Conclusions

CS 4620 Lecture 39

Announcements

- A7 due tonight

- Prelim/Final
  - Next Thu at 7pm
  - Contact us asap if you have constraints
  - Review session: Tuesday evening (time TBD)

Dynamic range

- Dynamic range $R_d = I_{max} / I_{min}$, or $(I_{max} + k) / (I_{min} + k)$
  - determines the degree of image contrast that can be achieved
  - a major factor in image quality
- Ballpark values
  - Desktop display in typical conditions: 20:1
  - Photographic print: 30:1
  - Desktop display in good conditions: 100:1
  - High-end display under ideal conditions: 1000:1
  - Digital cinema projection: 1000:1
  - Photographic transparency (directly viewed): 1000:1
  - High dynamic range display: 10,000:1

How many levels are needed?

- Depends on dynamic range
  - 2% steps are most efficient:

$$0 \mapsto I_{\min}; 1 \mapsto 1.02 I_{\min}; 2 \mapsto (1.02)^2 I_{\min}; \ldots$$

  - log 1.02 is about 1/120, so 120 steps per decade of dynamic range
    - 240 for desktop display
    - 480 to drive HDR display
- If we want to use linear quantization (equal steps)
  - one step must be < 2% (1/50) of $I_{\min}$
  - need to get from ~0 to $I_{\min} \cdot R_d$, so need about 50 R_d levels
    - 1500 for a print; 5000 for desktop display; 500,000 for HDR display
- Moral: 8 bits is just barely enough for low-end applications
  - but only if we are careful about quantization

Intensity quantization in practice

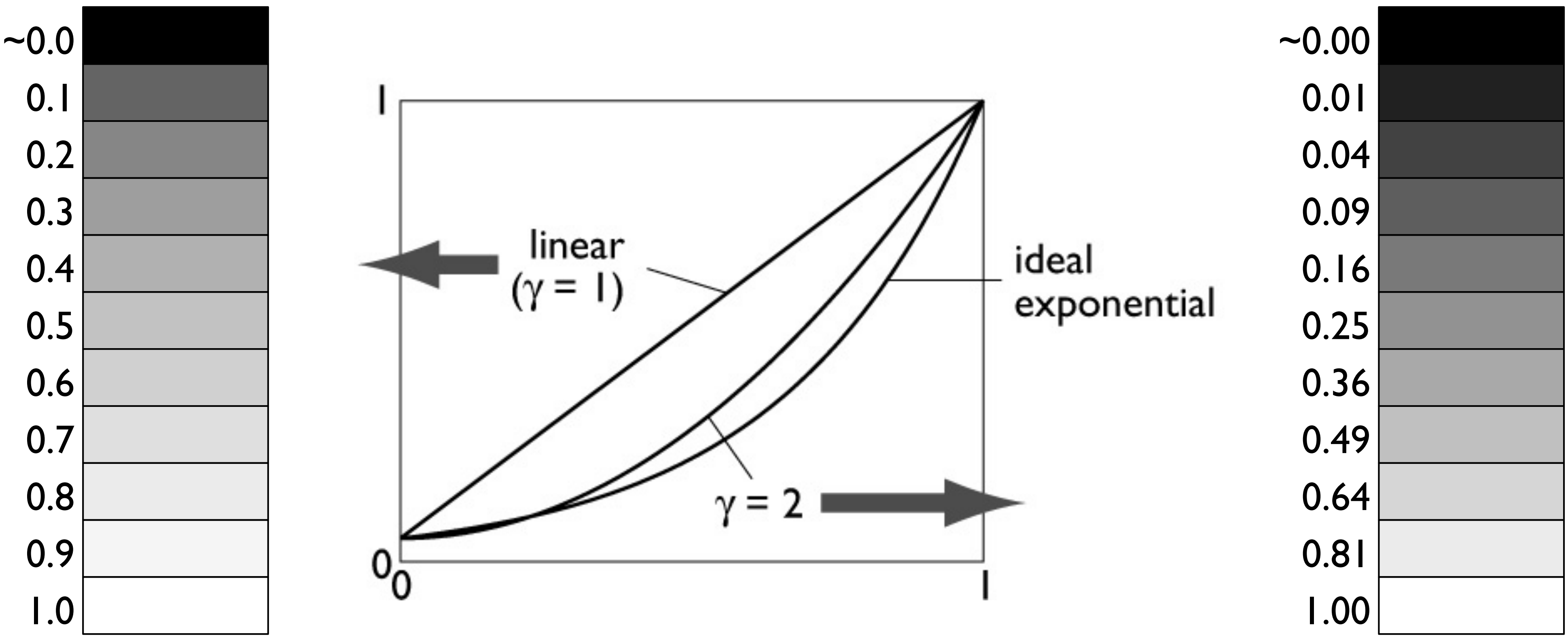
- Option 1: linear quantization $I(n) = (n/N)\, I_{\max}$
  - pro: simple, convenient, amenable to arithmetic
  - con: requires more steps (wastes memory)
  - need 12 bits for any useful purpose; more than 16 for HDR

