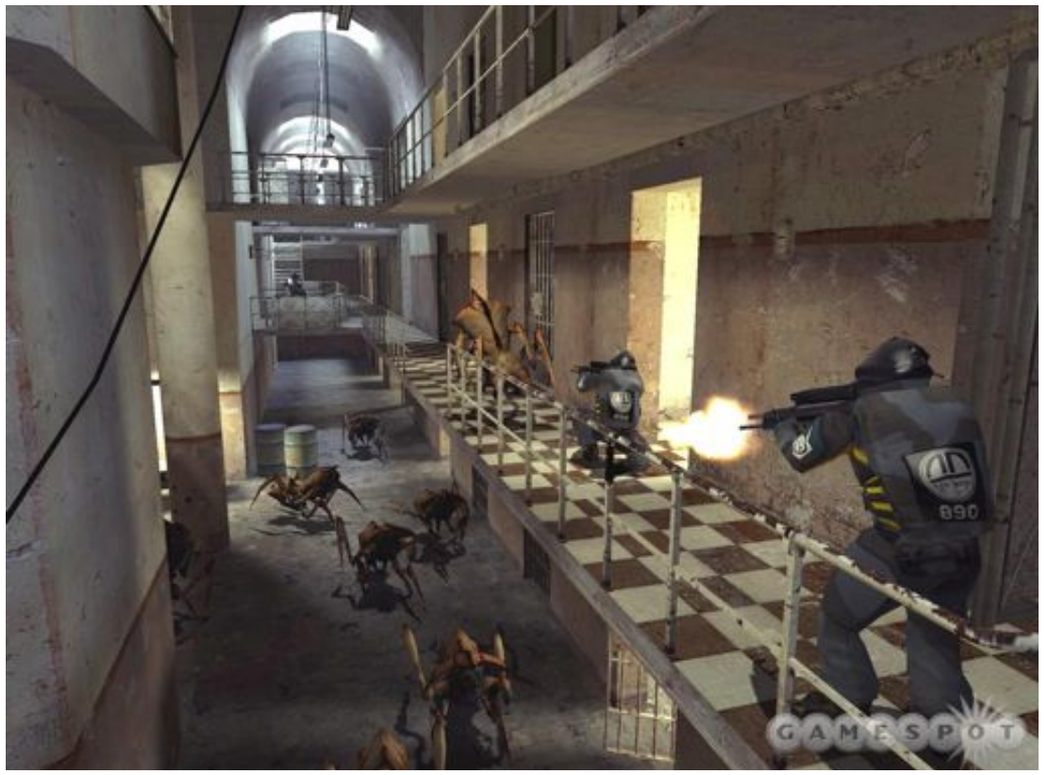


Let’s name some applications

Movies / Visual Effects (VFX)


THE HOBBIT
AN UNEXPECTED JOURNEY
DECEMBER 14, 2012

Video Games




ALAN
WAKE

Simulation

CAD-CAM & Design, Advertising

Hyltom Design Studio

Architecture

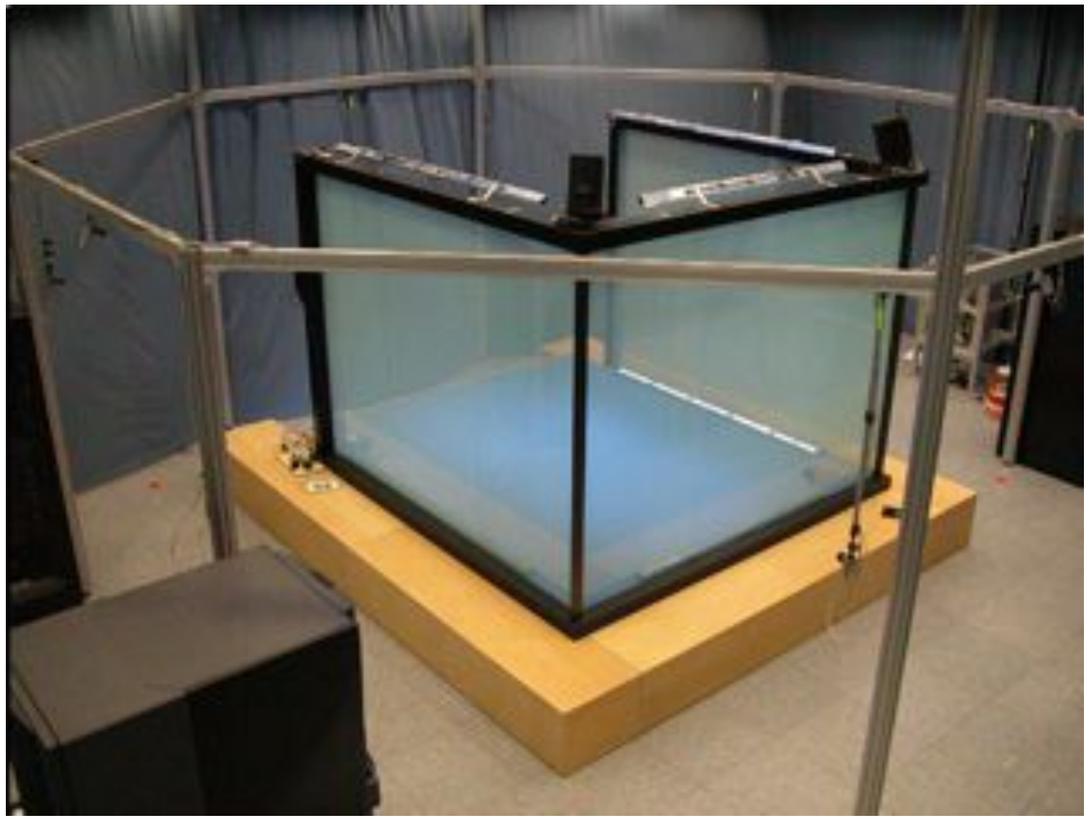
Global Illumination in Architecture

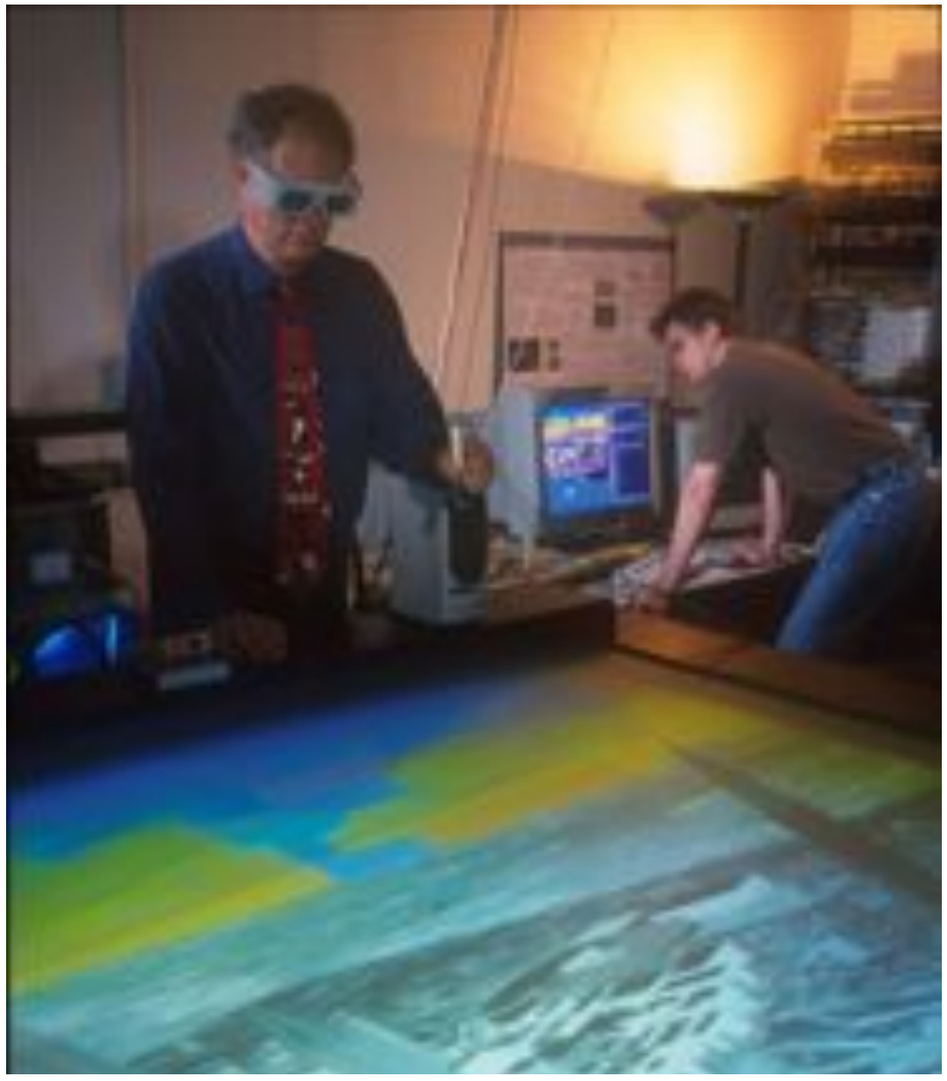
- The Light of Mies van der Rohe
  - by Henrik Wann Jensen, model by Stephen Duck

Global Illumination using Photon Mapping

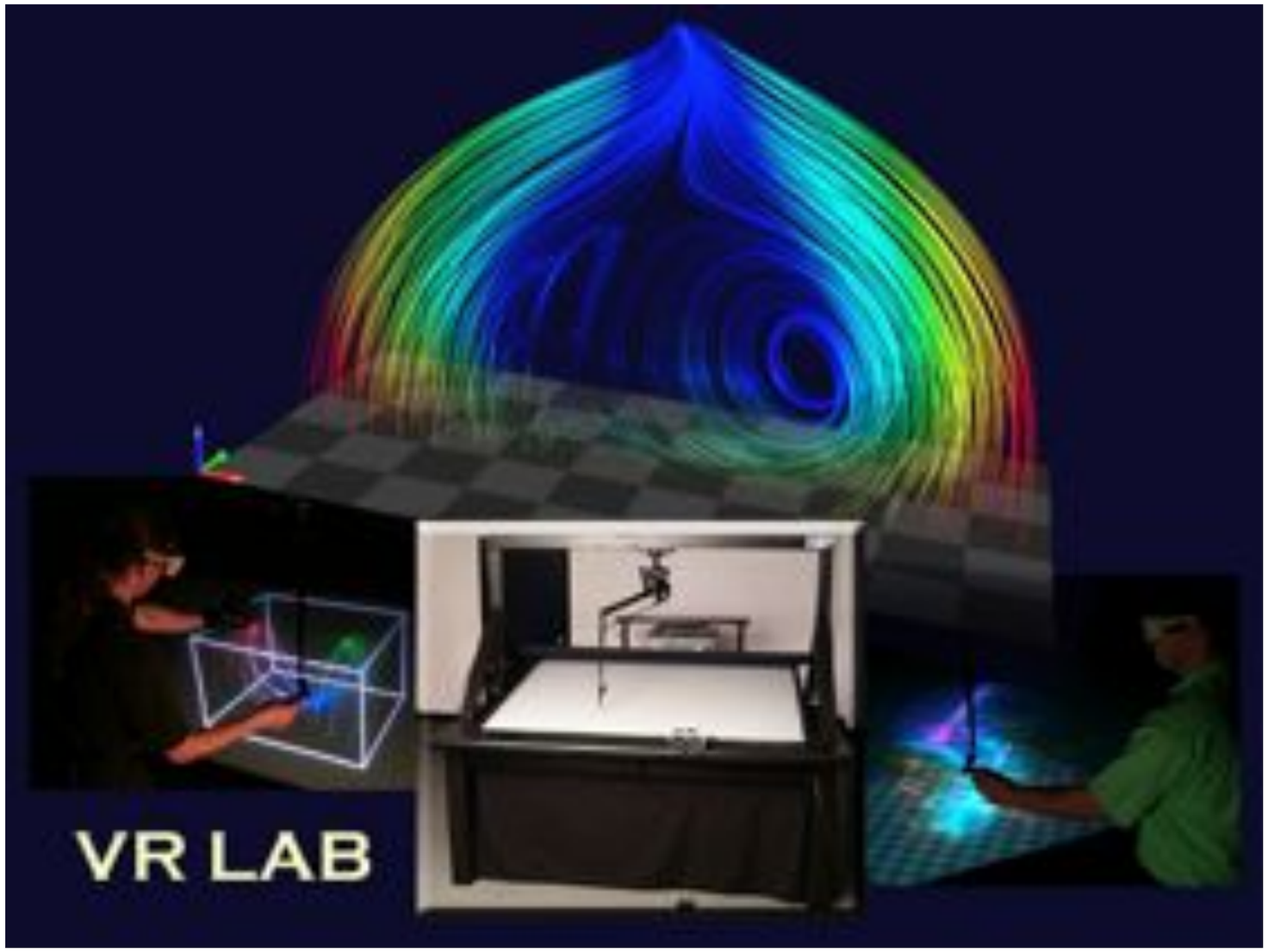
Henrik Wann Jensen

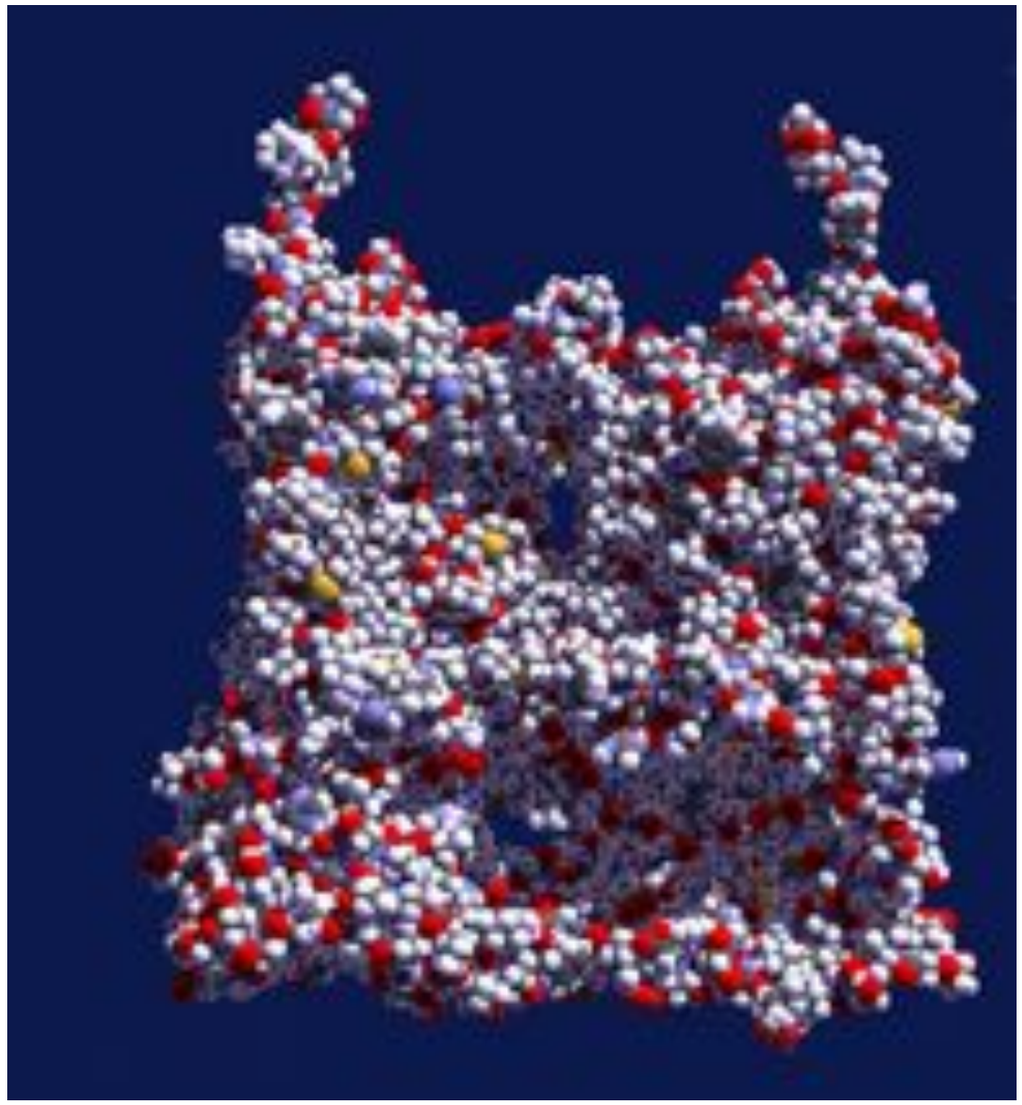
(Virtual Reality)



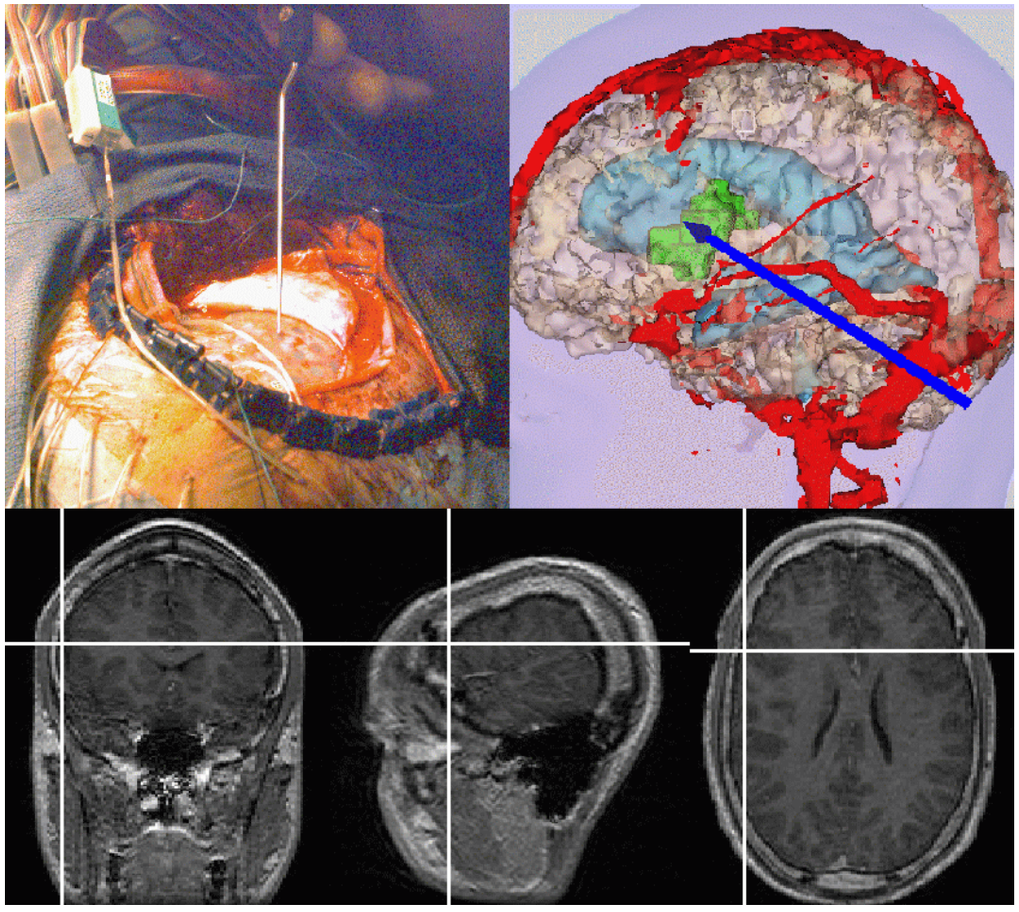


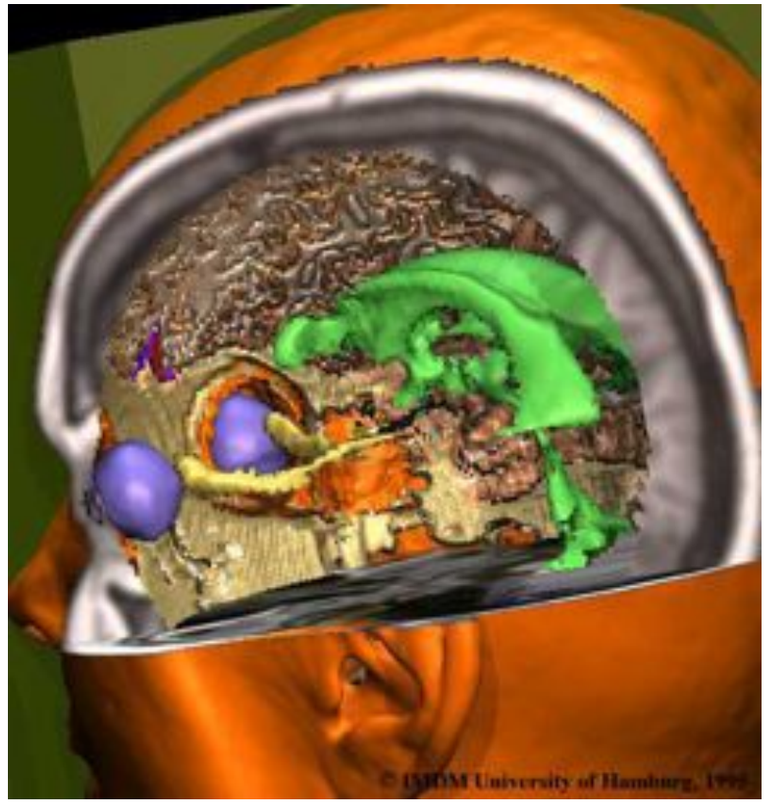
Scientific Visualization

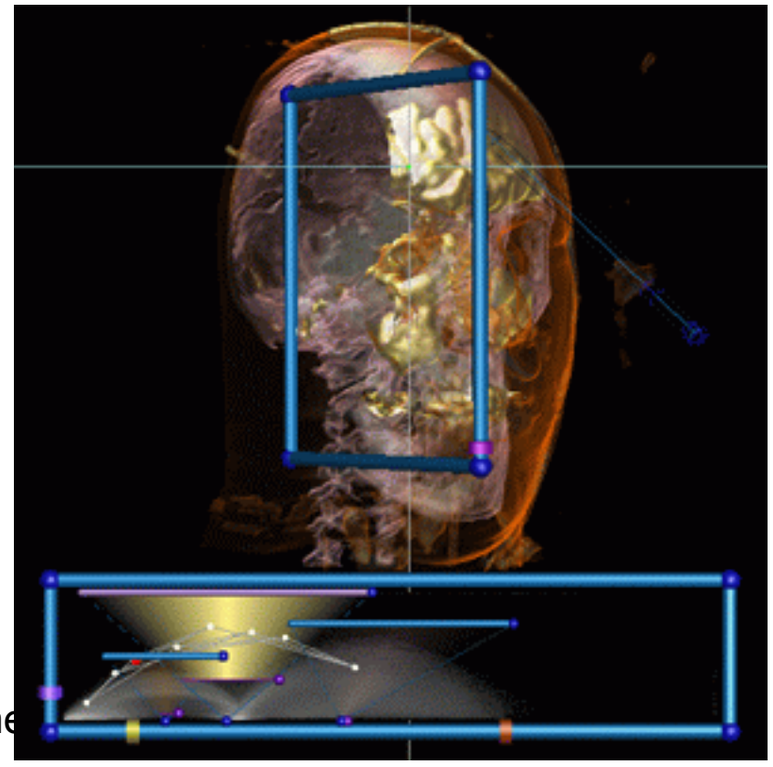




Medical Imaging







Differences between applications

- Games vs. movies?
- Architectural vs. medical visualization?