- Option 2: power-law quantization $I(n) = (n/N)^{\gamma}\, I_{\max}$
  - pro: fairly simple, approximates ideal exponential quantization
  - con: need to linearize before doing pixel arithmetic
  - con: need to agree on exponent
  - 8 bits are OK for many applications; 12 for more critical ones

Why gamma?

- Power-law quantization, or *gamma correction* is most popular
- Original reason: CRTs are like that
  – intensity on screen is proportional to (roughly) voltage^2
- Continuing reason: inertia + memory savings
  – inertia: gamma correction is close enough to logarithmic that there's no sense in changing
  – memory: gamma correction makes 8 bits per pixel an acceptable option

Gamma quantization



- Close enough to ideal perceptually uniform exponential

Gamma correction

- Sometimes (often, in graphics) we have computed intensities *a* that we want to display linearly
- In the case of an ideal monitor with zero black level,

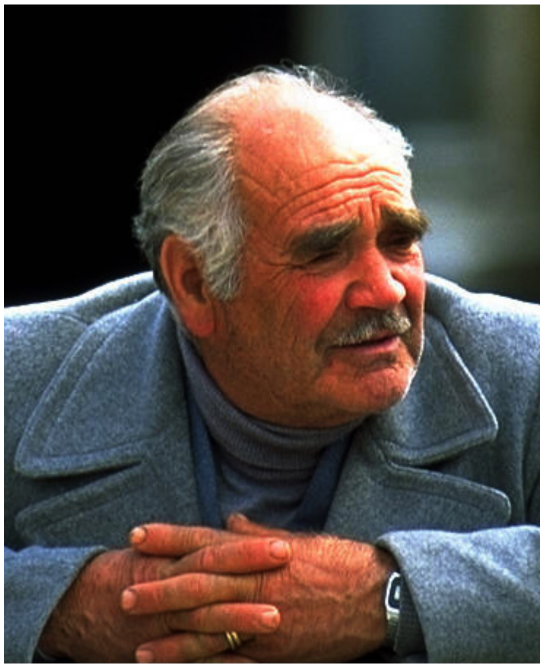
$$I(n) = (n/N)^{\gamma}$$

(where N = $2^n - 1$ in *n* bits). Solving for *n*:

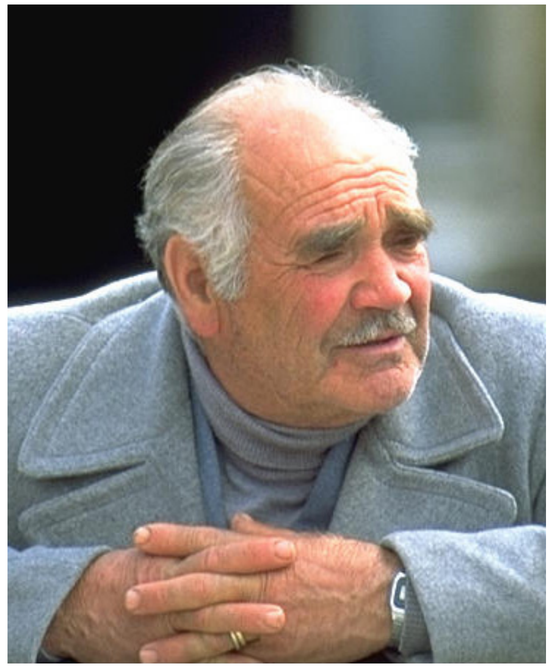
$$n(I) = N I^{\frac{1}{\gamma}}$$

- This is the "gamma correction" recipe that has to be applied when computed values are converted to 8 bits for output
  - failing to do this (implicitly assuming gamma = 1) results in dark, oversaturated images