Differences between applications

- Games vs. movies?
  - Games need real-time performance (30 frames per second, FPS)
  - Movie frames usually take hours to render – each
- Architectural vs. medical visualization?
  - Building visualization needs accurate simulation of illumination (want to know what it'll look like)
  - Medical visualization aims to highlight important features in real-world datasets
- Etc. etc. etc.

What you will learn in T-111.4310

- Fundamentals of computer graphics algorithms
  - Will give a pretty good idea of how to implement lots of the things just shown
- We'll concentrate on 3D,
  not 2D illustration or image processing
- Basics of real-time rendering
  and graphics hardware
- Basic OpenGL
  - Not the focus, though: Means, not the end.
- You will get C++ programming experience
  - Most things are written in it IRL

What you will NOT learn

- OpenGL and DirectX hacks
  - Most become obsolete every 18 months anyway!

- Software packages
  - CAD-CAM, 3D Studio MAX, Maya
  - Photoshop and other painting tools

- Artistic skills
- Game design

How much Math?

- Lots of simple linear algebra
  - Get it right, it will help you a lot!
- Some more advanced concepts
  - Homogeneous coordinates
  - Quaternions for interpolating rotations/orientations
  - Ordinary differential equations (ODEs) and their numerical solution
  - Sampling, antialiasing (some gentle Fourier analysis)
  - Monte-Carlo integration
- Always in a concrete and visual context
- Deeper mathematic exposition in advanced class

Prof Background

- (Navel gazing)


REMEDY


NVIDIA


A!
Aalto University
School of Science

MAX PAYNE 2

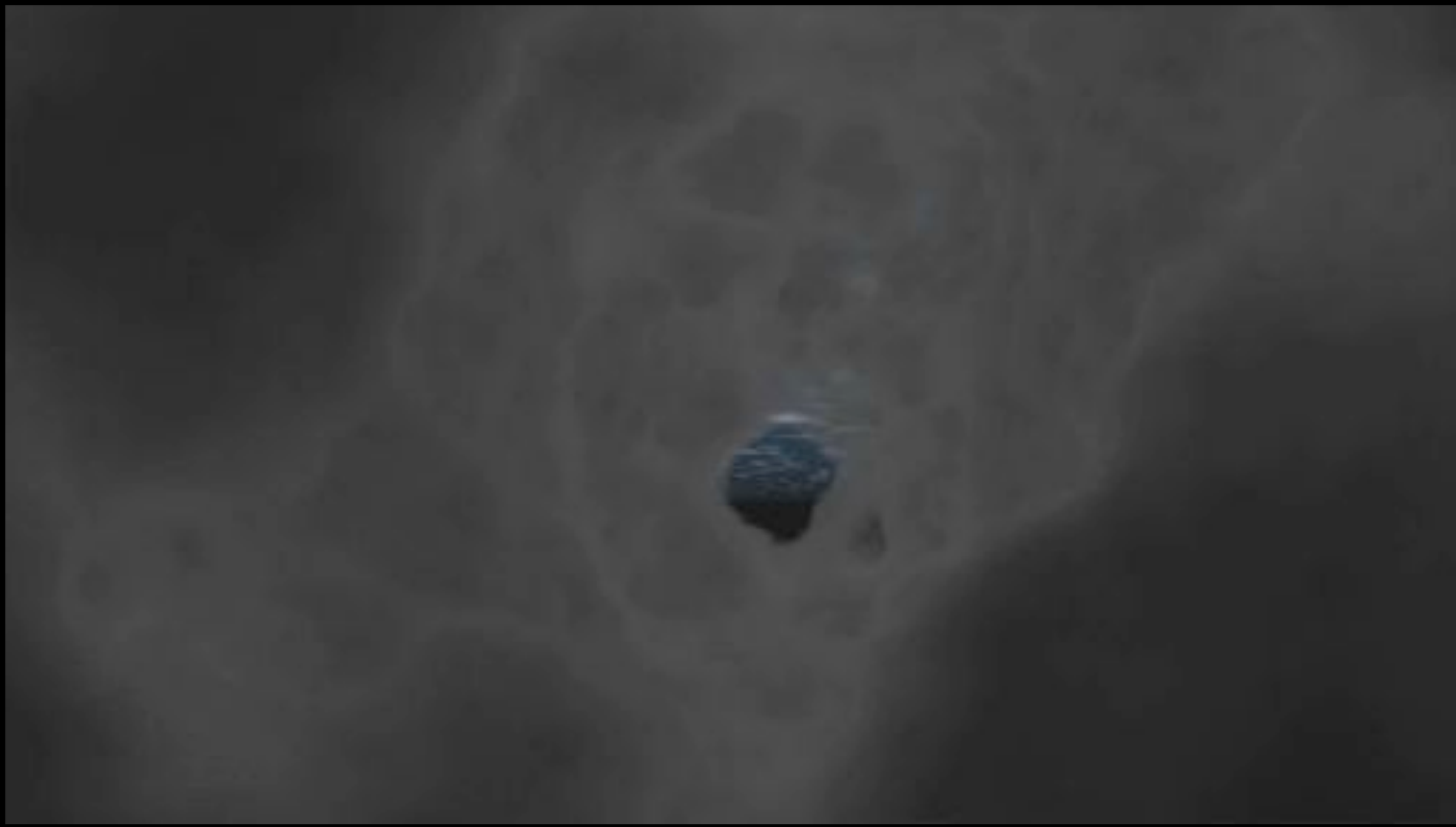
THE FALL OF MAX PAYNE

Prof Background

- (Navel gazing)

- I taught this class at MIT in 2009, now at Aalto

Fireflies by Blobtrox – Real Time, 4kb code+data (!!)



1st place in Assembly 2012 4k intro compo

code: MakeGho music: Gwaur

What was Going on There?

- Monte Carlo solution of the integrodifferential equation that governs light transport in a participating medium ("*volume rendering equation*")

What was Going on There?

- Monte Carlo solution of the integrodifferential equation that governs light transport in a participating medium ("*volume rendering equation*")

- Illustrates what I think is so damn cool about graphics – you can use math and algorithms to draw pretty pictures!

How do you make this picture?

Remedy Entertainment / Microsoft Games Studios

How do you make this picture?

How do you make this picture?

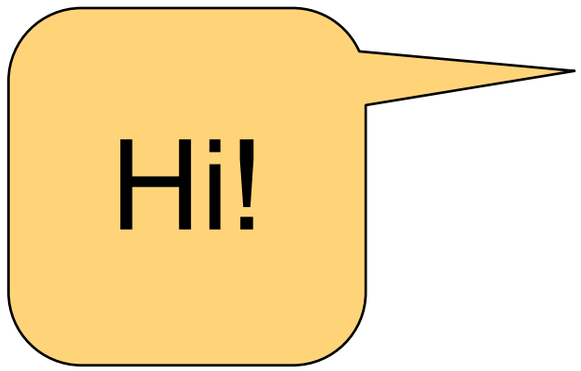
How do you make this picture?

How do you make this picture?

Modeling/Viewing Pipeline

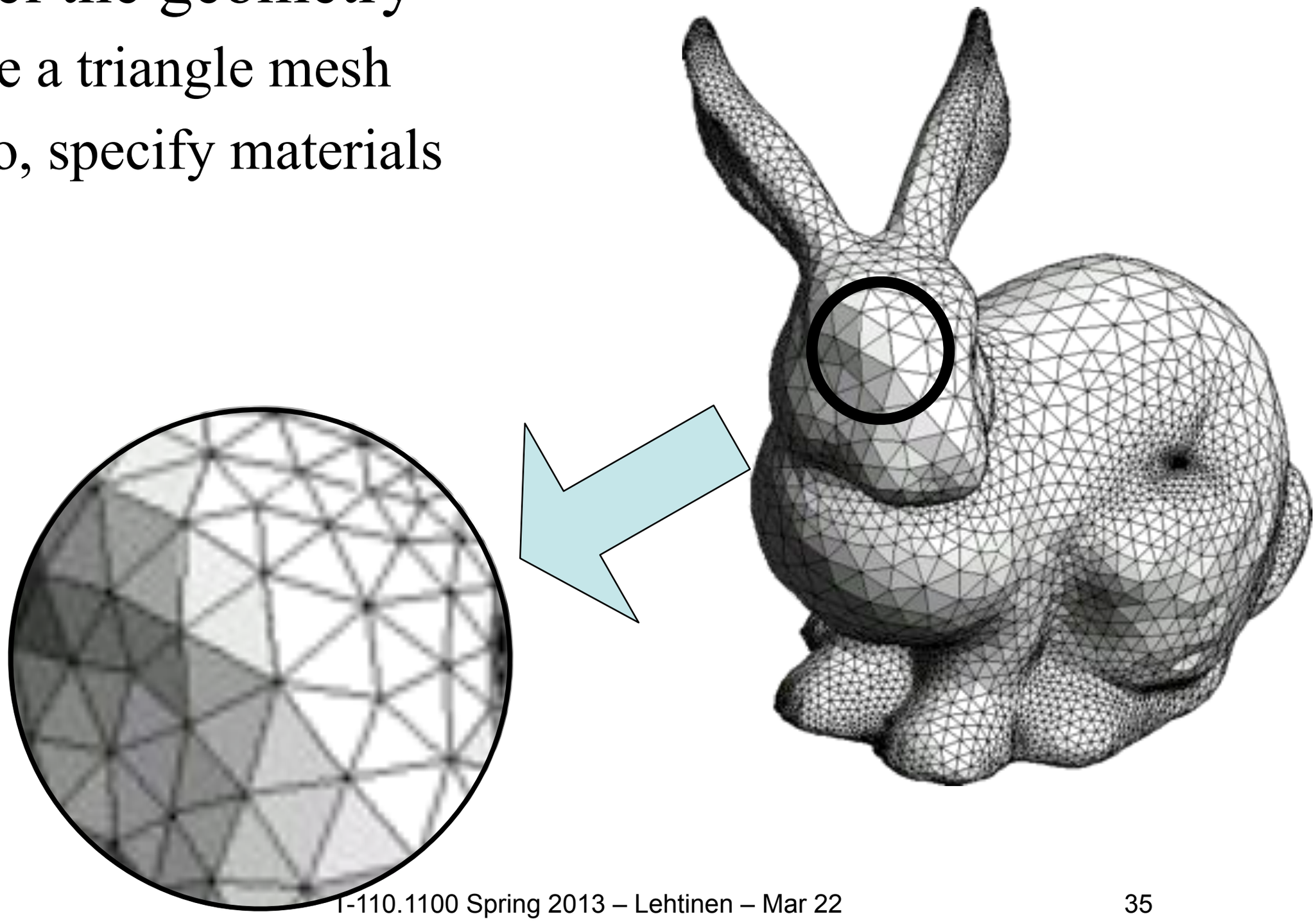
Meet the **Stanford Bunny**. He is one of the best-known characters in graphics.

See http://www.cc.gatech.edu/~turk/bunny/bunny.html for history.



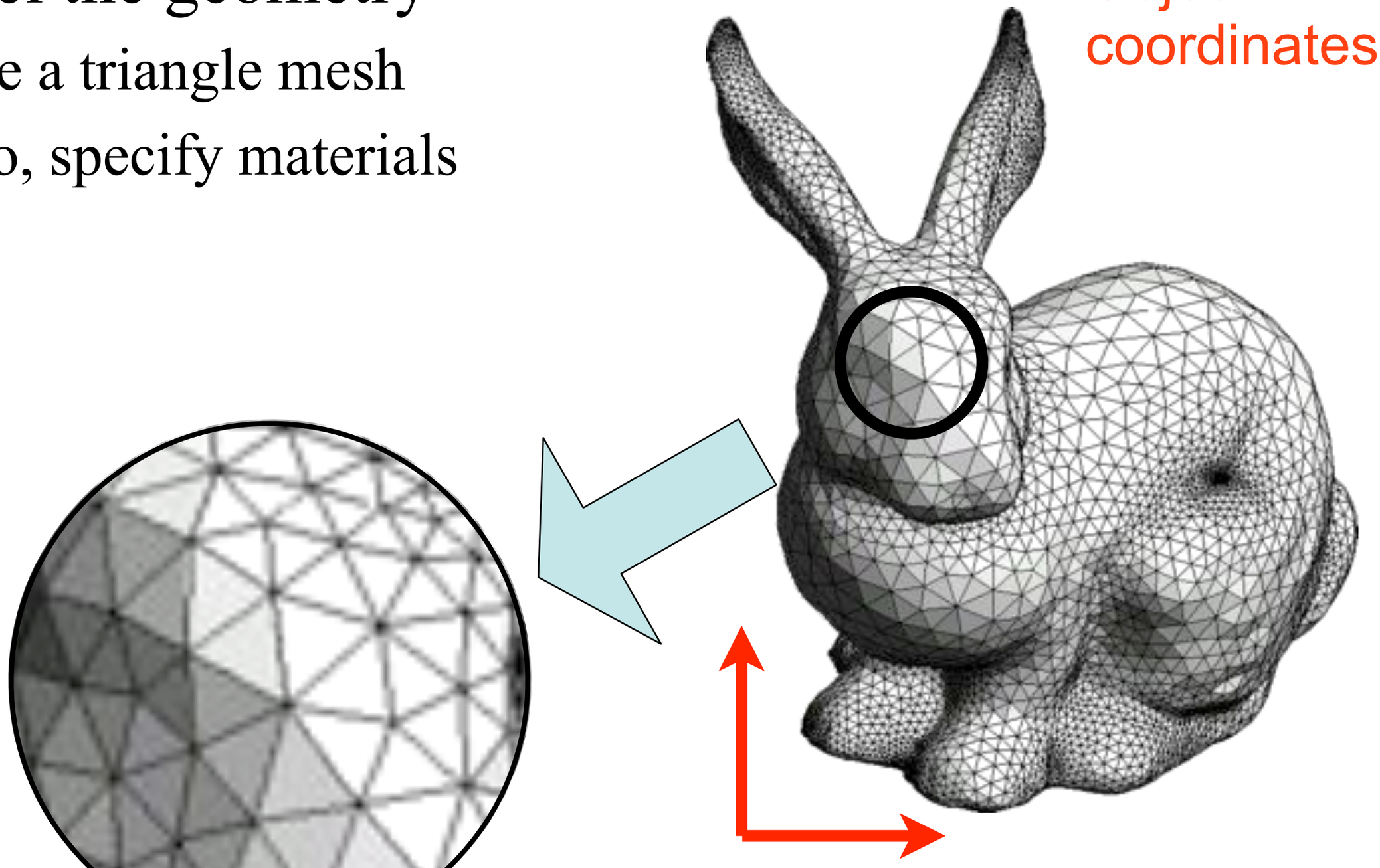
Modeling/Viewing Pipeline

1. Model the geometry
   - Here a triangle mesh
   - Also, specify materials



Modeling/Viewing Pipeline

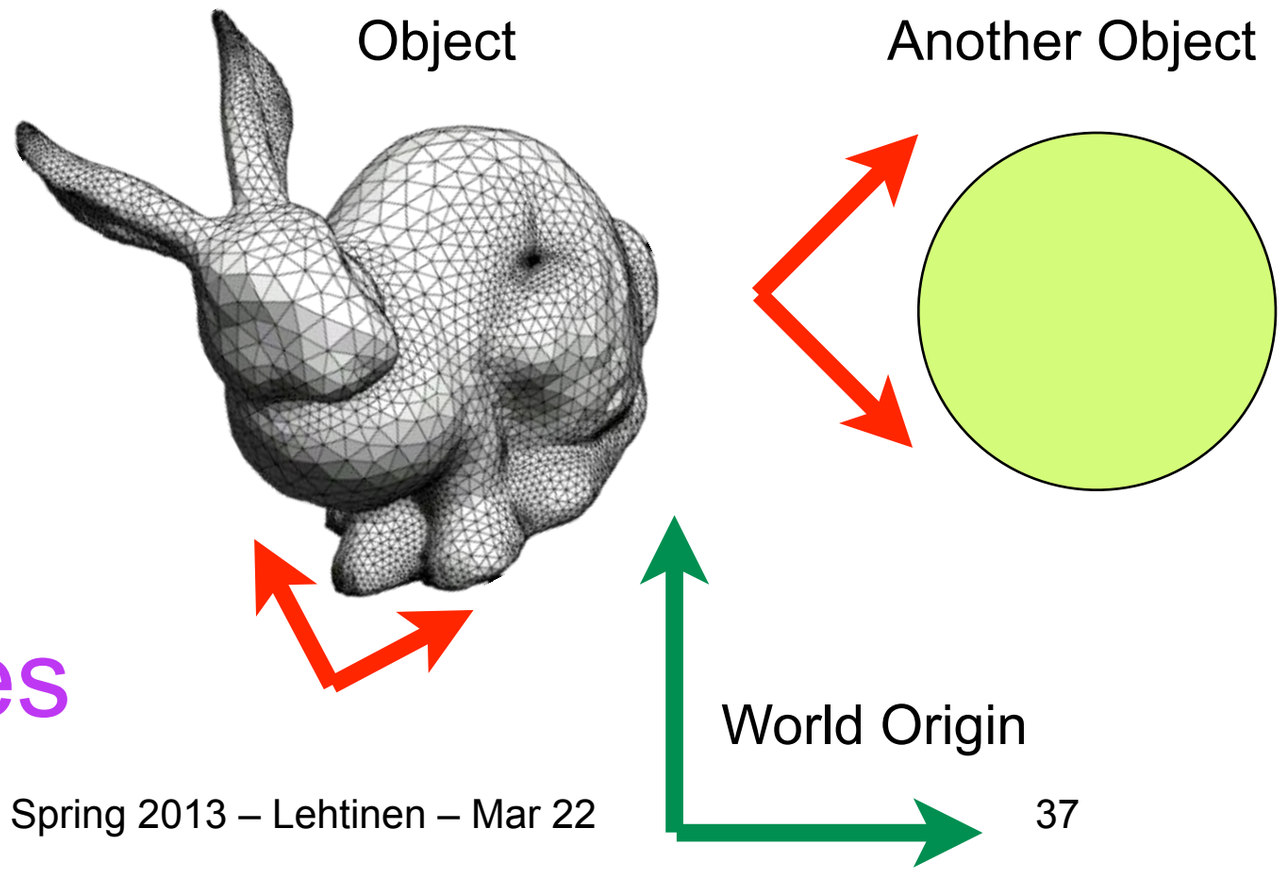
1. Model the geometry
   - Here a triangle mesh
   - Also, specify materials



Modeling/Viewing Pipeline

1. Model the geometry
2. Place the objects in **world space**
   - Each object has its own **object space**
   - Only one world space

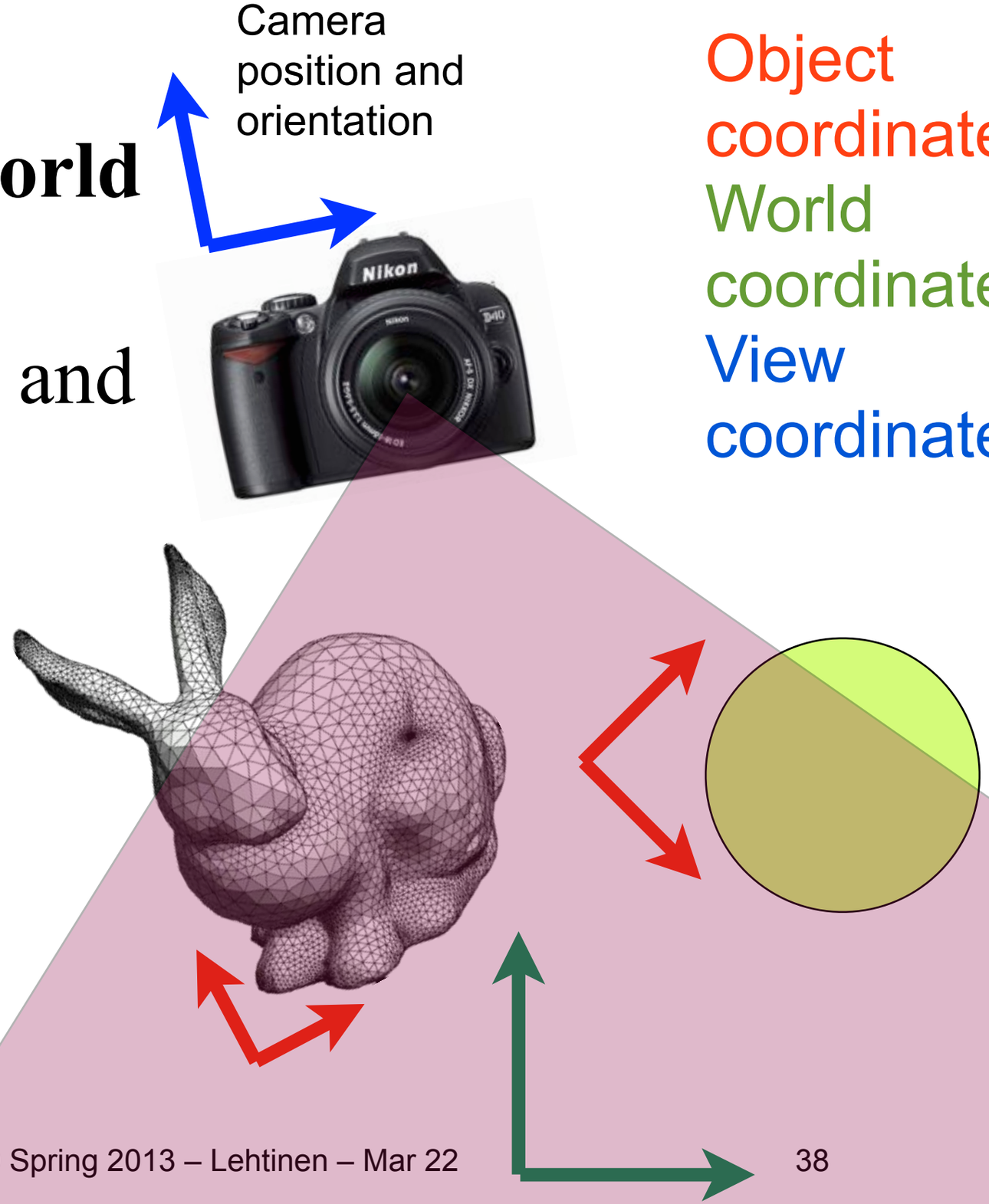
Object coordinates
World coordinates



Space ⇔ Coordinates

Modeling/Viewing Pipeline

1. Model the geometry
2. Place the objects in **world space**
3. Pick viewing position and direction



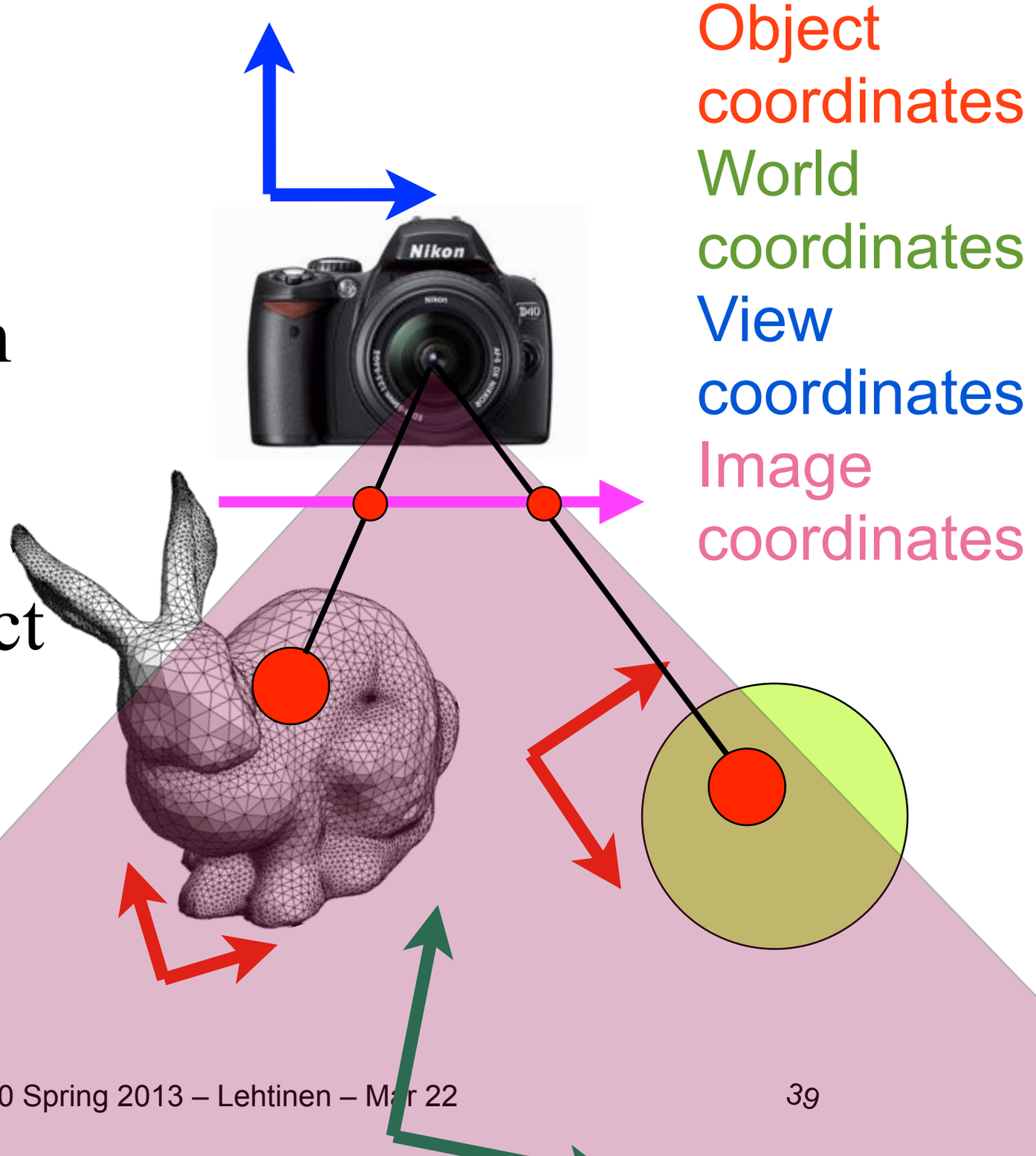
Object coordinates

World coordinates

View coordinates

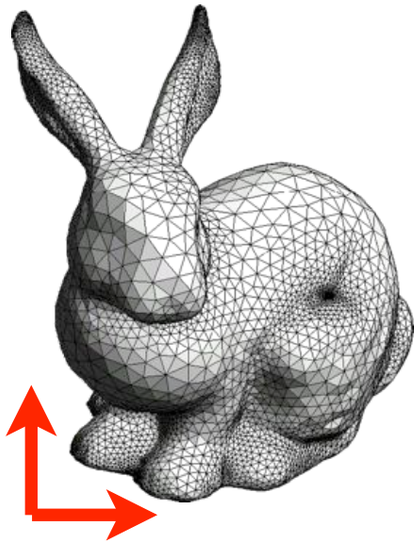
Modeling/Viewing Pipeline

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!

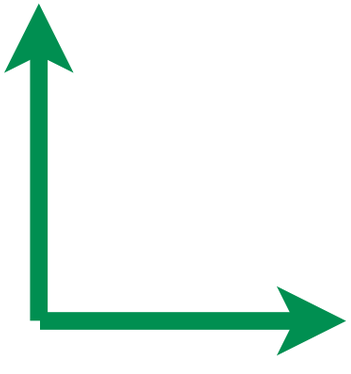


Modeling/Viewing Summary

Object coordinates



World coordinates



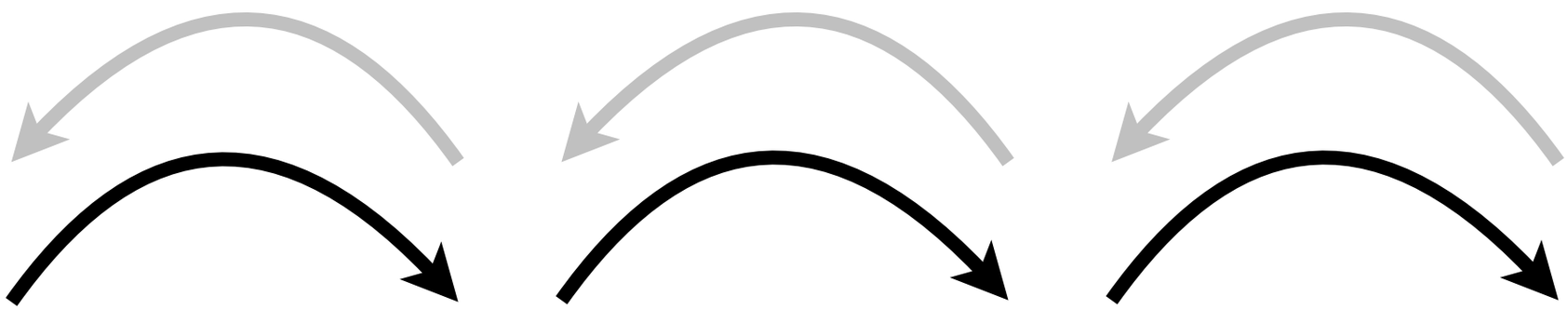
View coordinates

Image coordinates

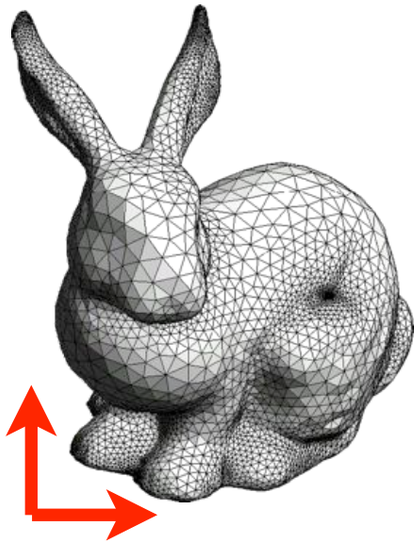
Emil Praun

Modeling/Viewing Summary

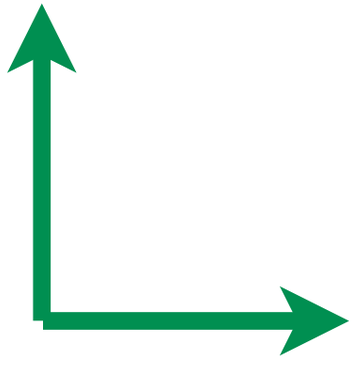
- Some algorithms go the same sequence in the opposite direction (e.g. ray tracing)



Object coordinates



World coordinates



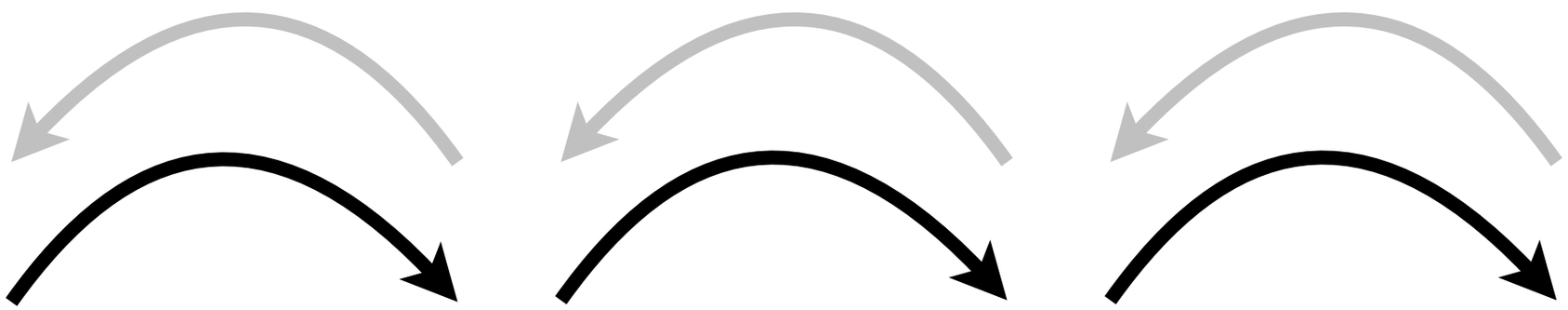
View coordinates

Image coordinates

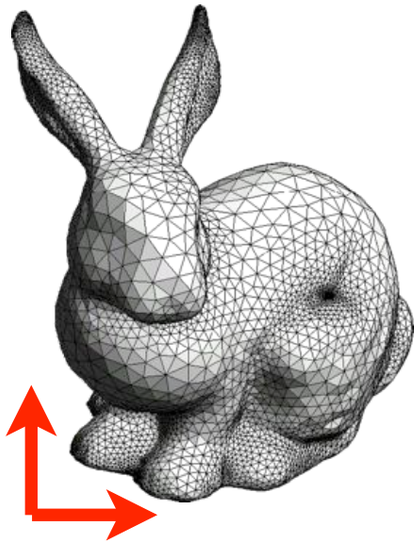
Emil Praun

Modeling/Viewing Summary

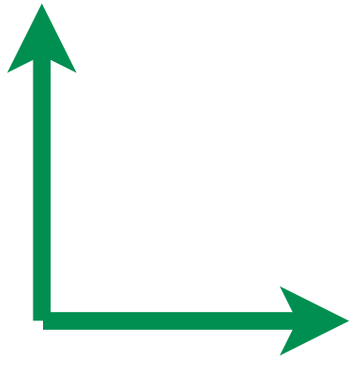
Animation:
Make these transformations vary with time



Object coordinates



World coordinates



View coordinates

Image coordinates

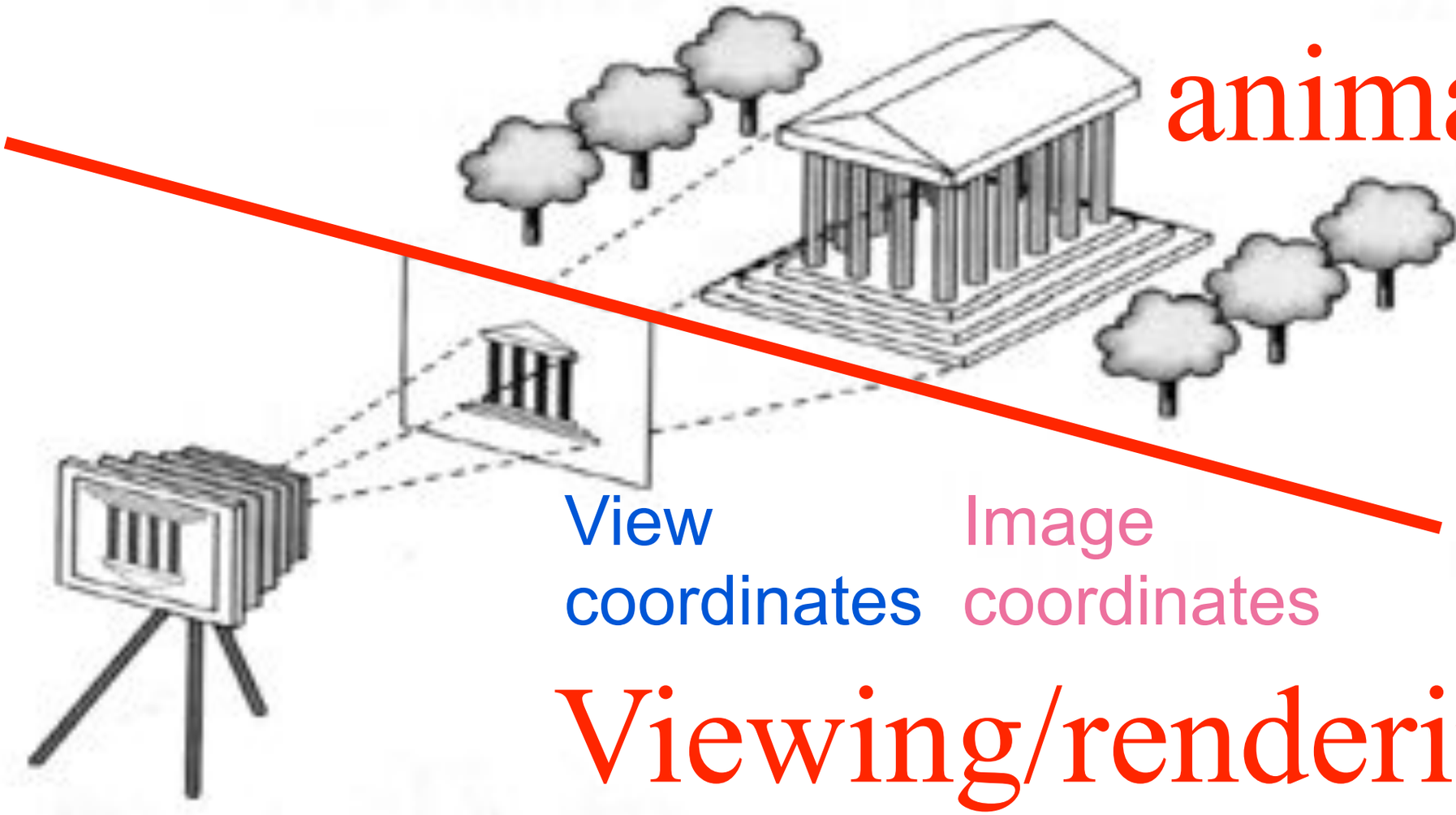
Emil Praun

Another View

Object coordinates

World coordinates

Modeling, animation



View coordinates

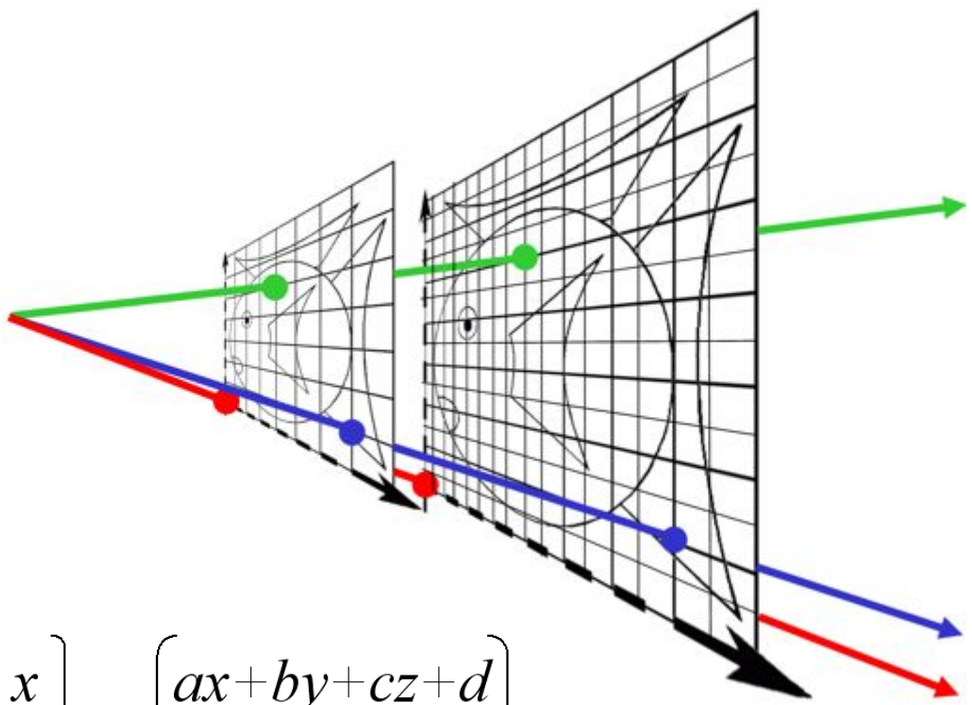
Image coordinates

Viewing/rendering

Questions?

Coordinate Transformations

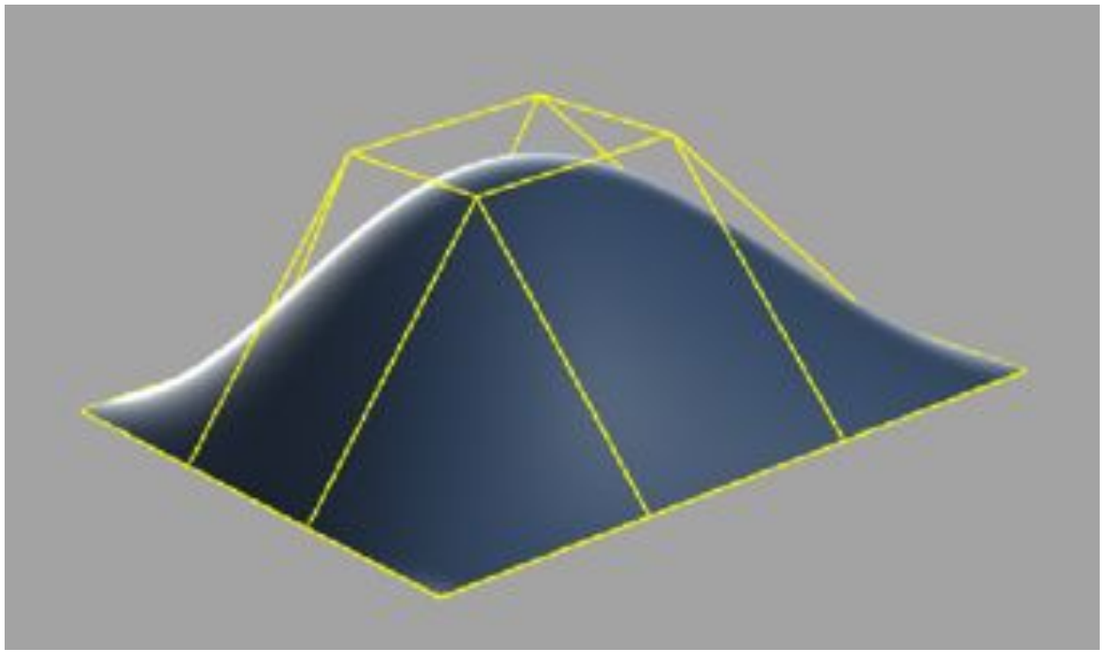
- Mostly linear algebra
- Homogeneous coordinates (remember your Kivelä!)
  - Neat way of treating affine and perspective transforms as linear
- Perspective (for viewing)