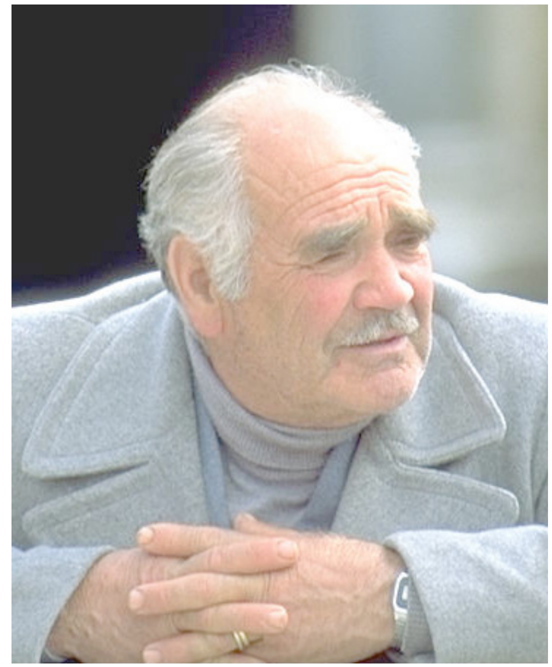
Gamma correction



corrected for
γ lower than
display

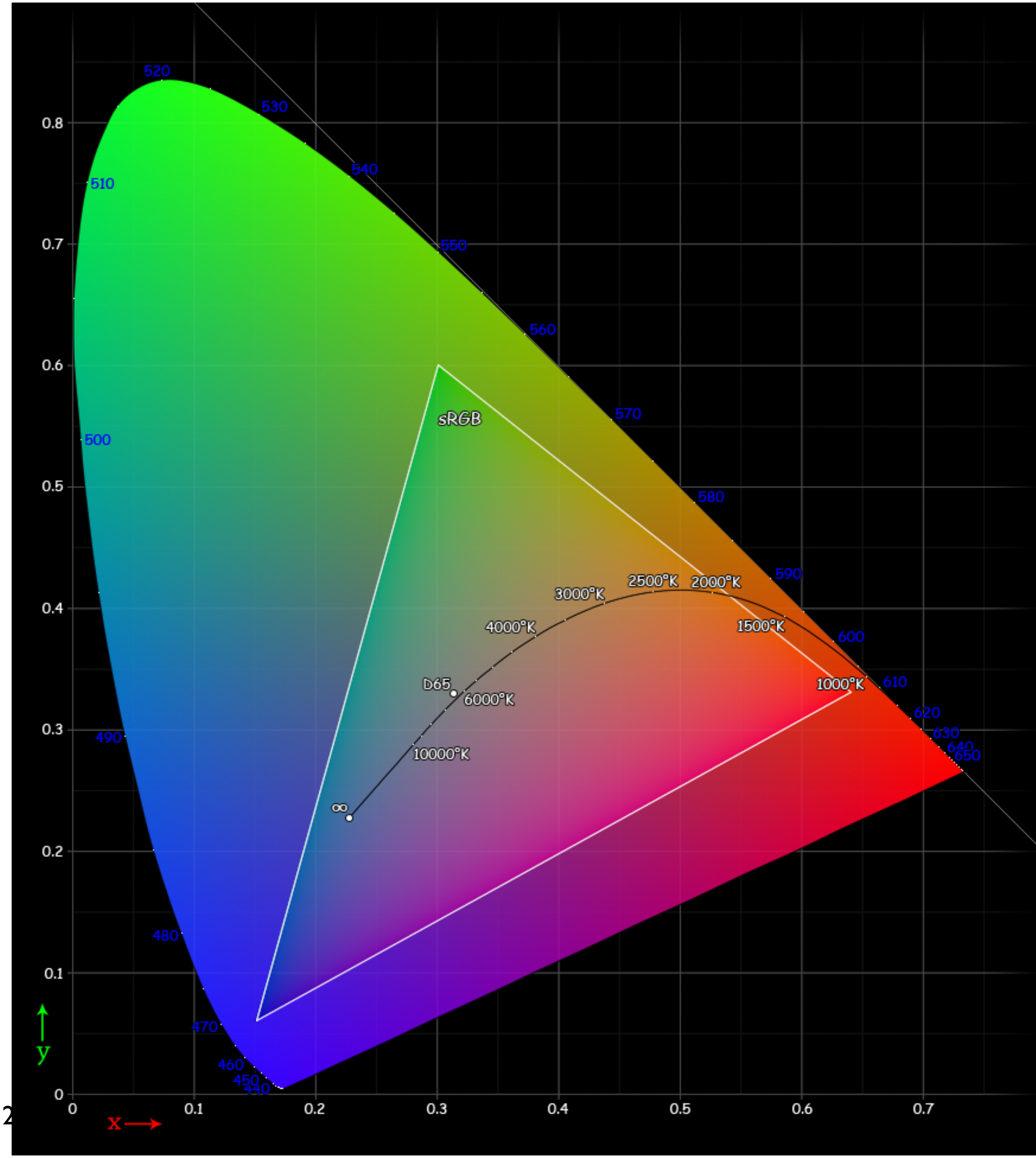


OK



corrected for
γ higher than
display

[Philip Greenspun]



520
530
540
510
550
560
570
500
580
590
600
610
620
630
640
650
490
480
470
460
450
440
sRGB
3000°K
2500°K
2000°K
4000°K
1500°K
1000°K
D65
6000°K
10000°K
∞
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
y
x