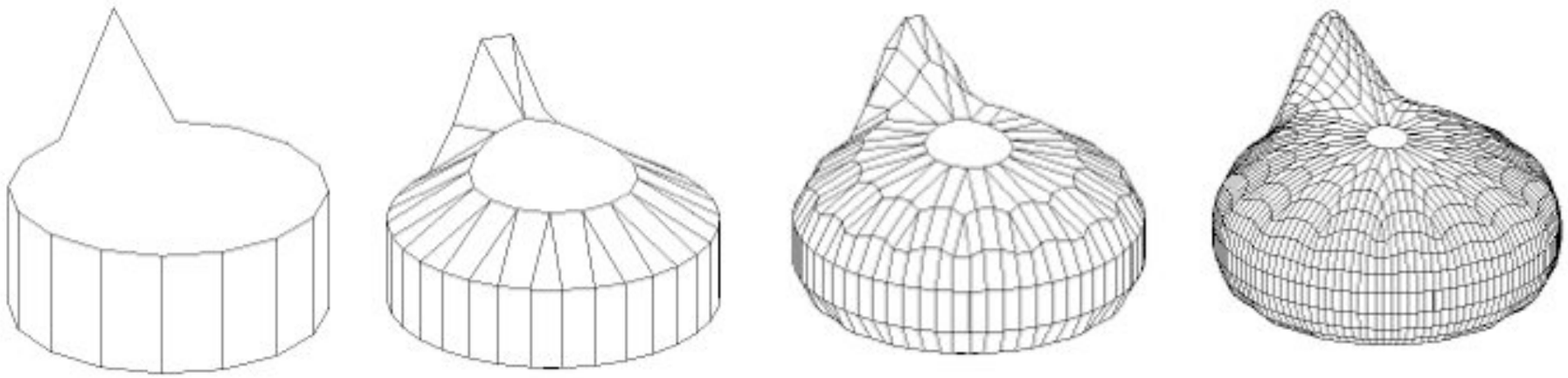


$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} ax+by+cz+d \\ ex+fy+gz+h \\ ix+jy+kz+l \\ 1 \end{bmatrix}$$

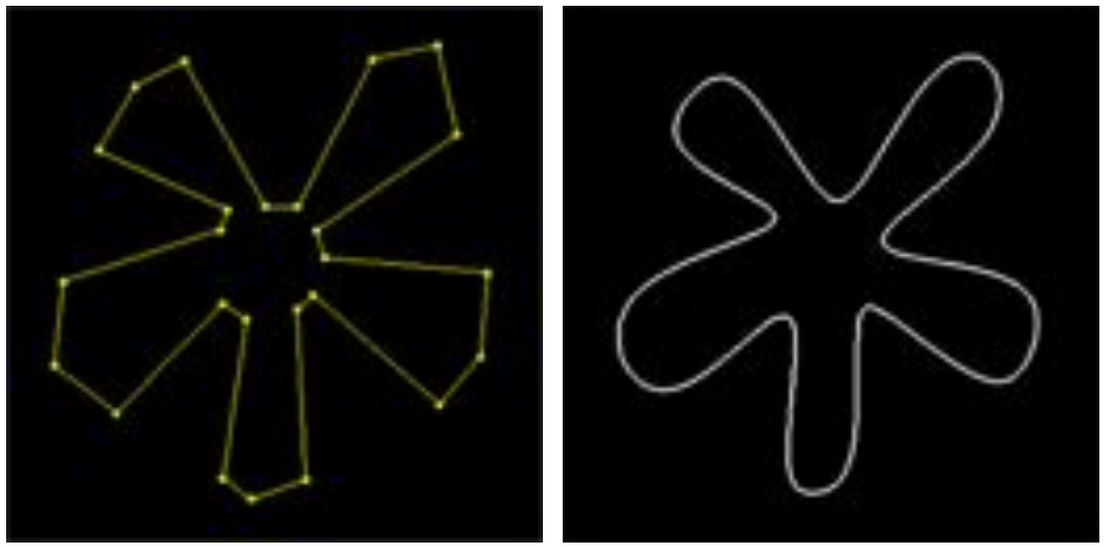
Modeling

- 2D curves, triangle meshes, smooth surfaces (Bézier, splines), subdivision surfaces



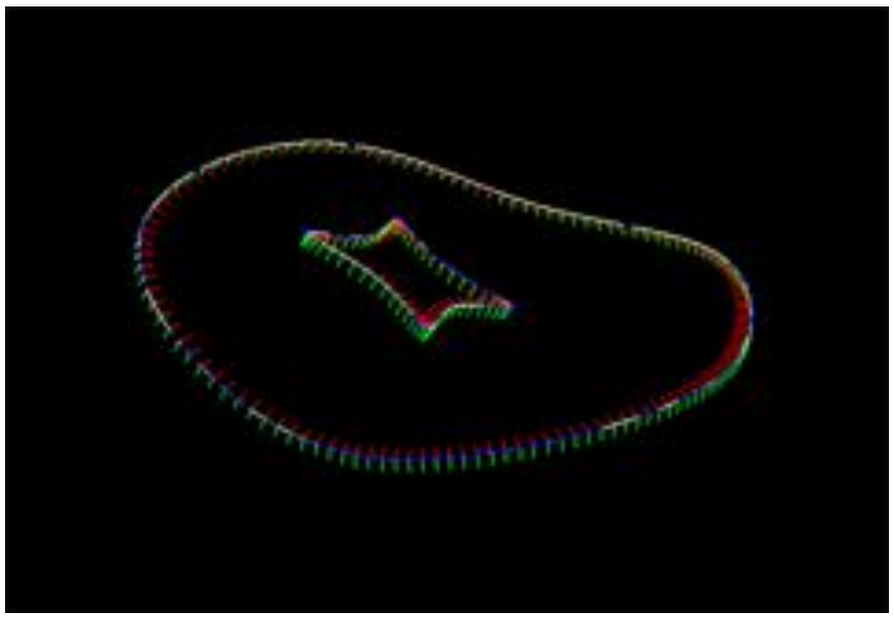


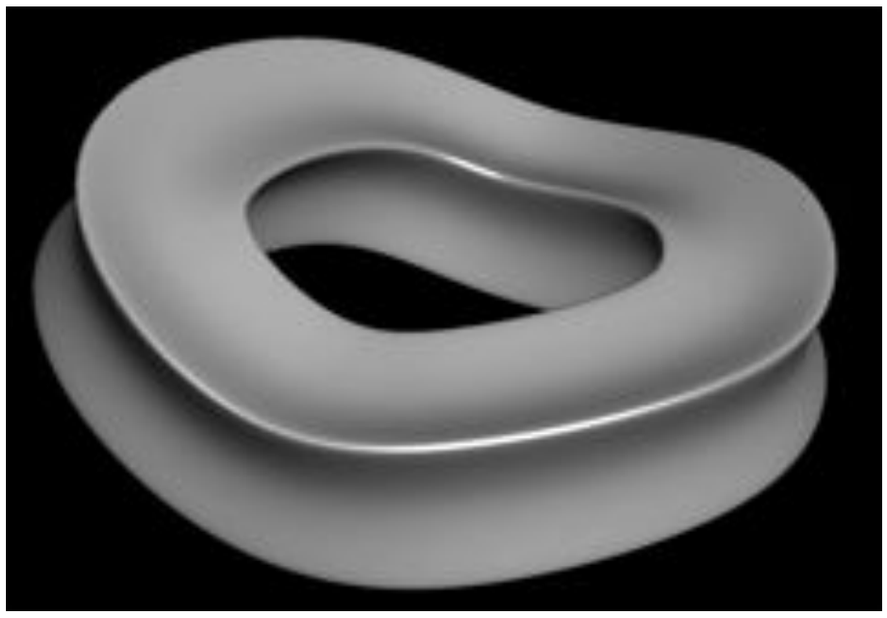
Assignment 1: Curves & Surfaces



Spline curves

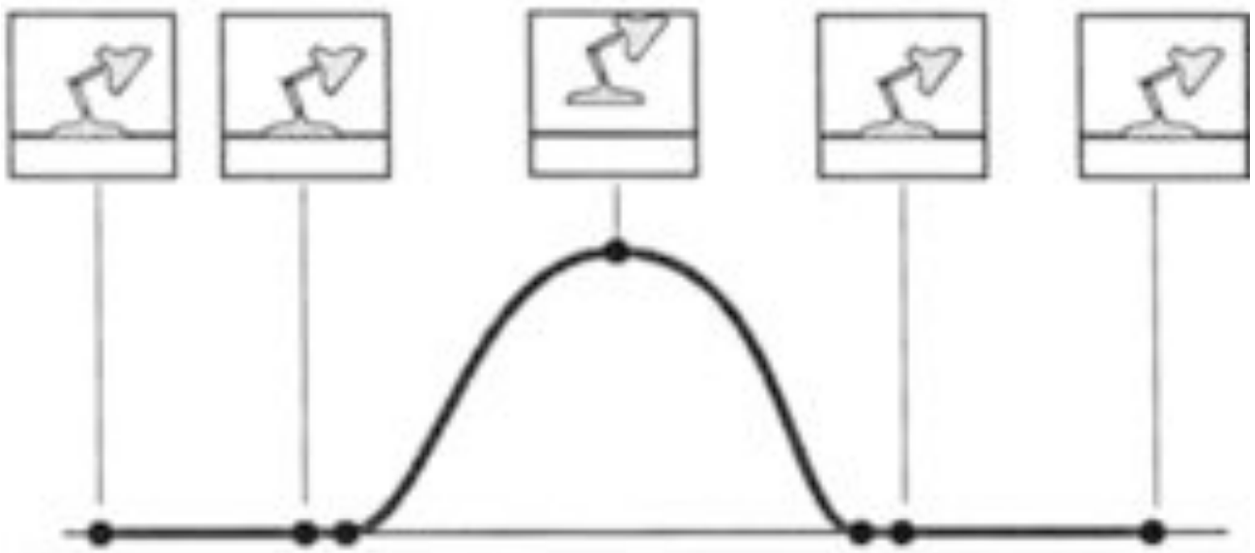
Surfaces of revolution



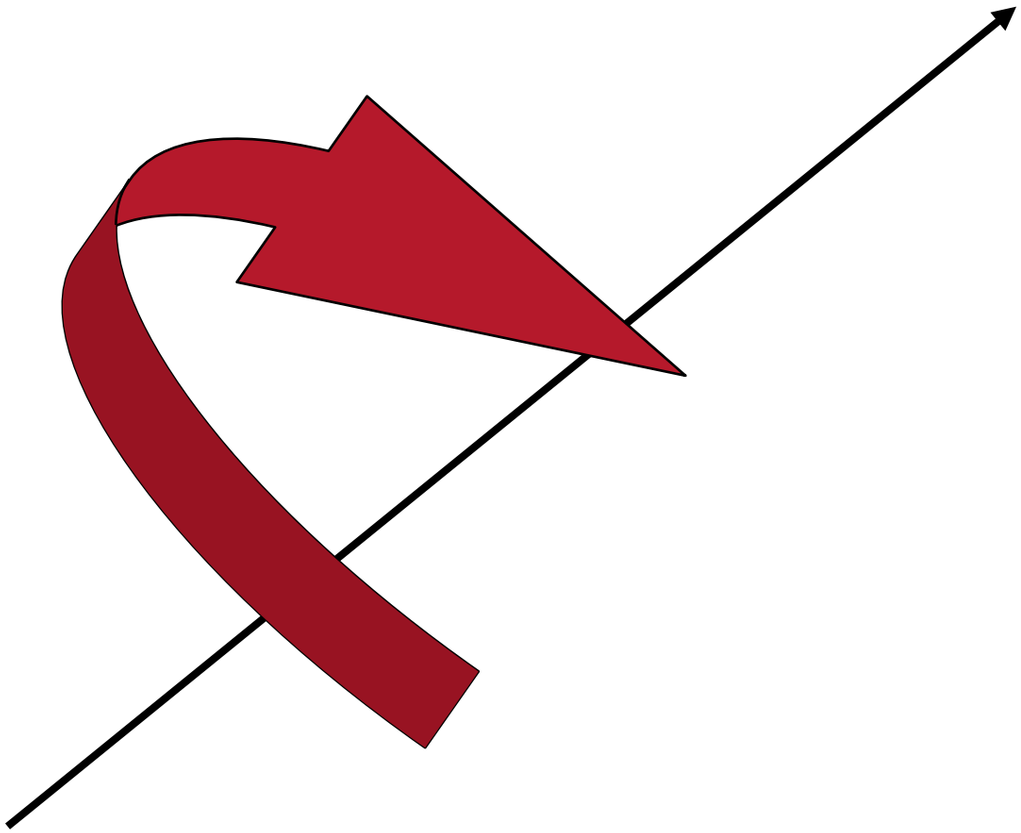


Sweep surfaces

Animation: Keyframing



ACM © 1987 "Principles of traditional animation applied to 3D computer animation"



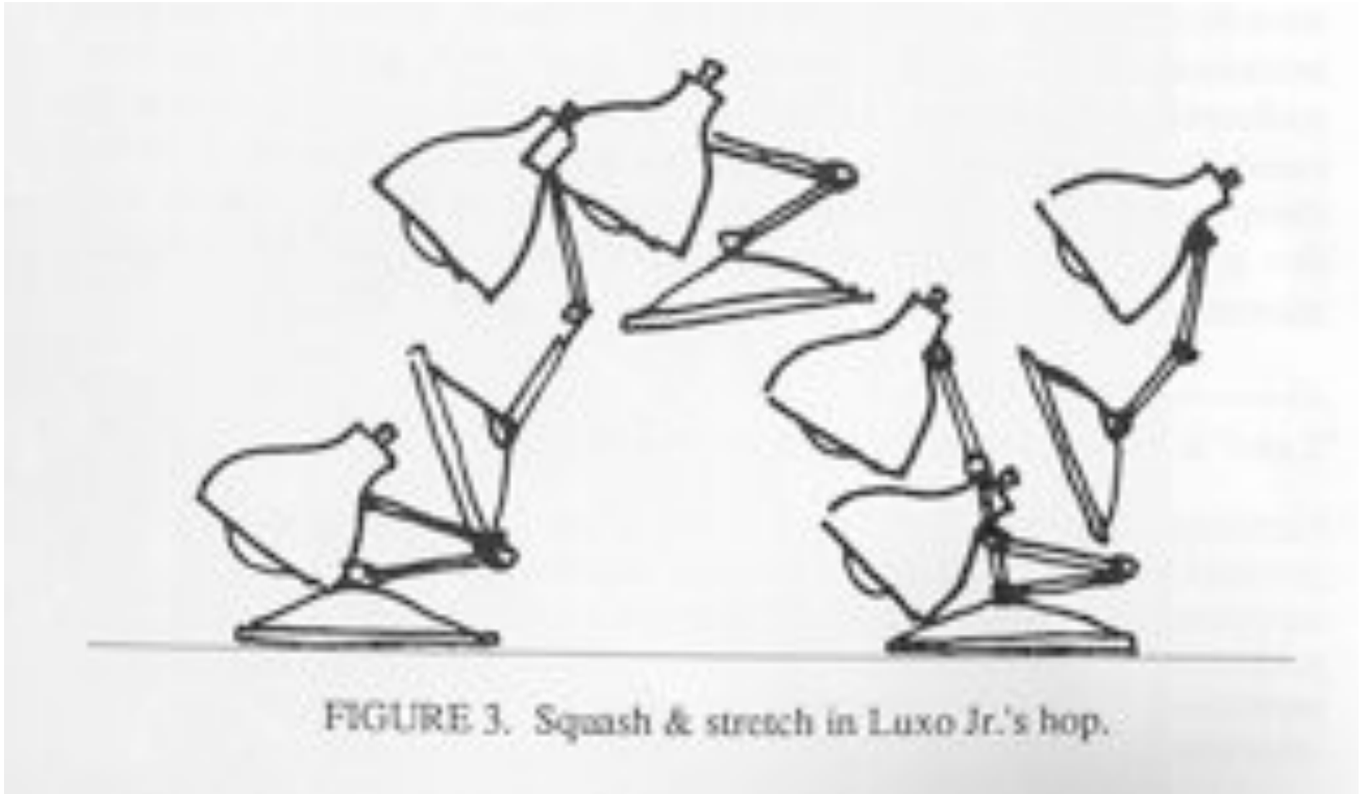
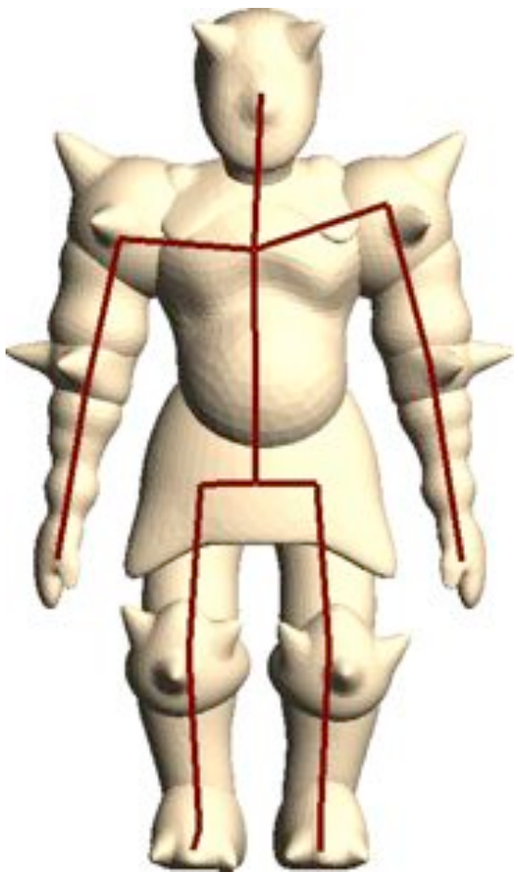


FIGURE 3. Squash & stretch in Luxo Jr.'s hop.

Character Animation: Skinning

- Animate simple “skeleton”
- Attach “skin” to skeleton
  - Skin deforms smoothly with skeleton
- Used everywhere (games, movies)



Ilya Baran

Epic Games

Pinocchio by Ilya Baran

- Automatic rigging, used in e.g. Blender

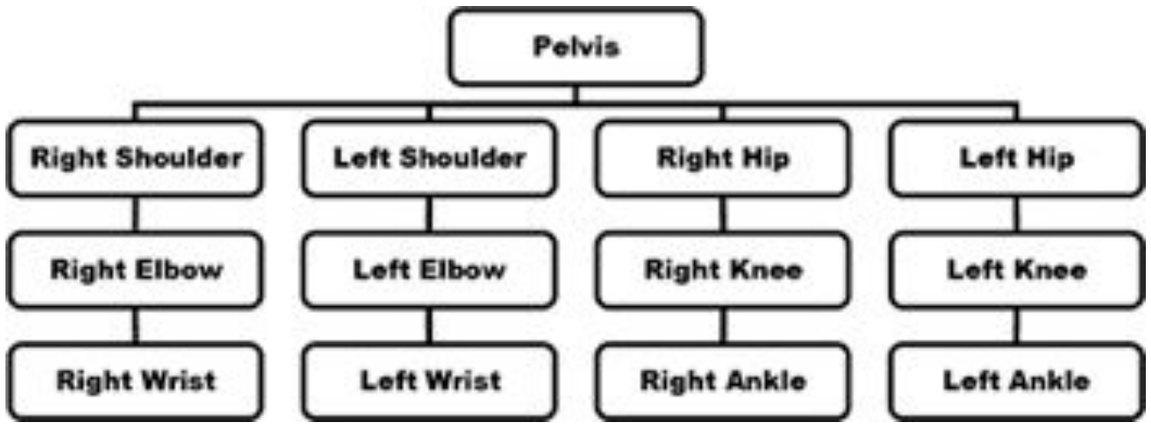
Pinocchio

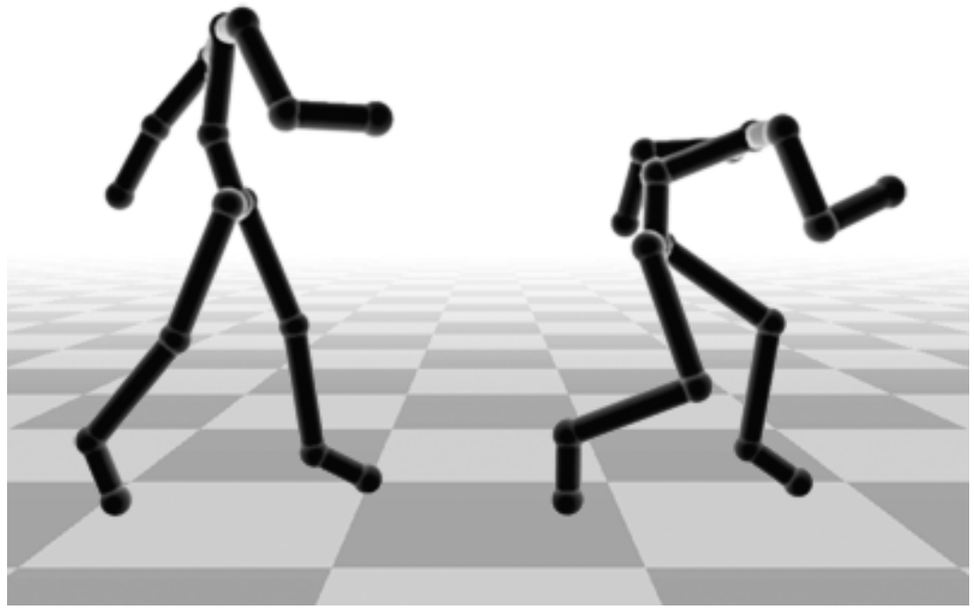
- This is an example of research done at universities
  - This particular one, at MIT in Boston