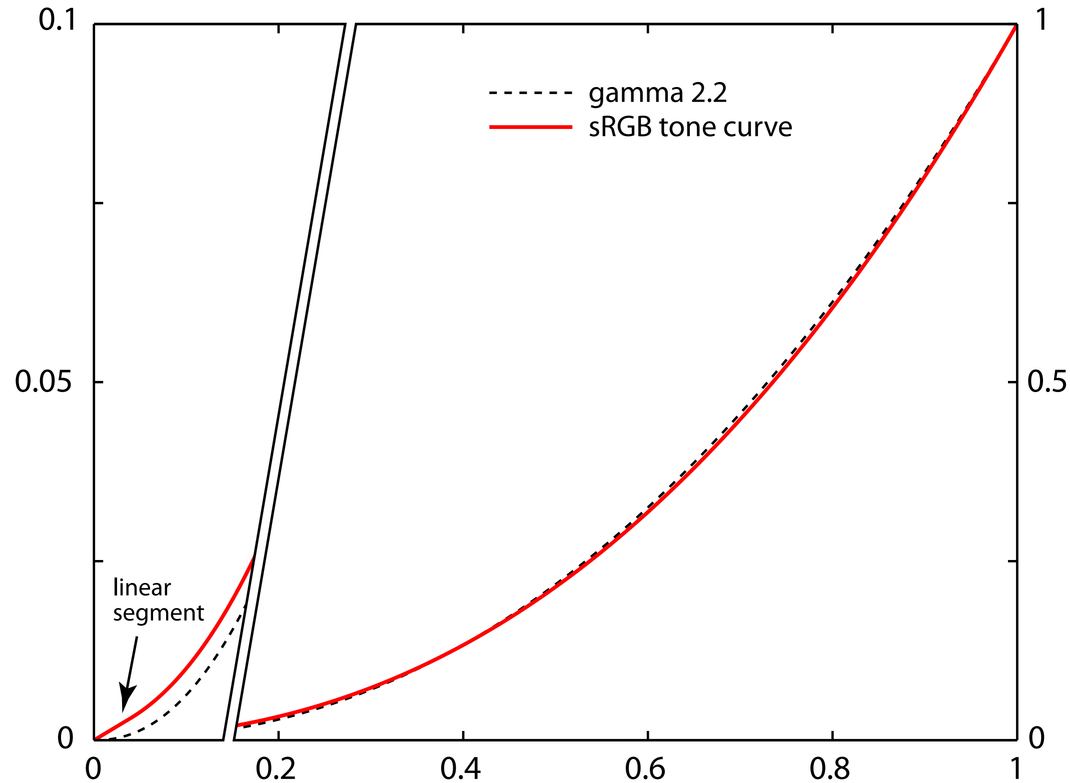
sRGB quantization curve

- The predominant standard for "casual color" in computer displays
  - consistent with older typical practice
  - designed to work well under imperfect conditions
  - these days all monitors are calibrated to sRGB by default
  - in practice, usually defines what your pixel values mean

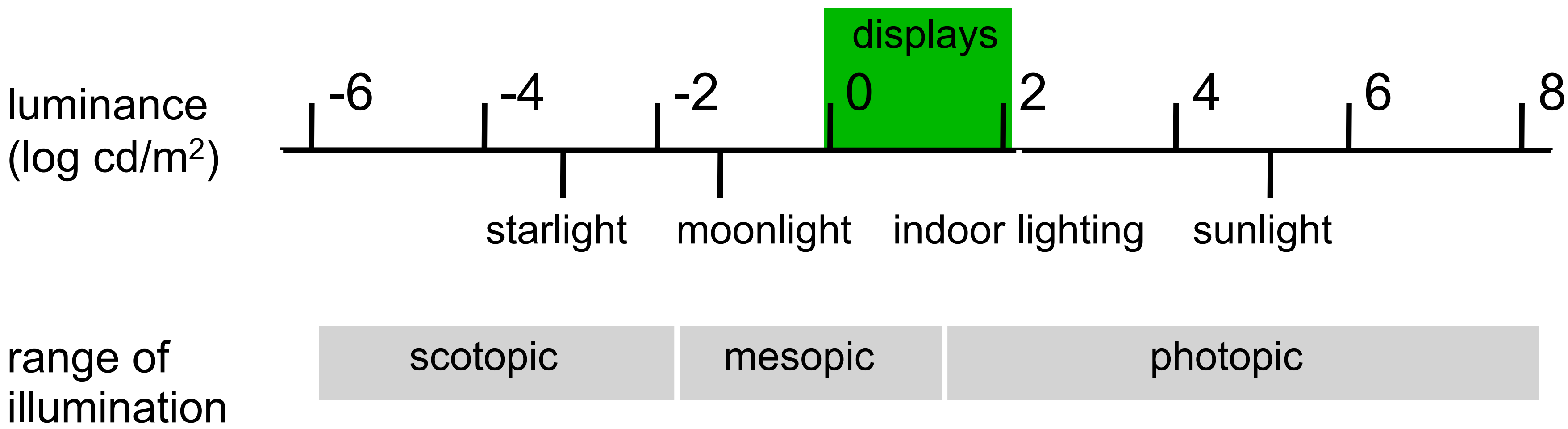
$$I(C) = \begin{cases} \frac{C}{12.92}, & C \leq 0.04045 \\ \left(\frac{C+a}{1+a}\right)^{2.4}, & C > 0.04045 \end{cases}$$