- See Ilya's SIGGRAPH 2007 paper here

Assignment 2: Hierarchical Modeling

- Animate character skeleton as tree of transformations

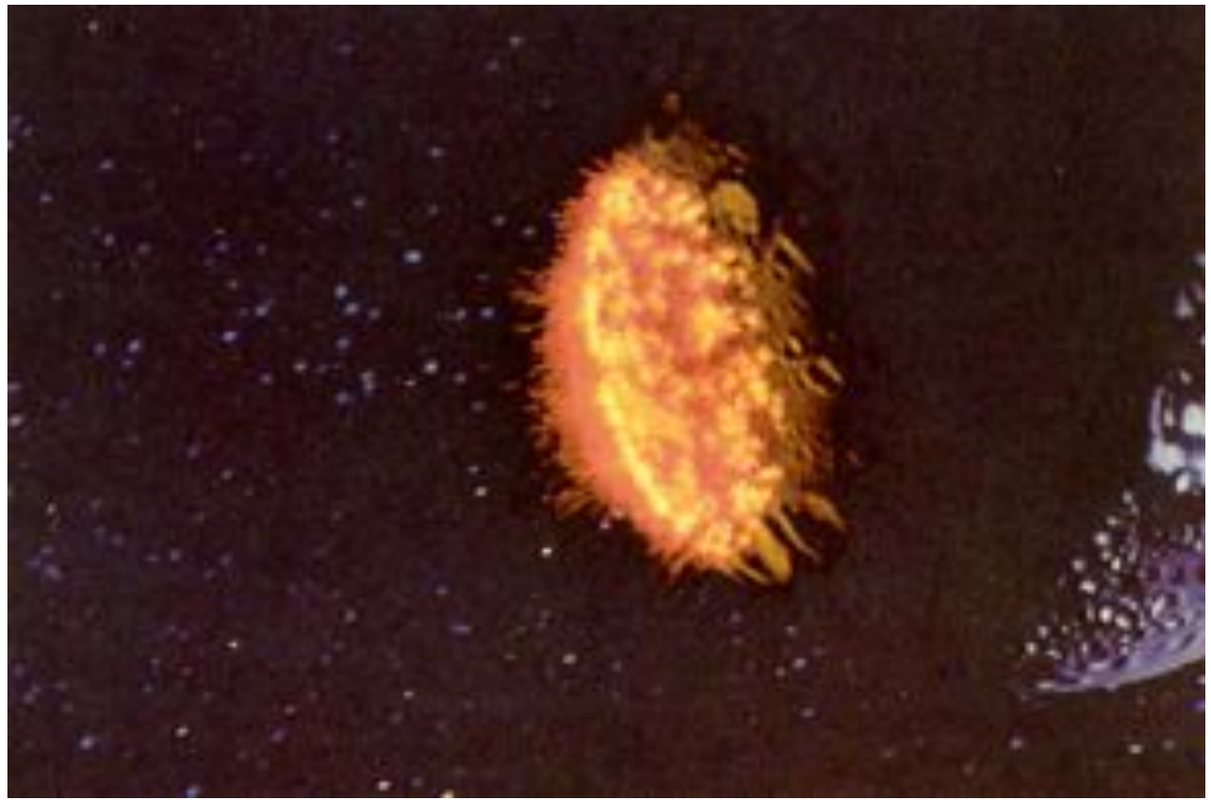


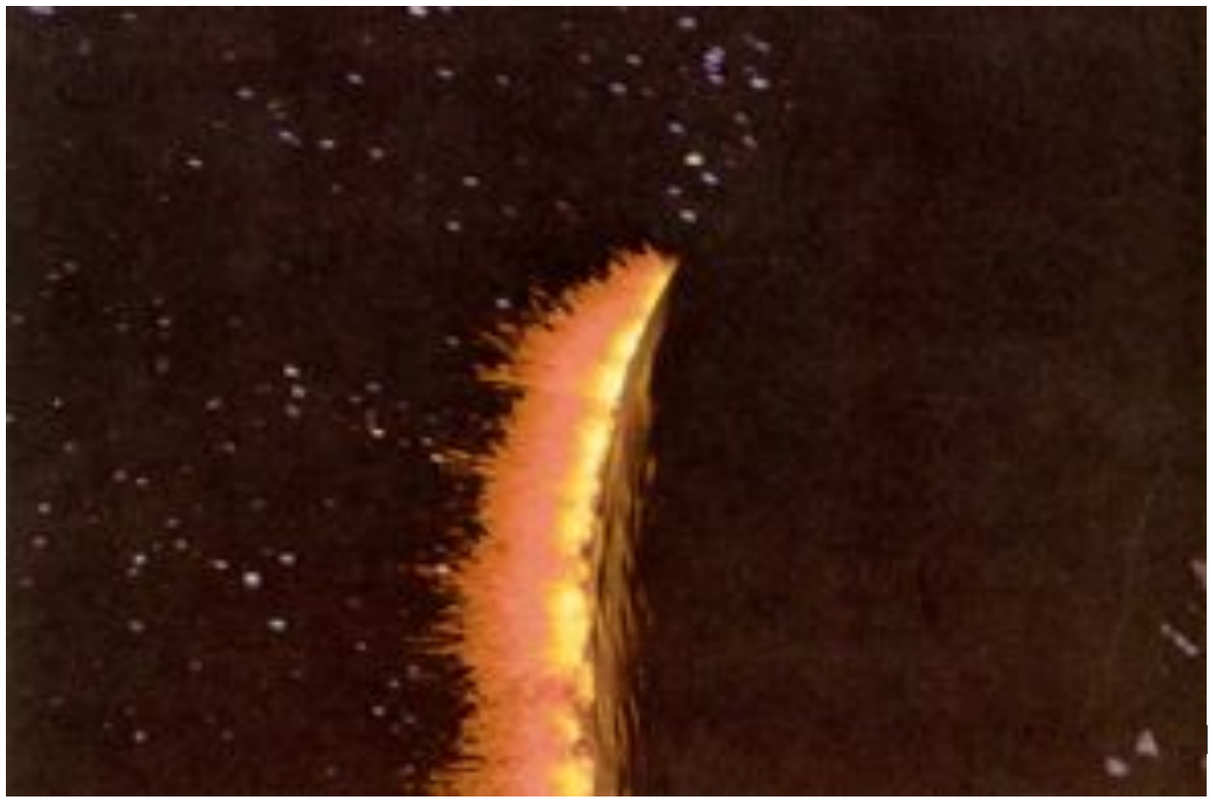


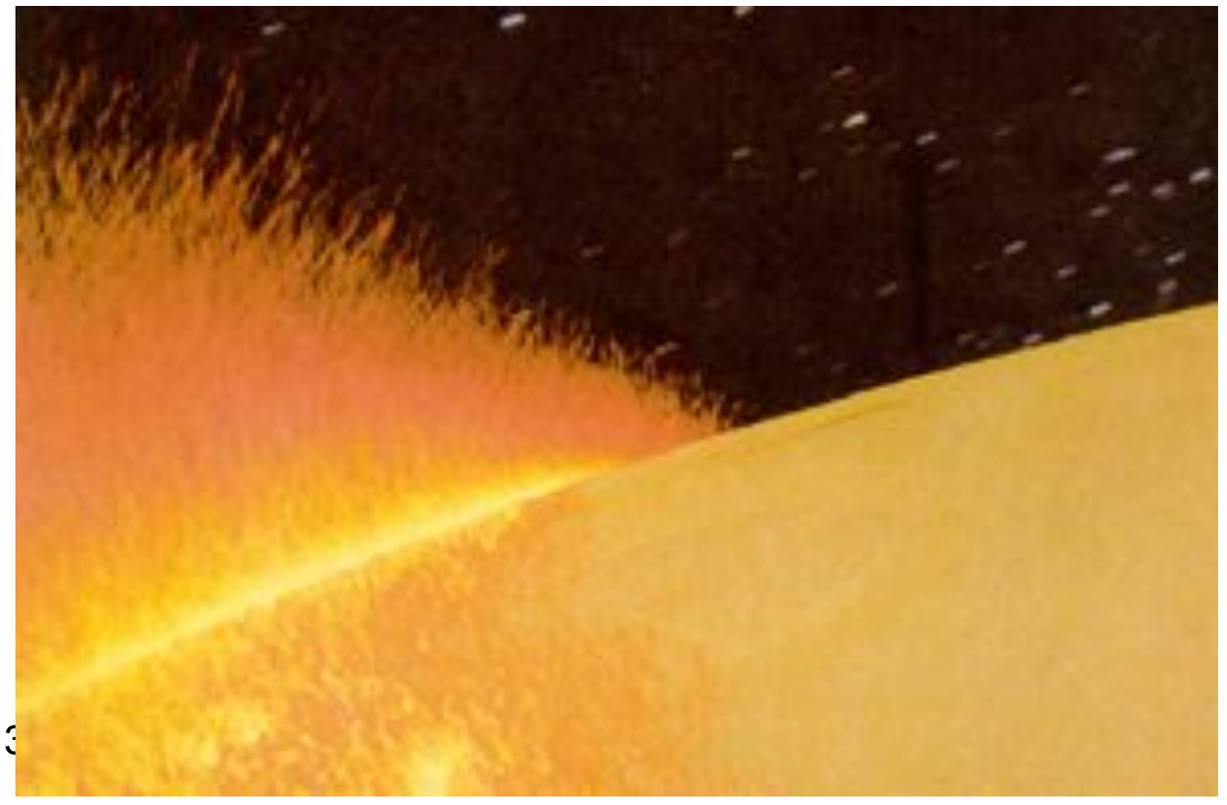
- "Skinning": smooth surface deformation based on animated skeleton

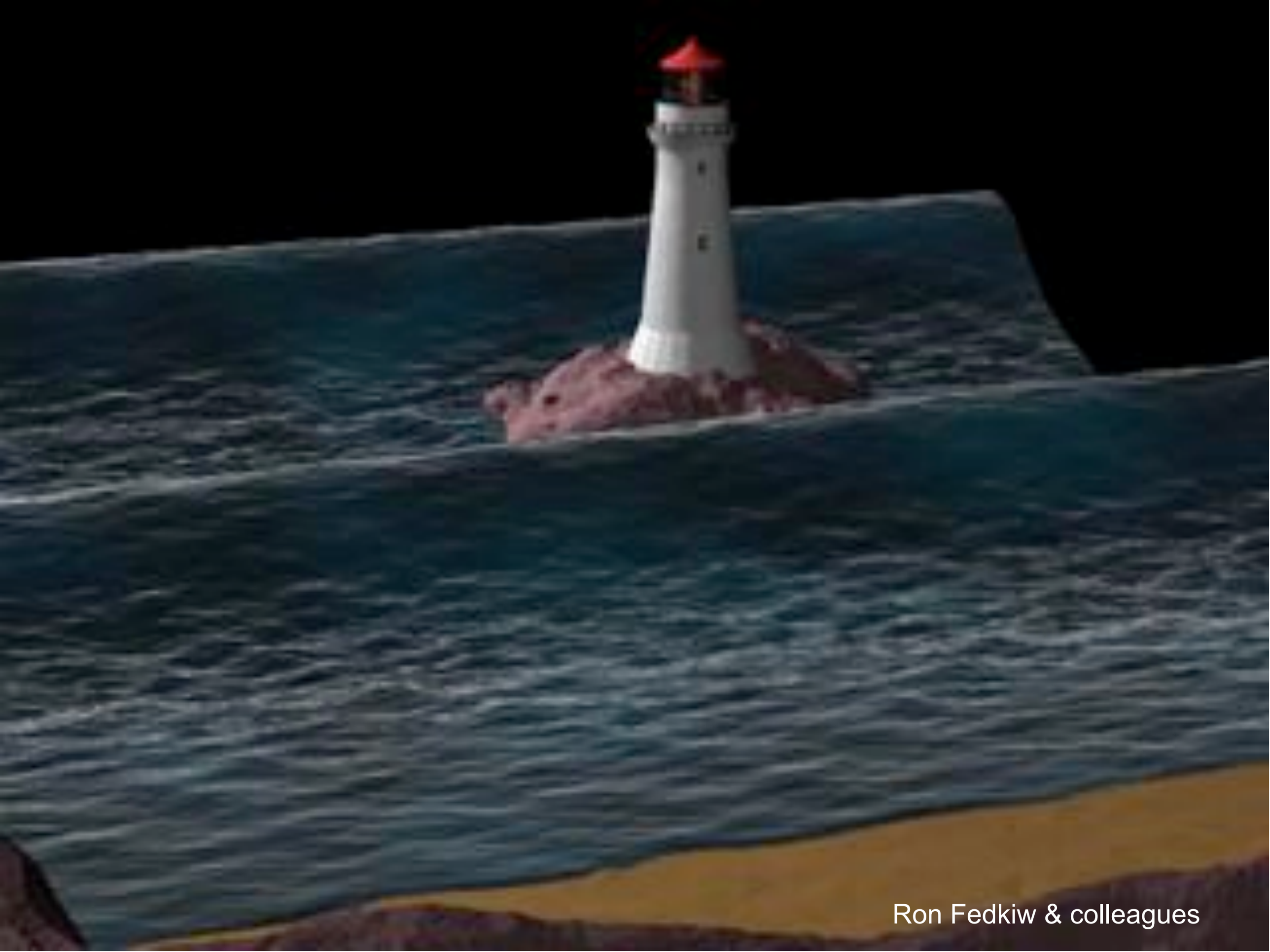
Particle systems (ODEs)

Star Trek 2





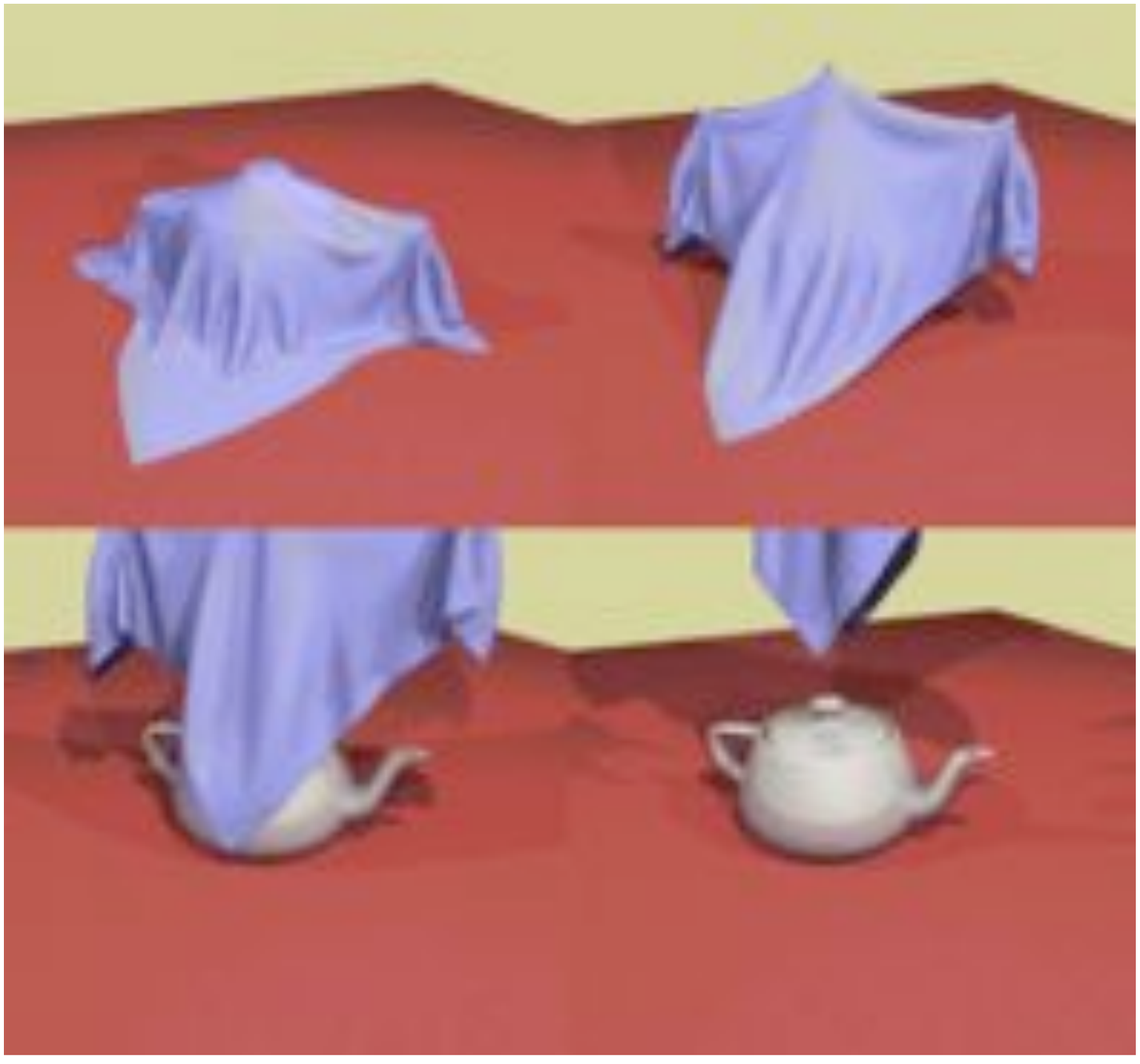



Ron Fedkiw & colleagues

“Physics” (ODEs)

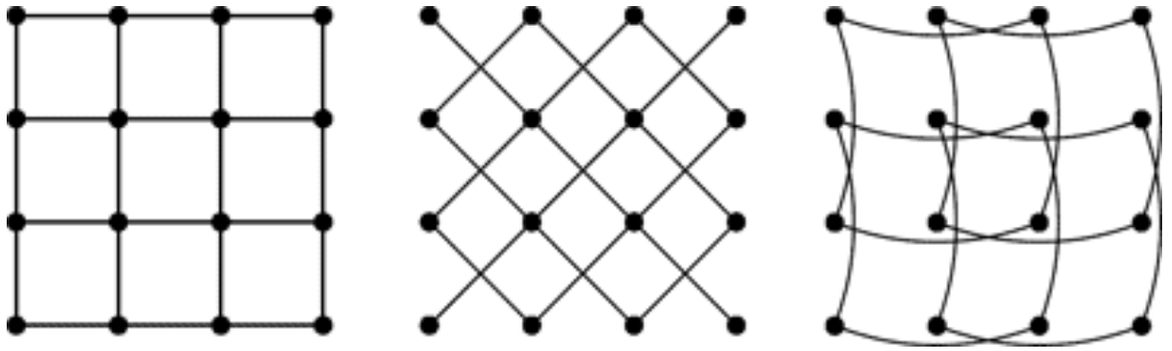
- Fire, smoke
- Cloth
  - VLC

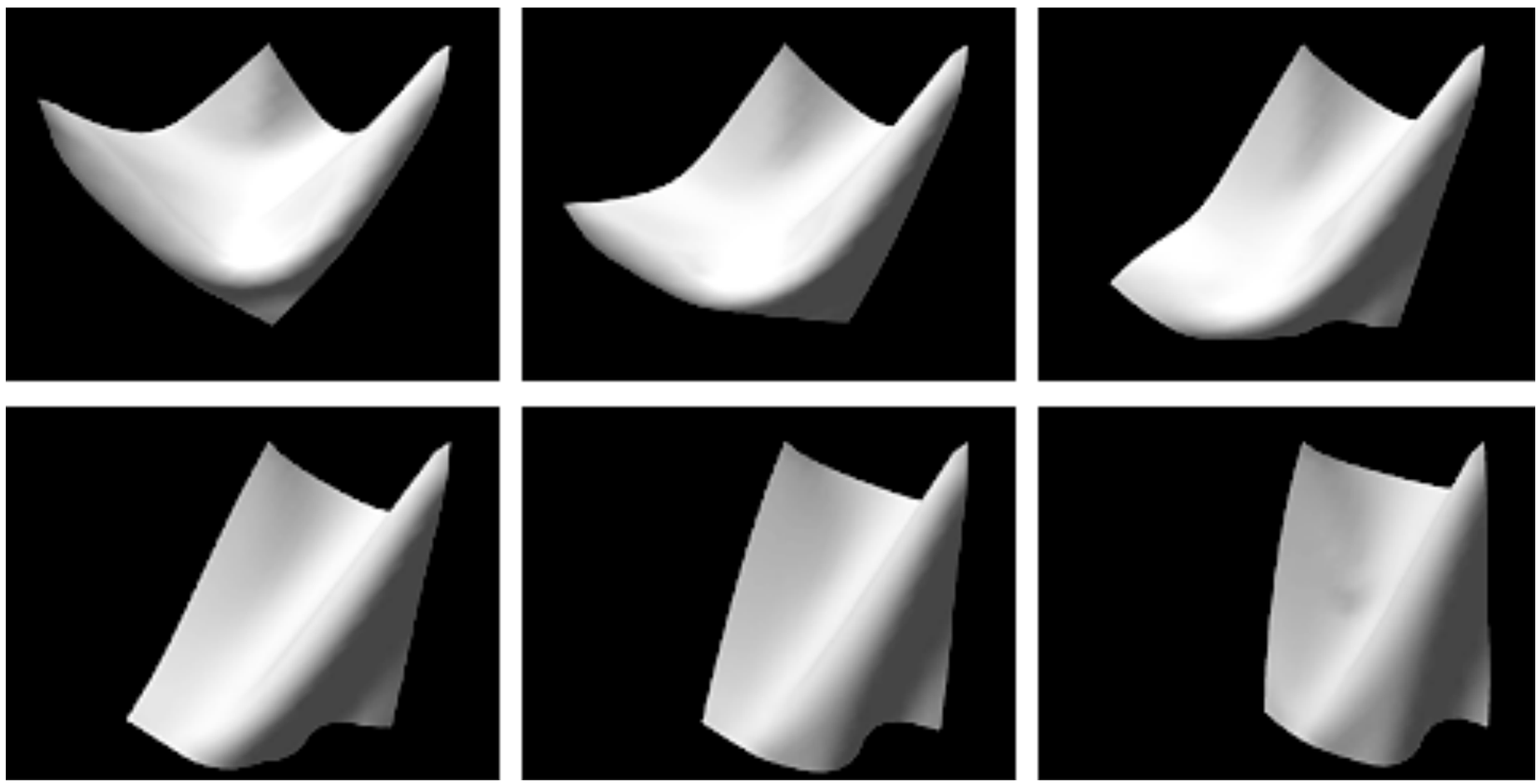
- Quotes because we do “visual simulation”