$$C = n/N$$

$$a = 0.055$$



Real World has high dynamic range



- poor visibility
- no color vision
- low acuity

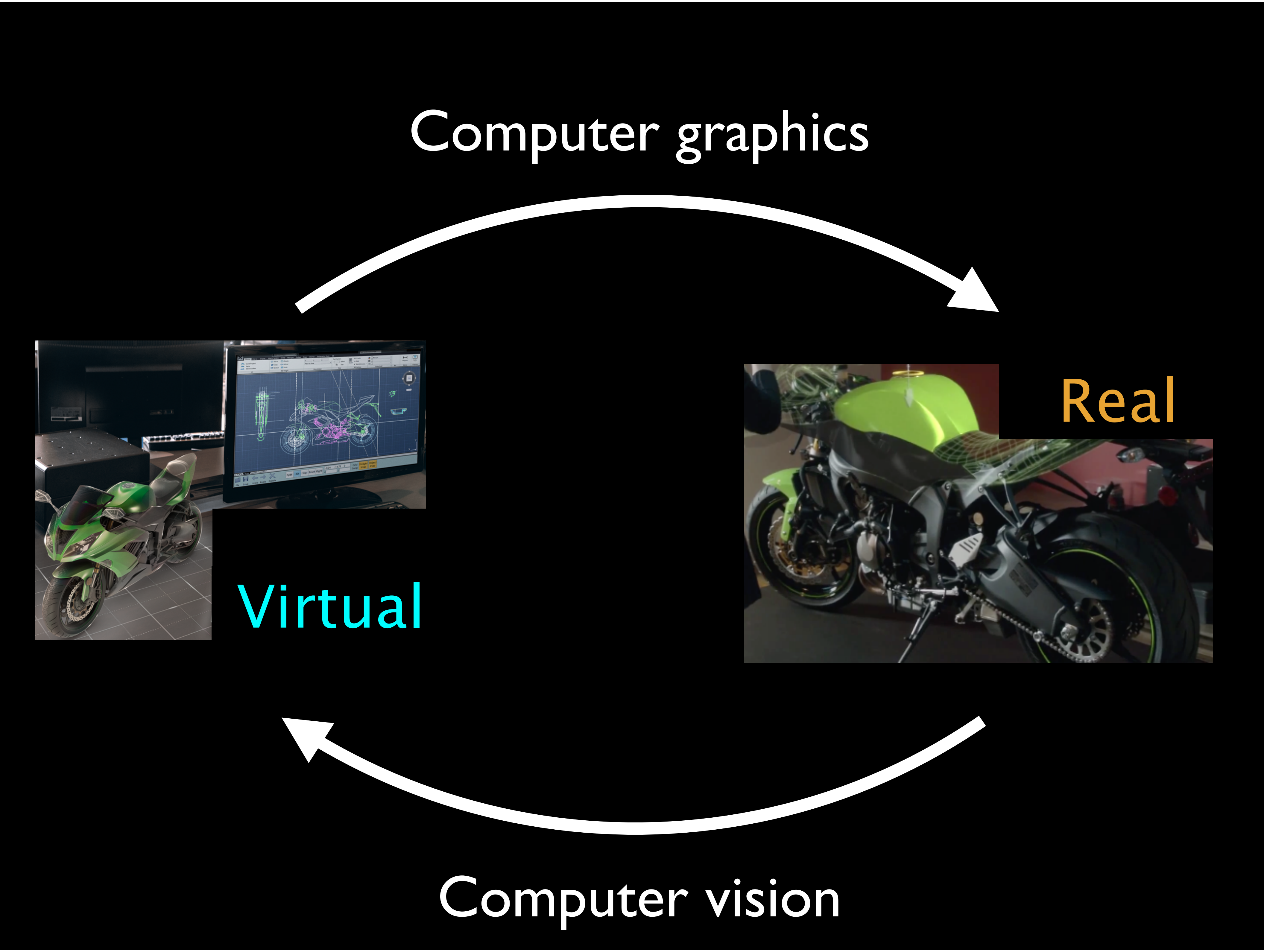
- good visibility
- good color vision
- high acuity