Assignment 3: Physics

- Simulate cloth as a mass-spring network
  - ODE numerics



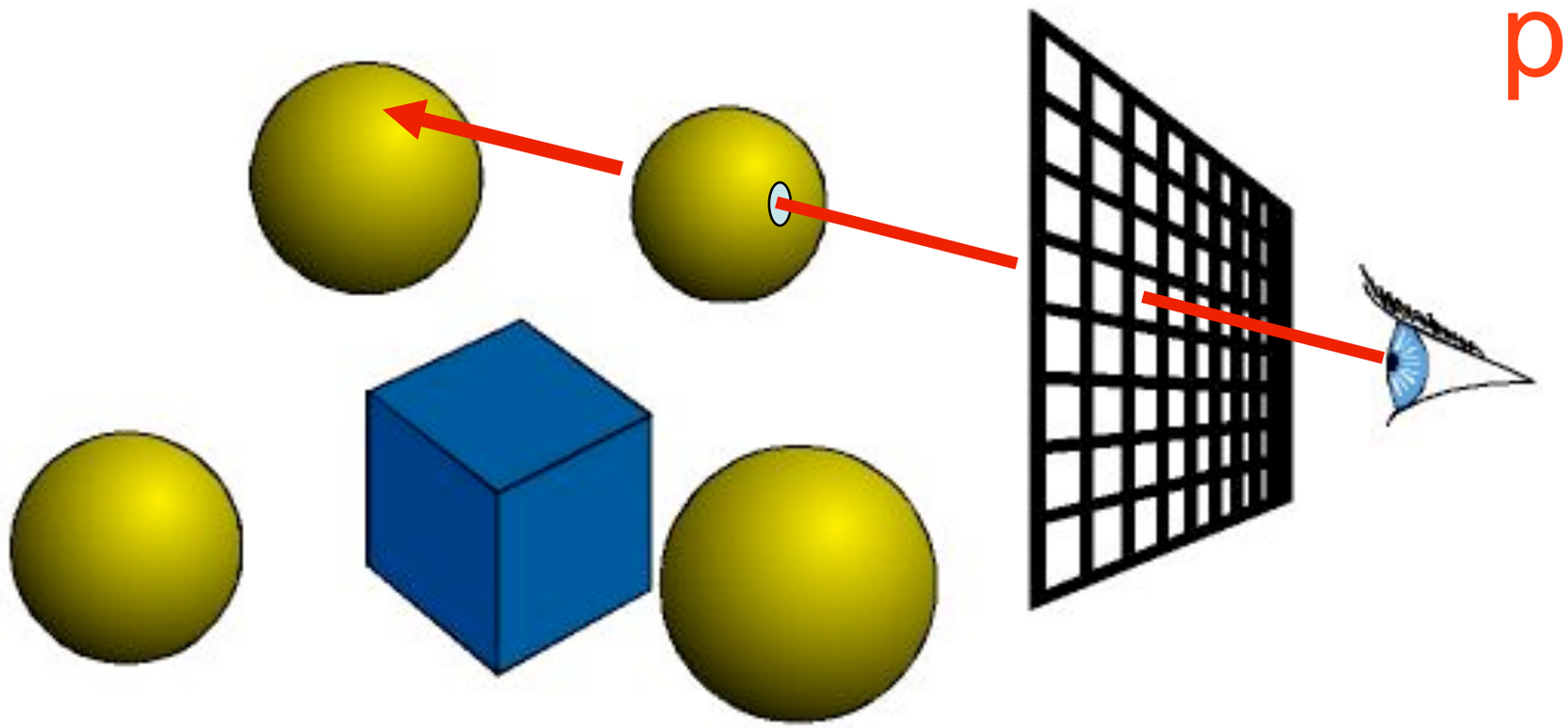


Eye Candy: Real-time fluid simulation

Rendering: Ray Casting

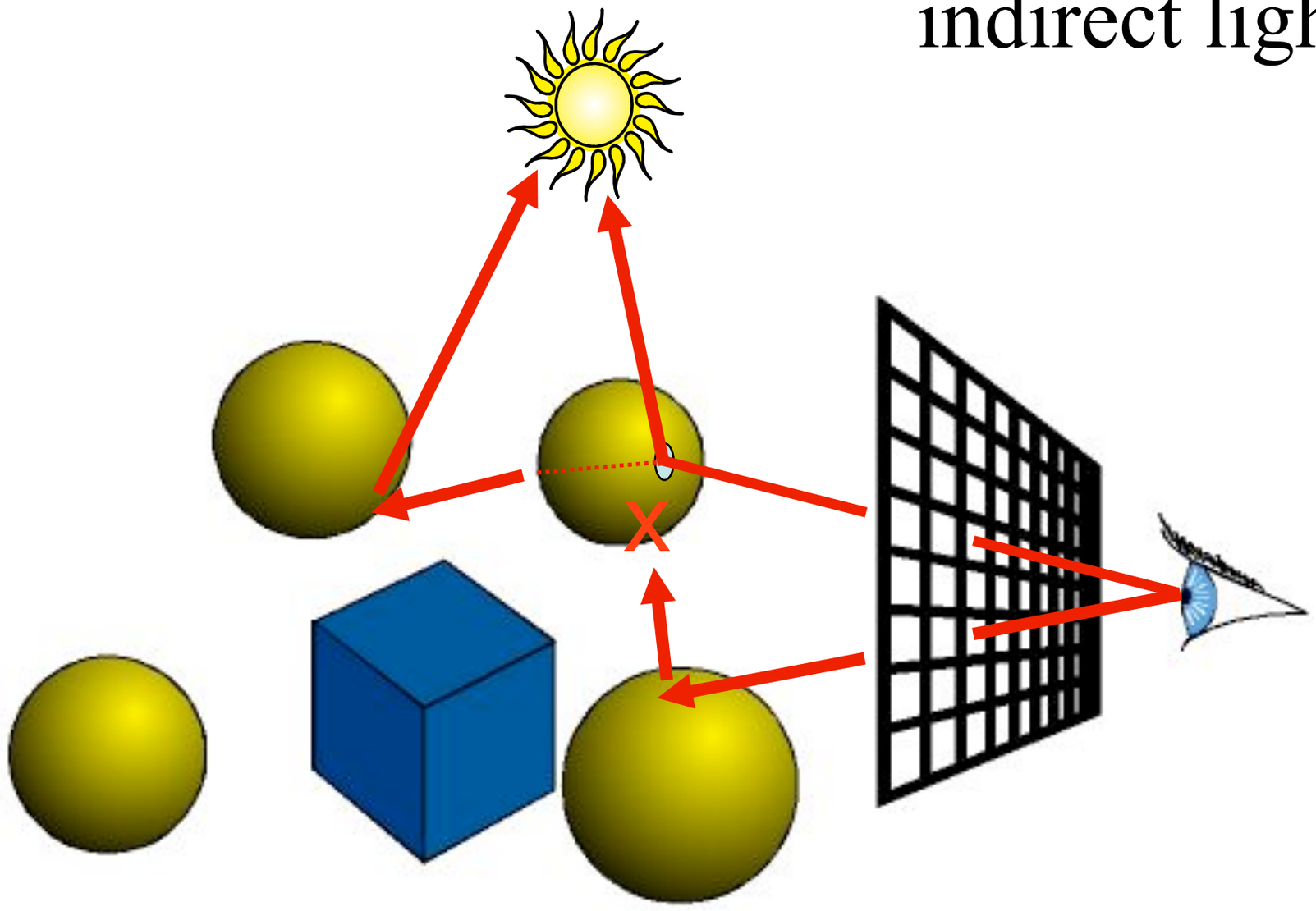
- For every pixel
  construct a ray from the eye
  - For every object in the scene
    - Find intersection with the ray
    - Keep if closest

Visibility or "hidden surface" problem

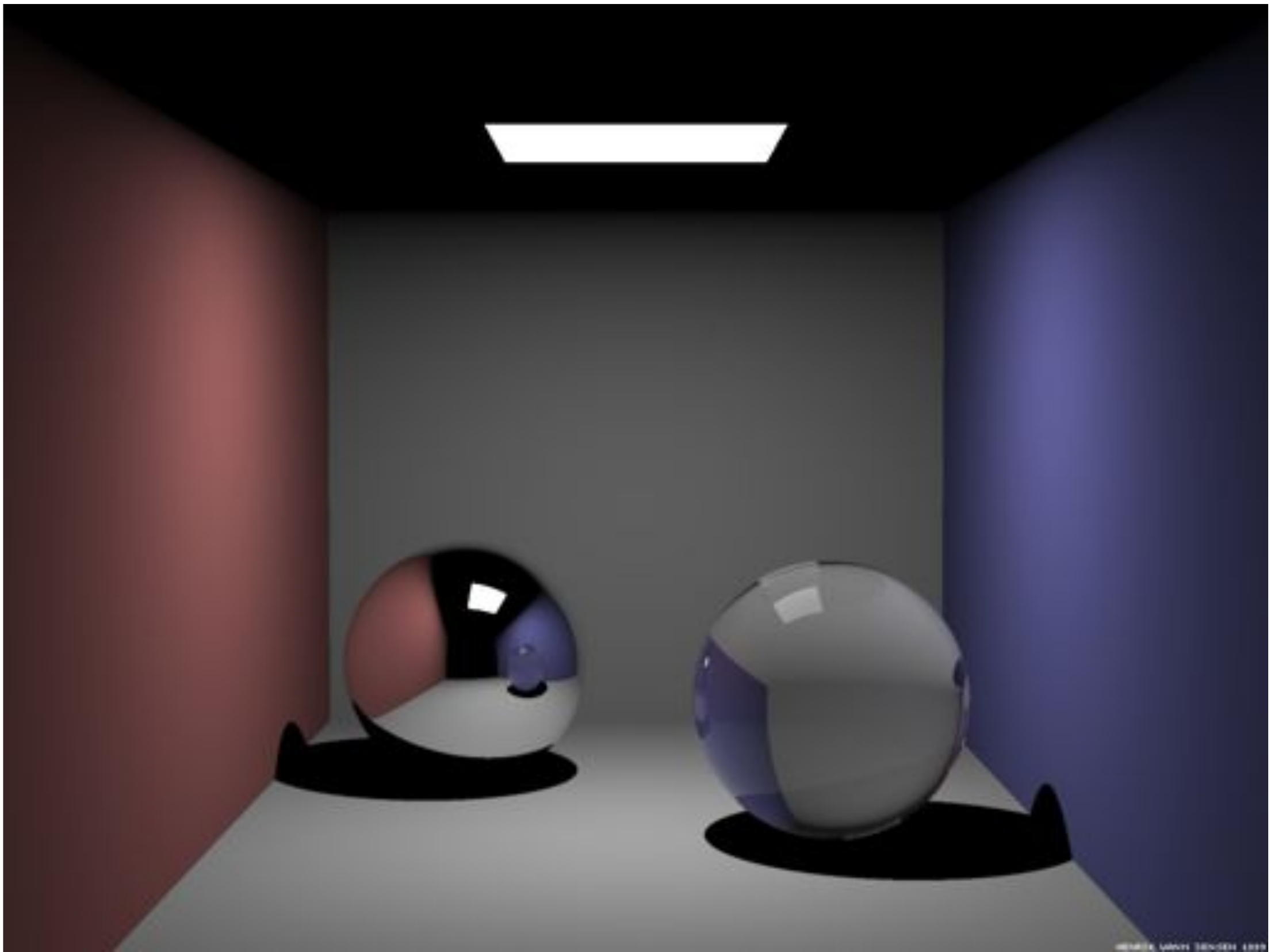


Rendering: Ray Tracing

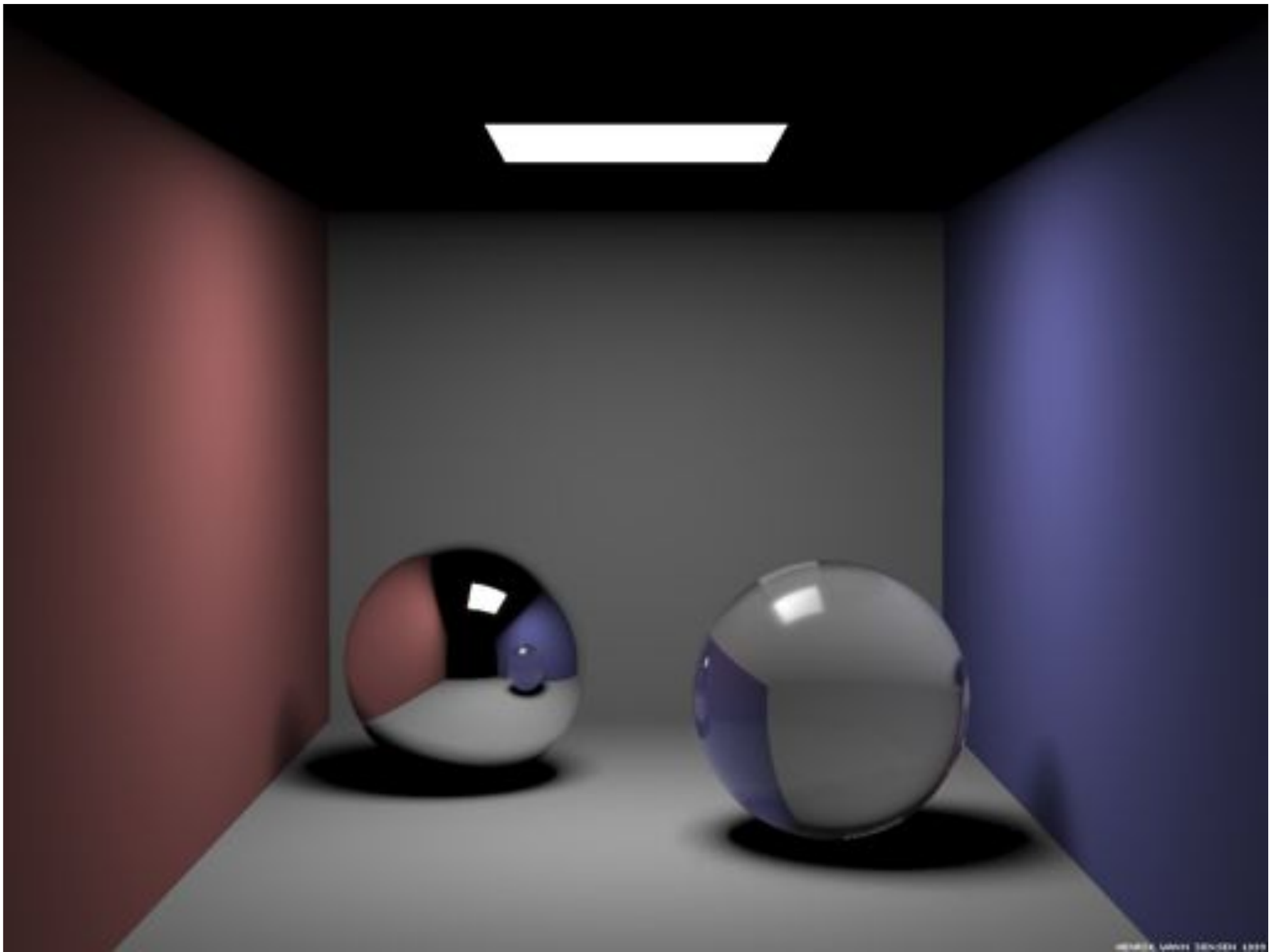
- **Shading**: Interaction of light and material
- Secondary rays (shadows, reflection, refraction, indirect lighting)



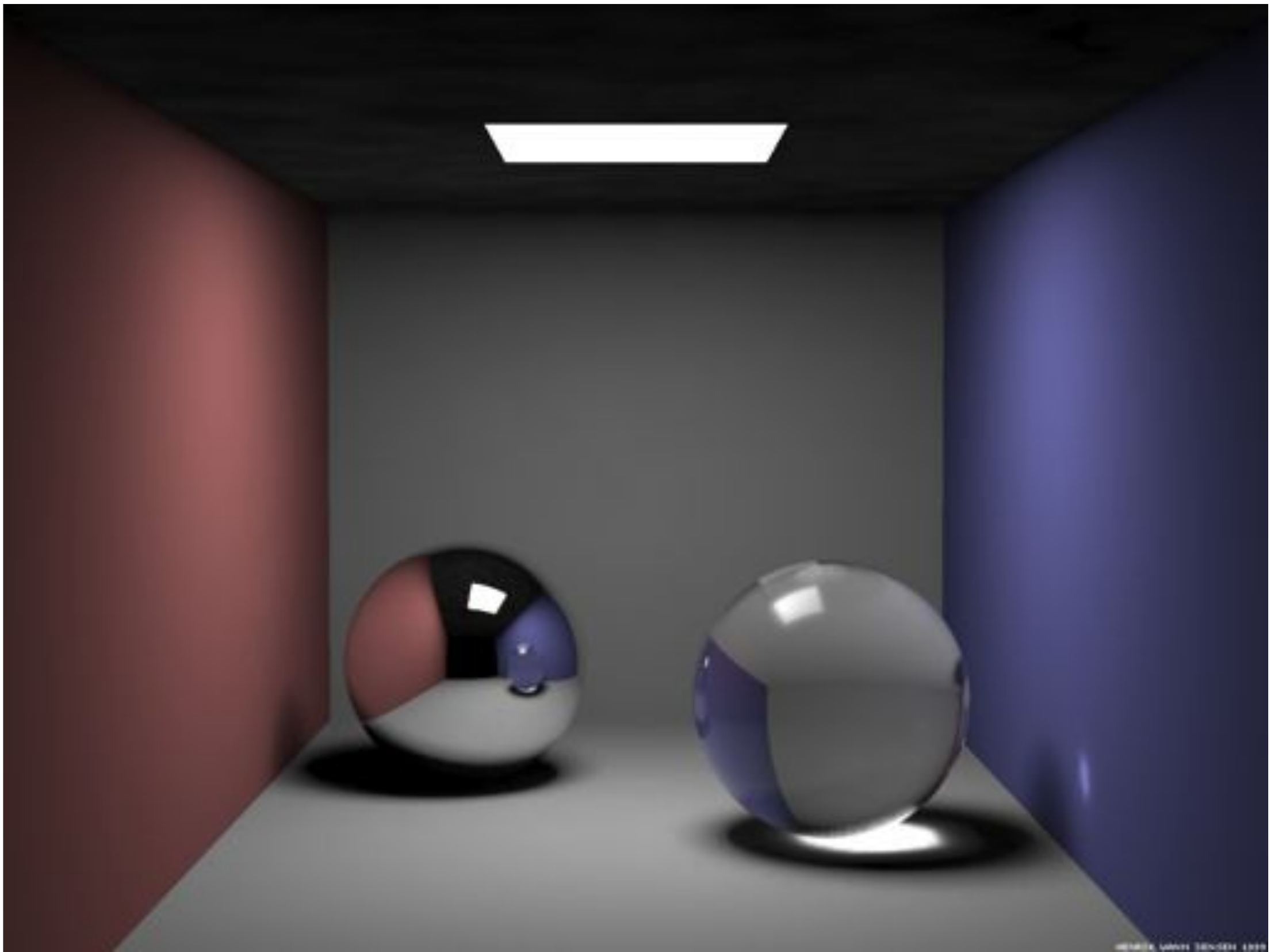
Traditional Ray Tracing



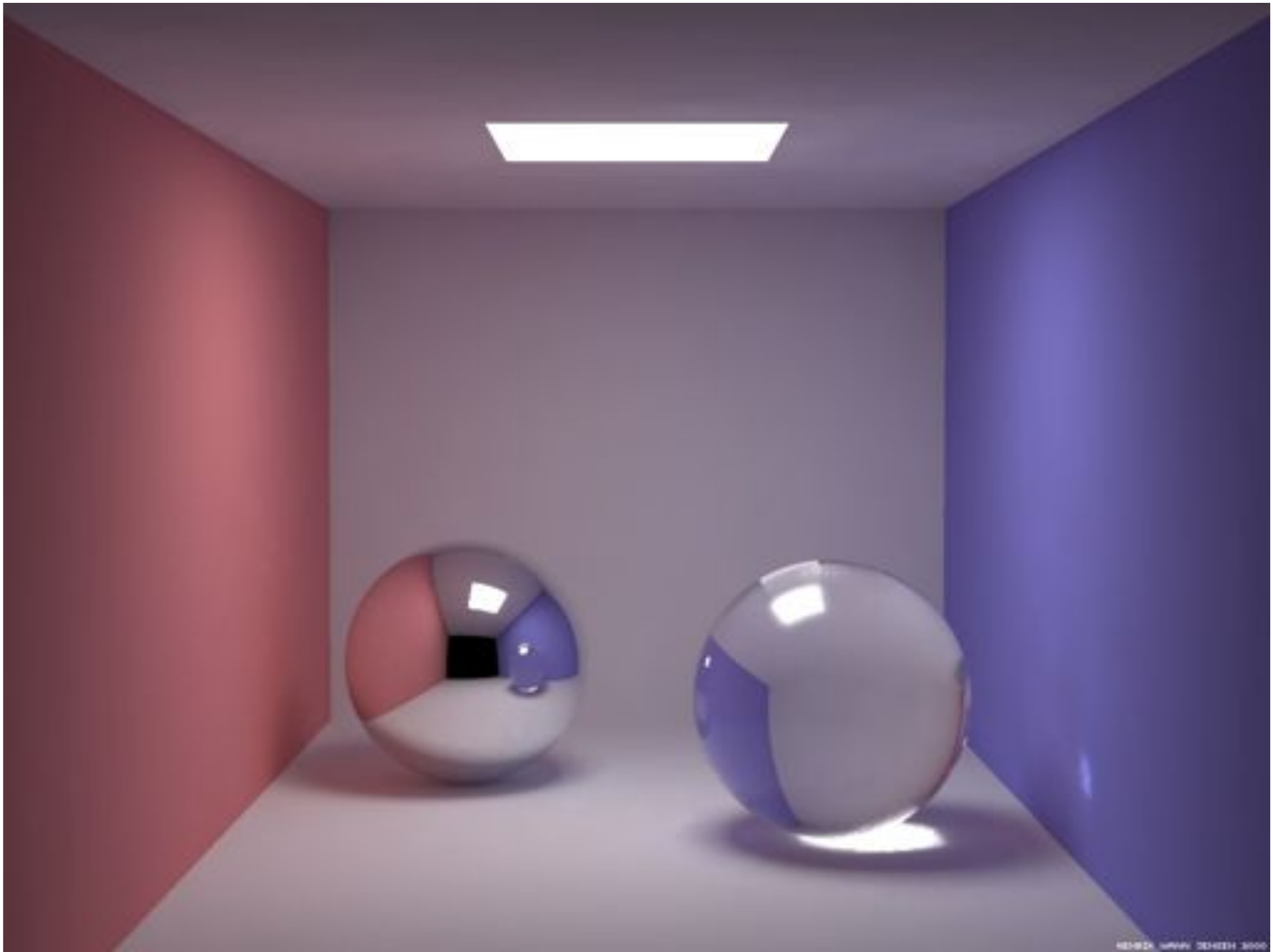
Ray Tracing + Soft Shadows



Ray Tracing + Caustics

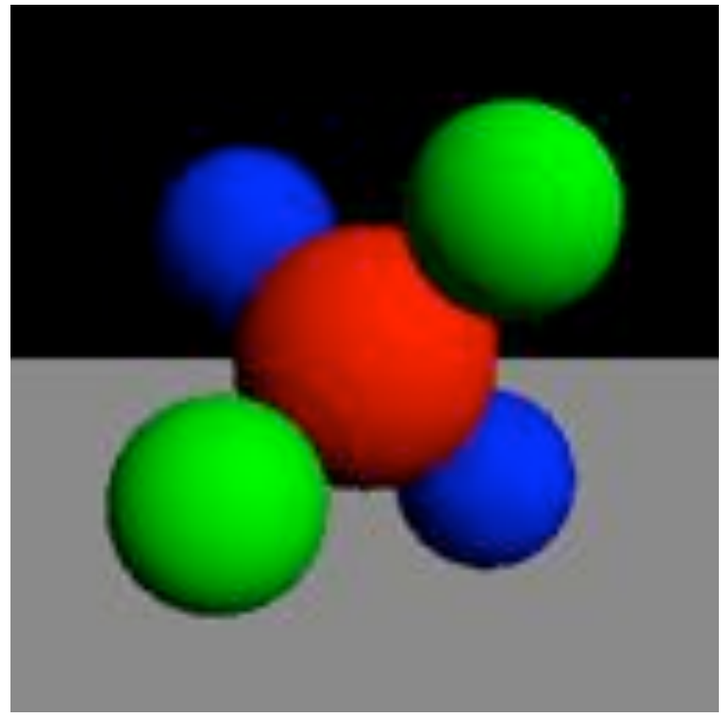


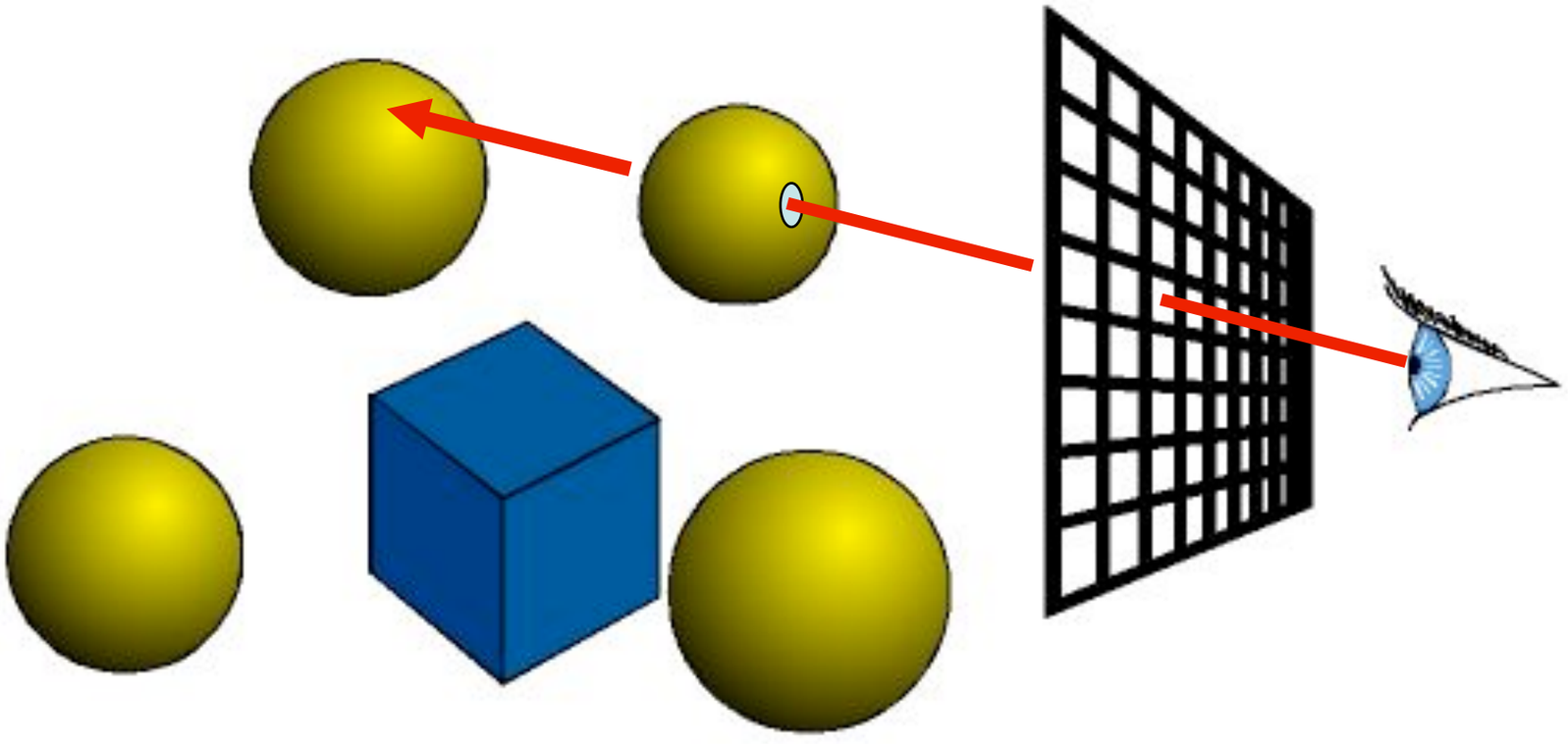
Global (Indirect) Illumination



Assignment 4: Ray Casting+Tracing

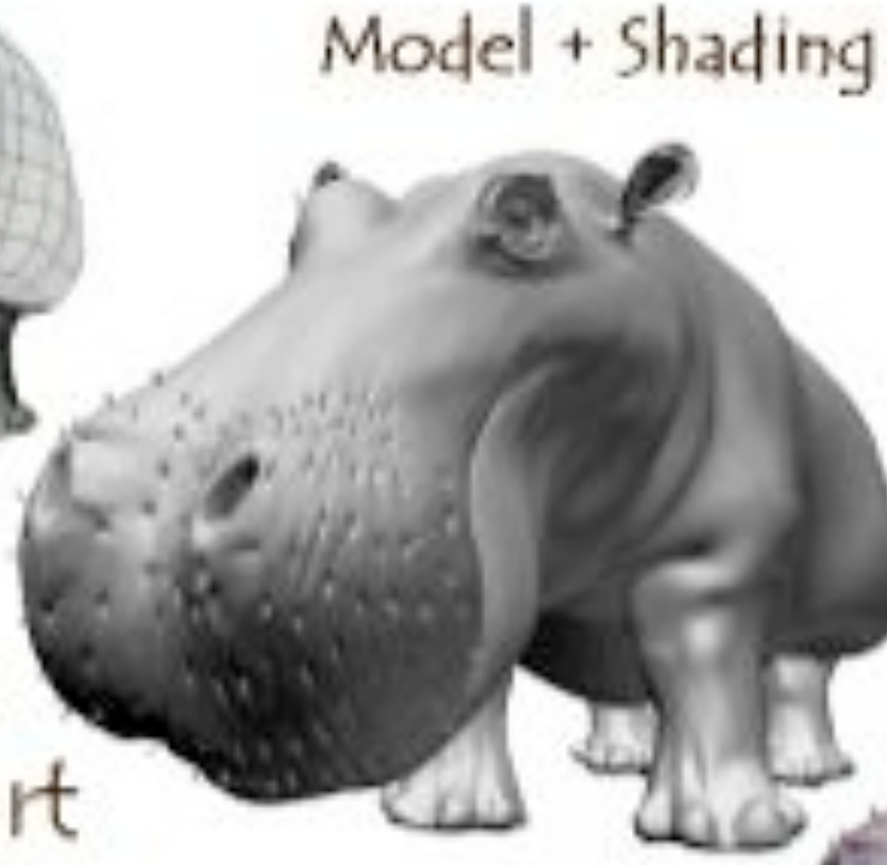
- Cast rays from the viewpoint
- Intersect with scene primitives
- Compute simple shading





Textures and Shading

Model



Model + Shading

Model + Shading + Textures

At what point do things start looking real?

For more info on the computer artwork of Jeremy Birn see http://www.3drender.com/jbirn/productions.html

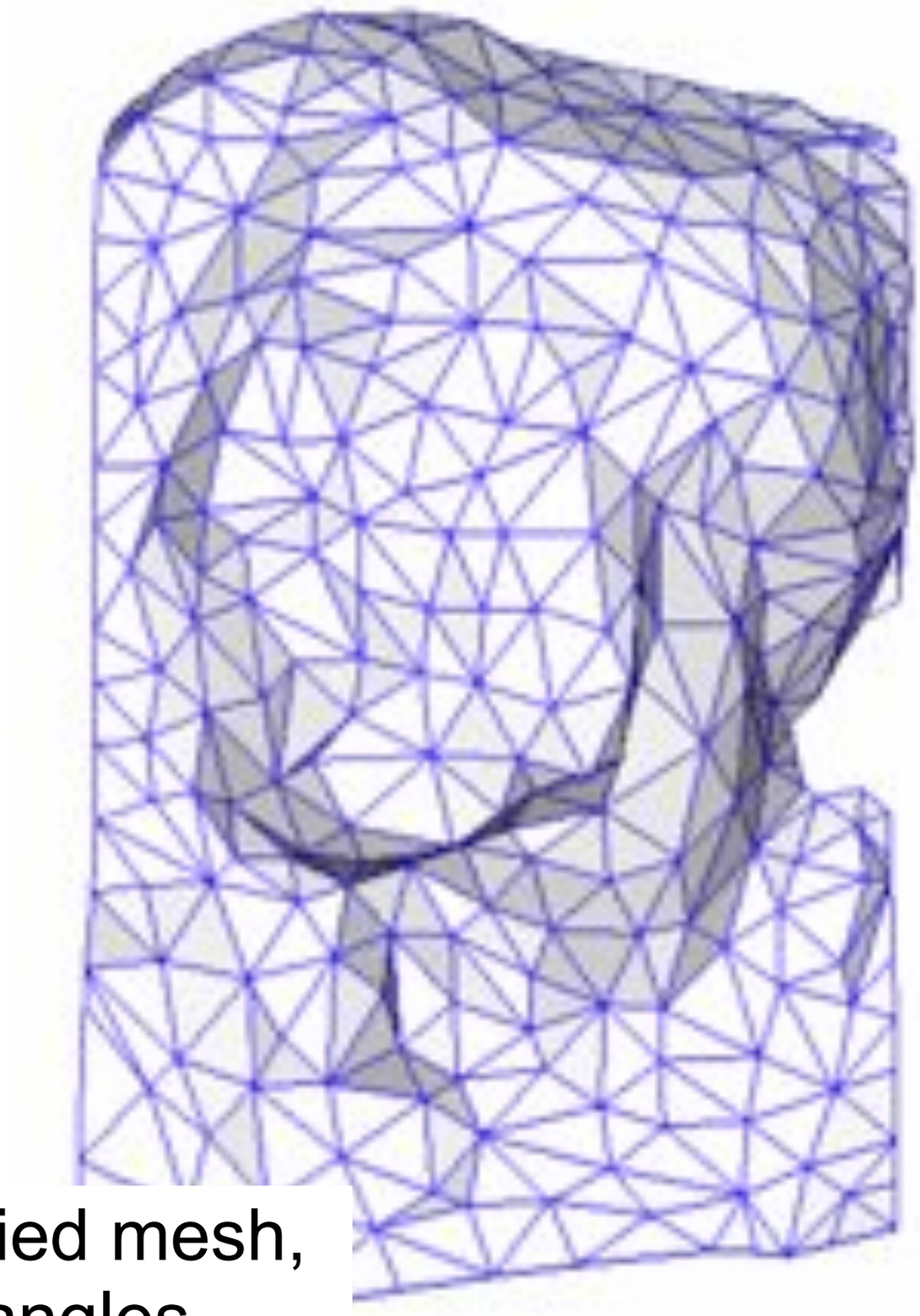
Normal Map Example

Paolo Cignoni

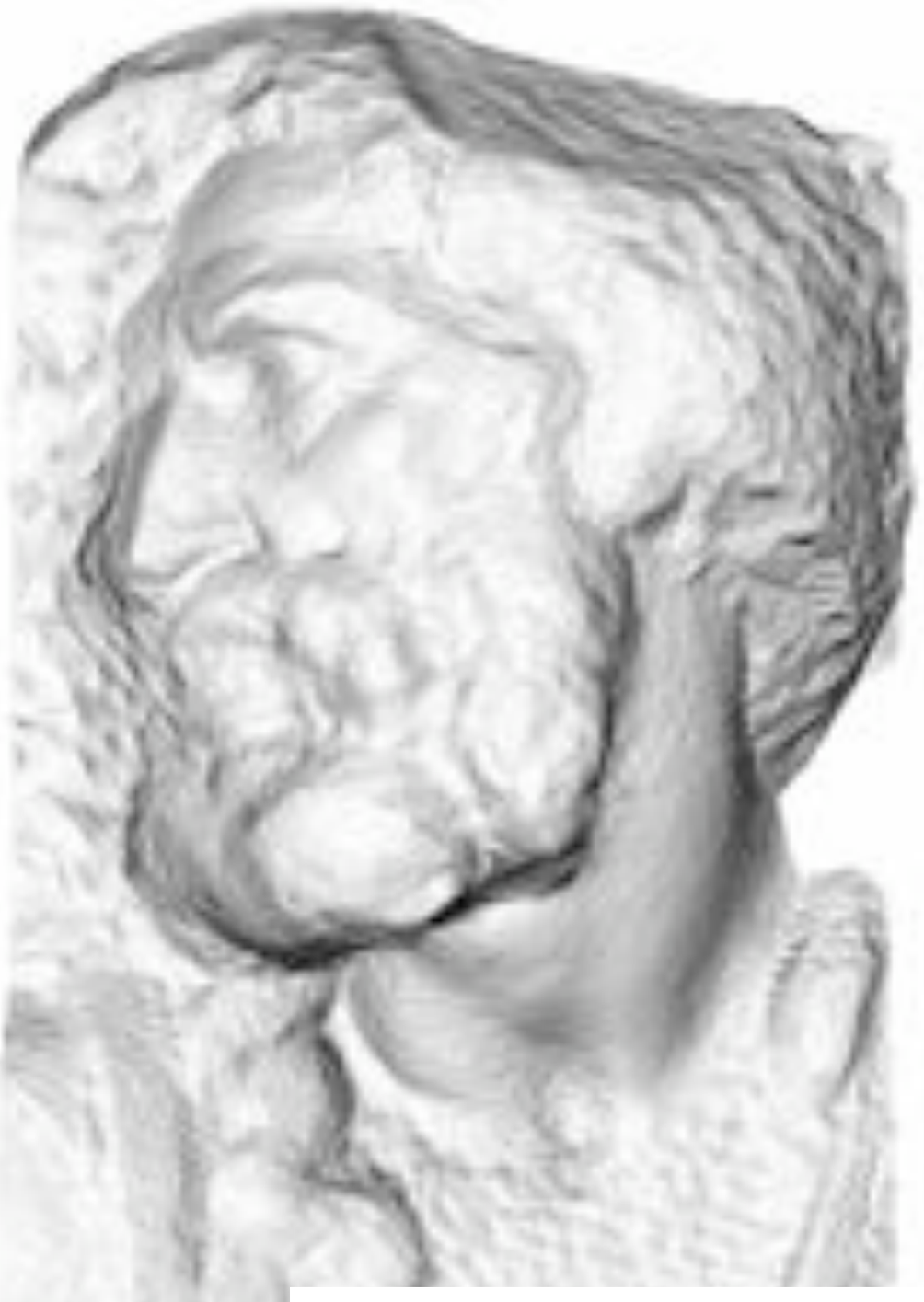
Original Mesh
4M triangles

Normal Map Example

Paolo Cignoni



Simplified mesh, 500 triangles

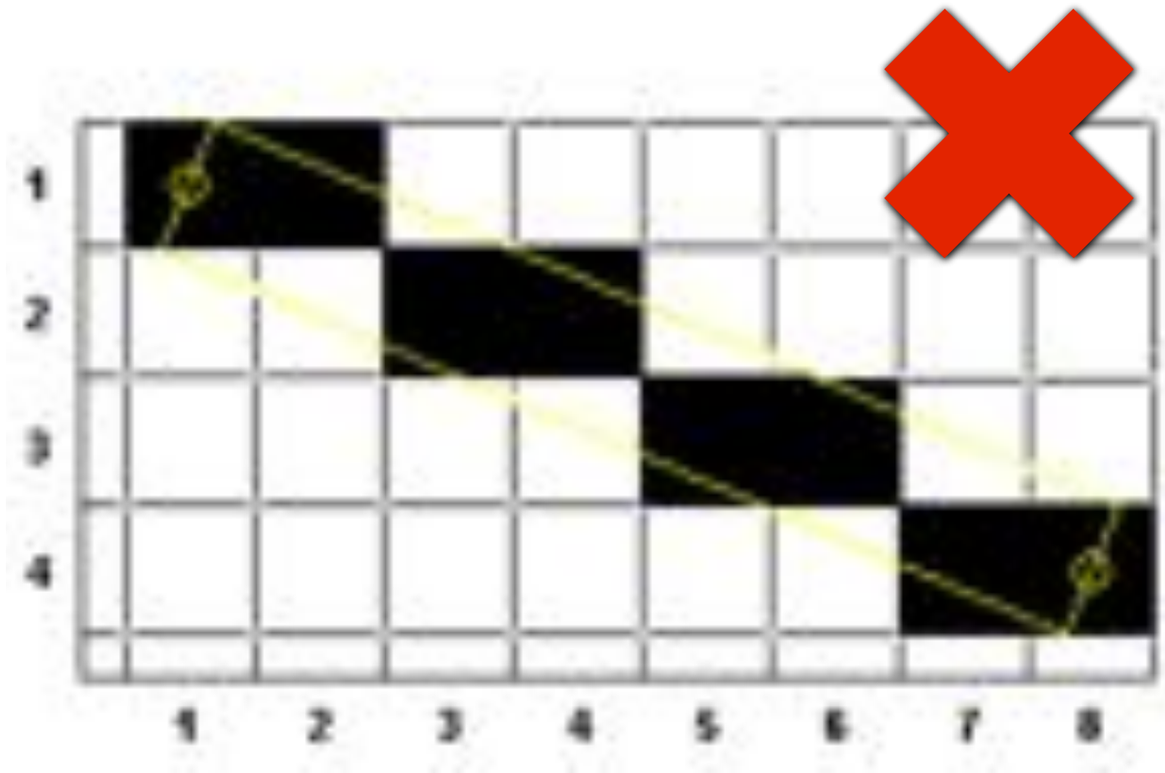


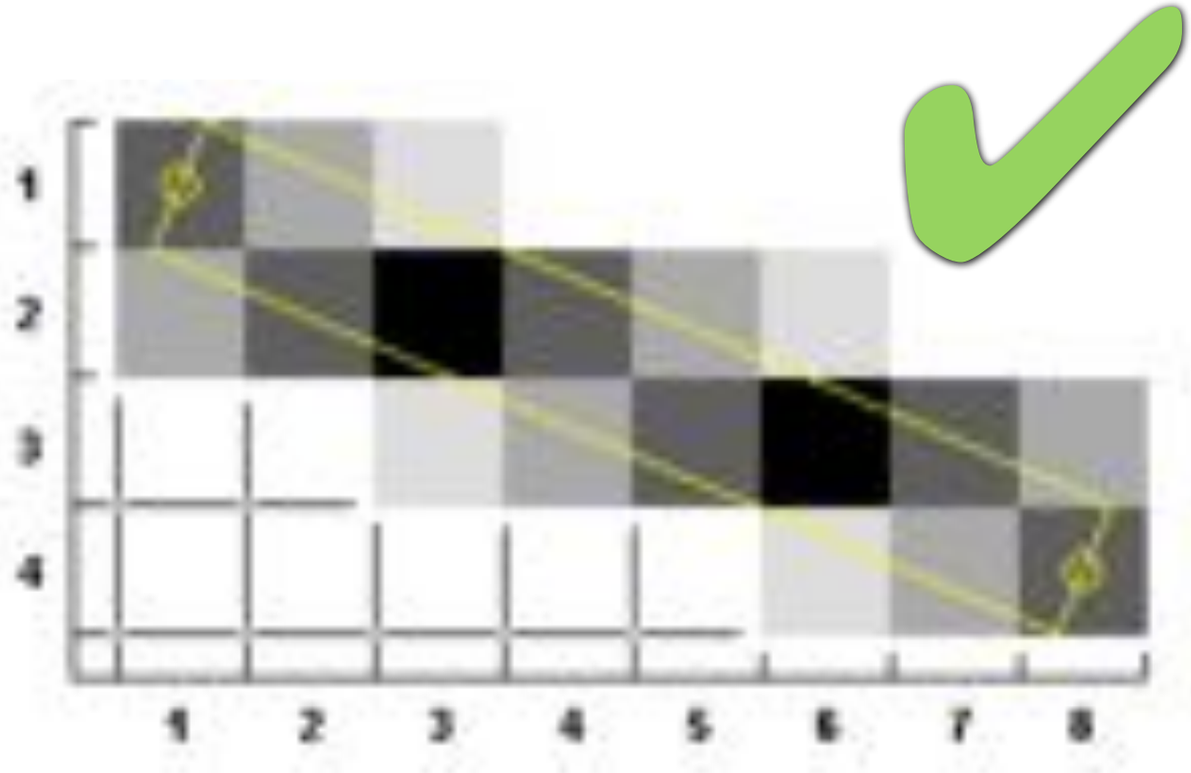
Simplified mesh + normal mapping

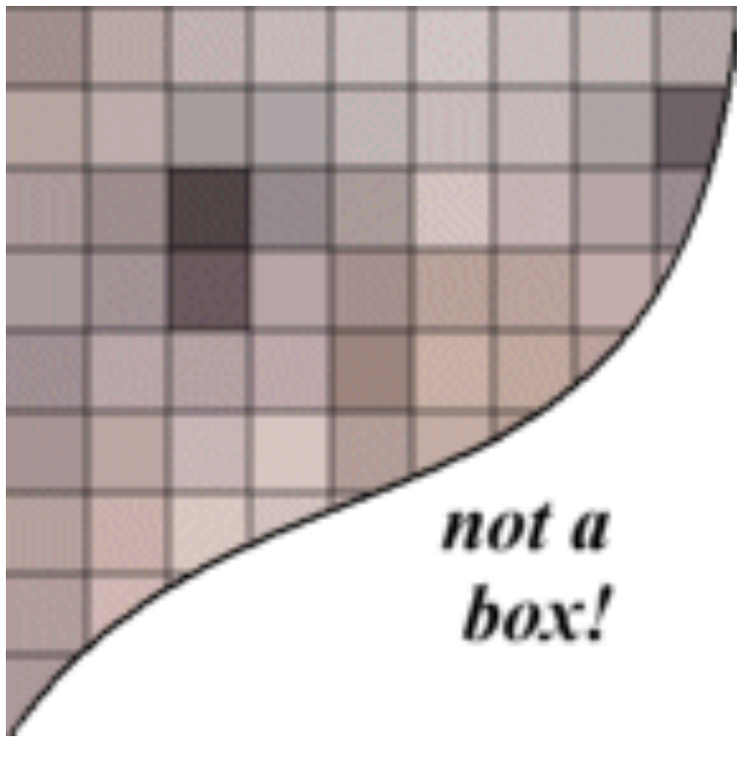
ehtinen

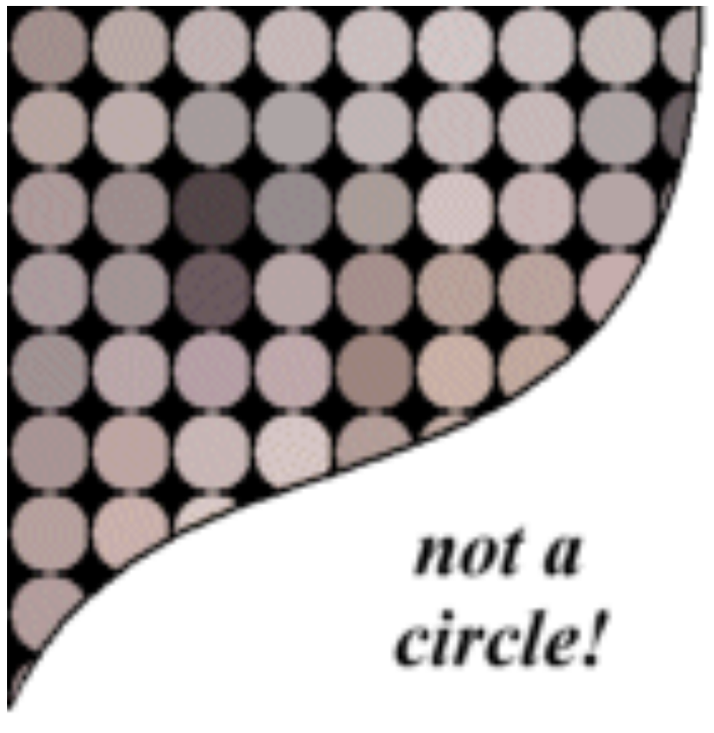
Yes, All This Works

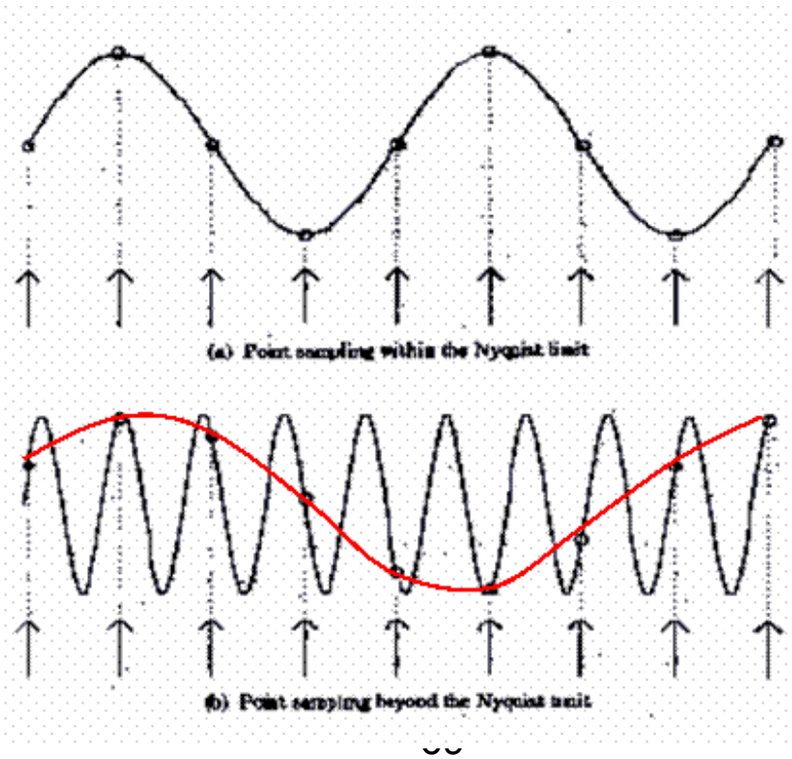
Sampling & Antialiasing











Shadows

Figure 12. Frame from *Luxo Jr.*

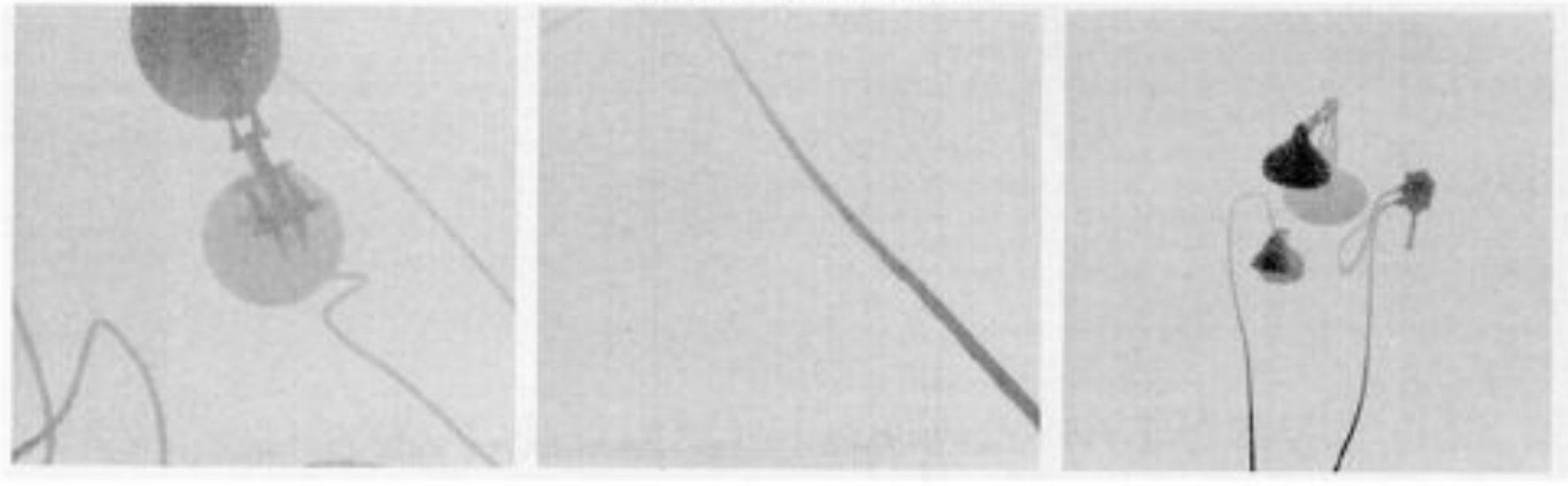
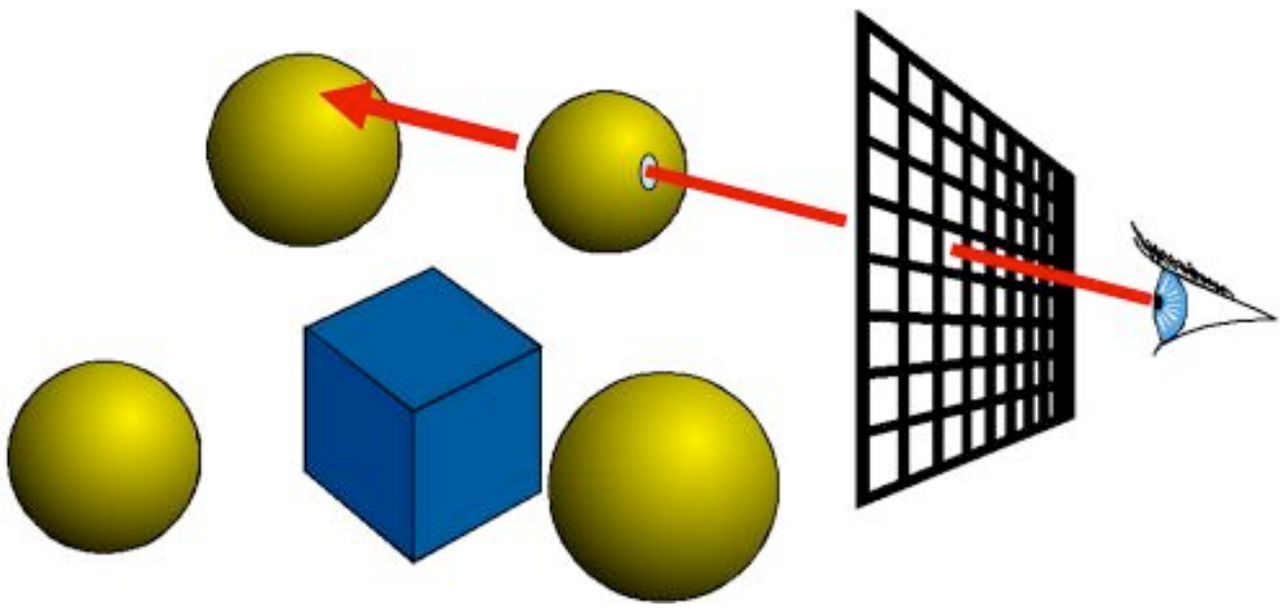


Figure 13. Shadow maps from *Luxo Jr.*

The Graphics Pipeline

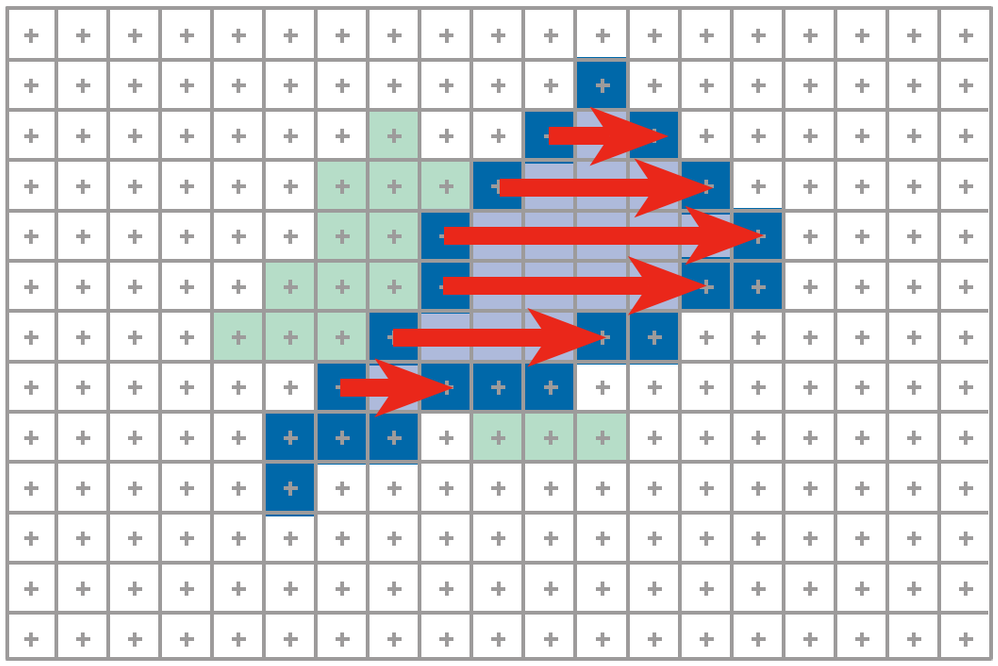
Ray Casting

```
For each pixel
    For each object
      Does object hit pixel?
```



GPU

```
For each triangle
    For each pixel
      Does pixel hit triangle?
```



The Graphics Pipeline

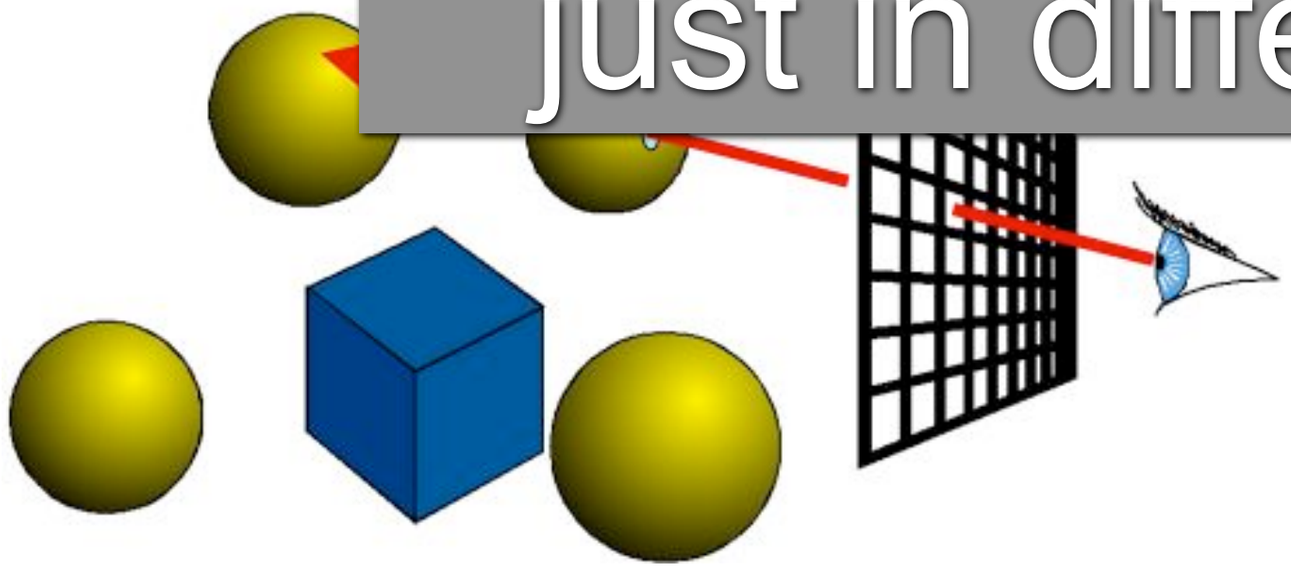
Ray Casting

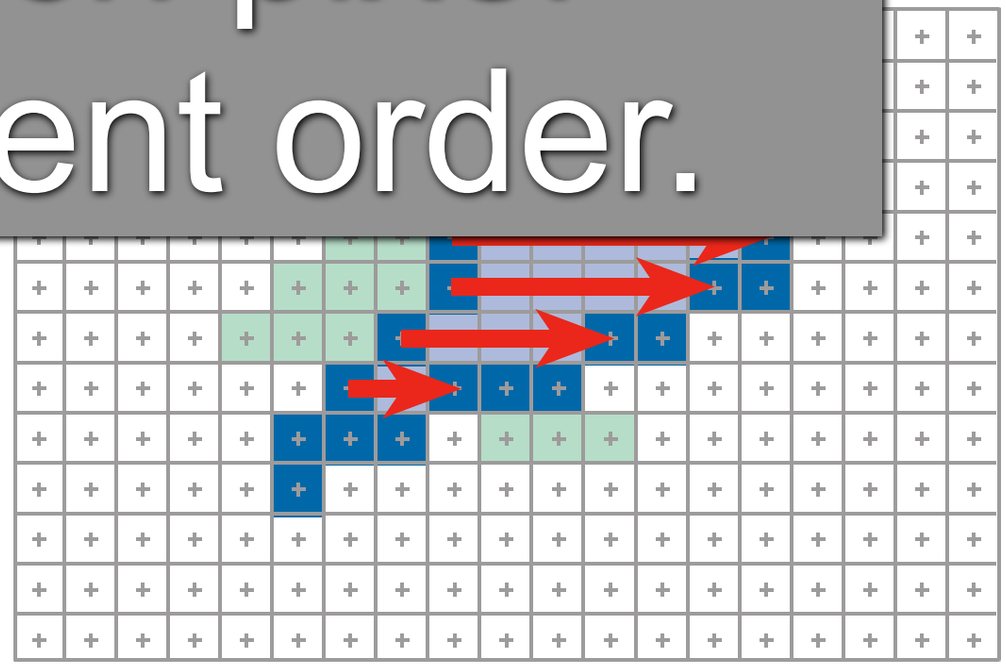
GPU

```
For each
    For e
        Ser
```

```
ls
```

Both are ways of determining what is visible in each pixel – just in different order.





Phew! That's a lot of stuff!

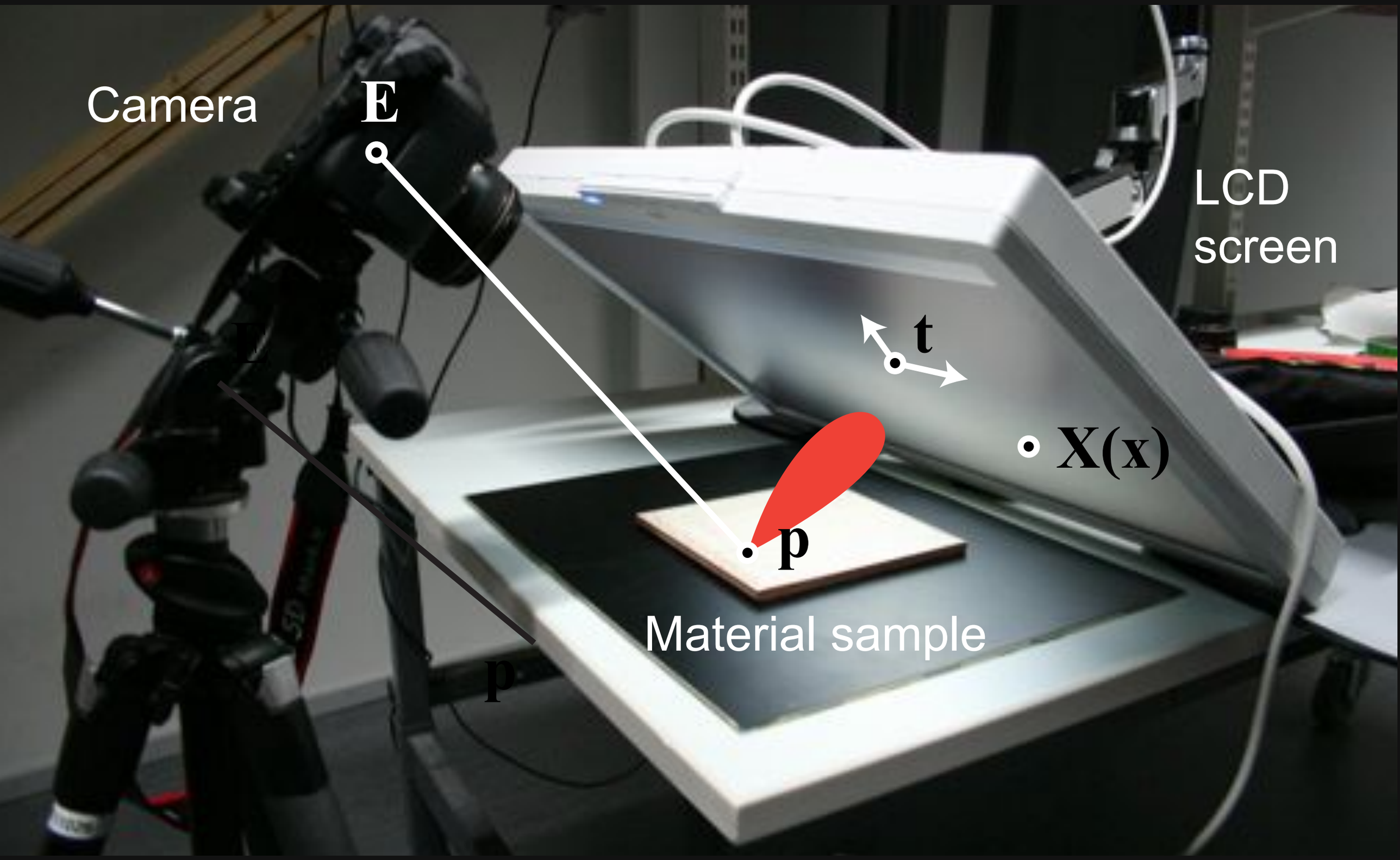
- BUT: Mastering all this takes you a long way towards cool applications!

Little Big Planet

More Research Goodness

- My PhD student Miika Aittala is advancing the state of the art in realistic material appearance capture and rendering
  - Highly competitive field

- Video


Camera
E
LCD
screen
t
X(x)
p
Material sample
p

Questions?

T-111.4310 Prerequisites

- Not strictly enforced

- Calculus, Linear Algebra
  - Solving equations, derivatives, integrals
  - Vectors, matrices, basis, solving systems of equations
  - *Optional review/introductory session*

- All assignments are in C++
  - *Optional review/introductory session*

Grading Policy

- Assignments: 70%
  - Two-week programming assignments
  - Must be completed individually
  - *No final project*

- Midterm Quiz+Final Exam: 25%

- Participation: 5%
  - Can make a difference to your grade!

Assignments: Scoring

- Fulfill all requirements: you get a 10
  - Partial success judged case by case

- All assignments include starter code and detailed instructions, maybe going into more detail than lectures

- Each assignment includes a number of extra credit tasks ranging from easy to possibly very hard
  - Open-ended scale

Extra Credit: Why Bother?

- 1st: it's fun!
- 2nd: prizes!

- Fall 2012: *Three students with highest total assignment scores got a private tour at Remedy Entertainment*, makers of Max Payne, Alan Wake, Death Rally
  - Face-to-face with the people who do the games and the tech
- Advanced class (5310), this spring: *Final rendering competition prize a very fast GPU donated by NVIDIA*
  - And...

Juror

Luca Fascione
Rendering Research Lead
Weta Digital

What Makes Graphics Fun?

- You can look at it from several angles
  - Anything that looks good will do in many applications...
    - Means you can really be creative once you know the basics.
  - ...but doing stuff "right" can be really involved.
    - Feels pretty nice when all that math and CS gives you a pretty picture or animation!
  - There is a continuum in between

That’s All Folks!

- Looking forward to seeing you in the fall!