Converting from HDR to LDR

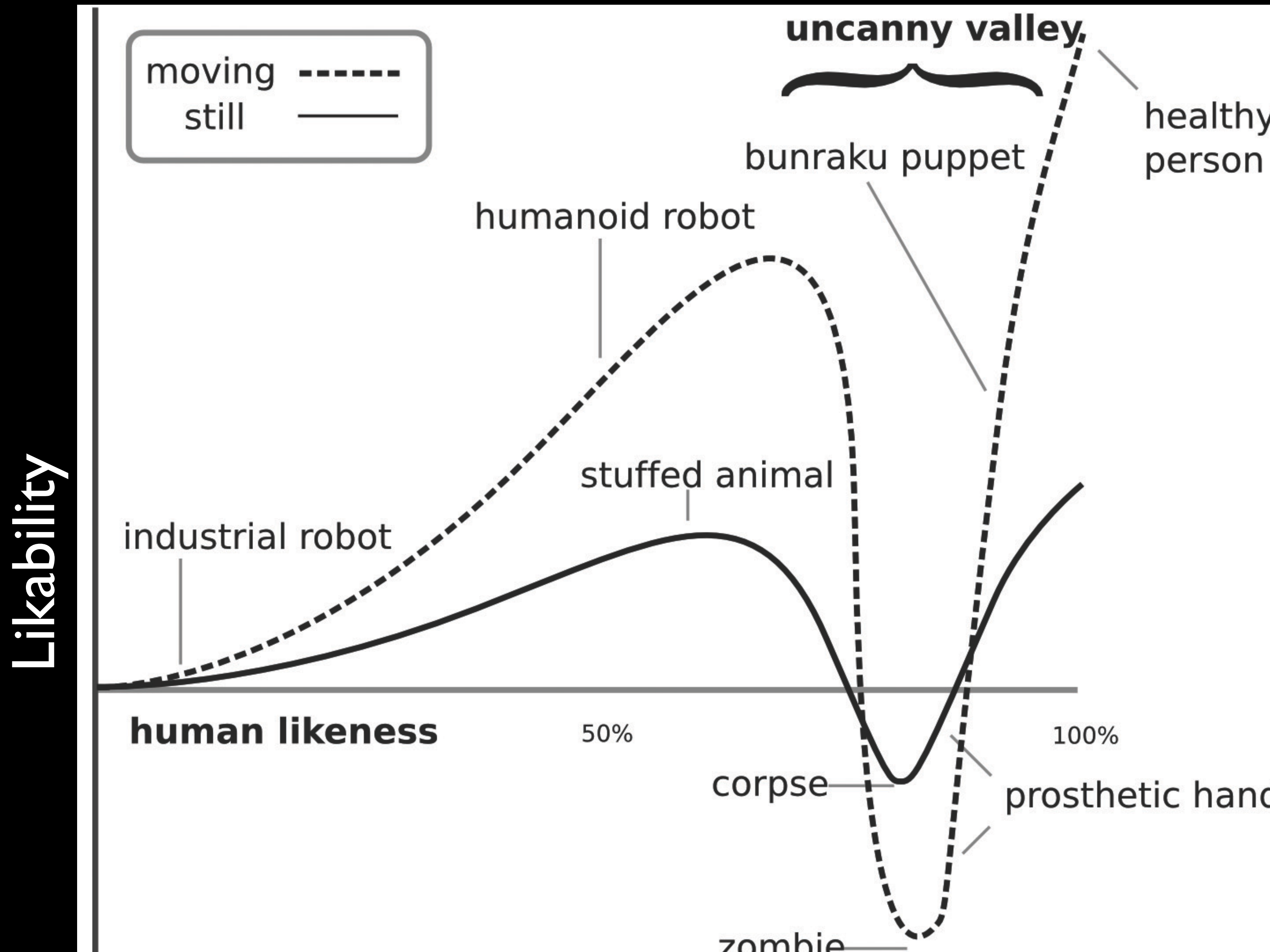
- "High dynamic range" — pixels can be arbitrarily bright or dark
- "Low dynamic range" — there are limits on the min and max

- Simplest solution: just scale and clamp
- More flexible: introduce a contrast control

- Scale factor *a* is "exposure"
  - often quoted on a power-of-2 scale


Computer graphics
Virtual
Real
Computer vision

Real-time Realism

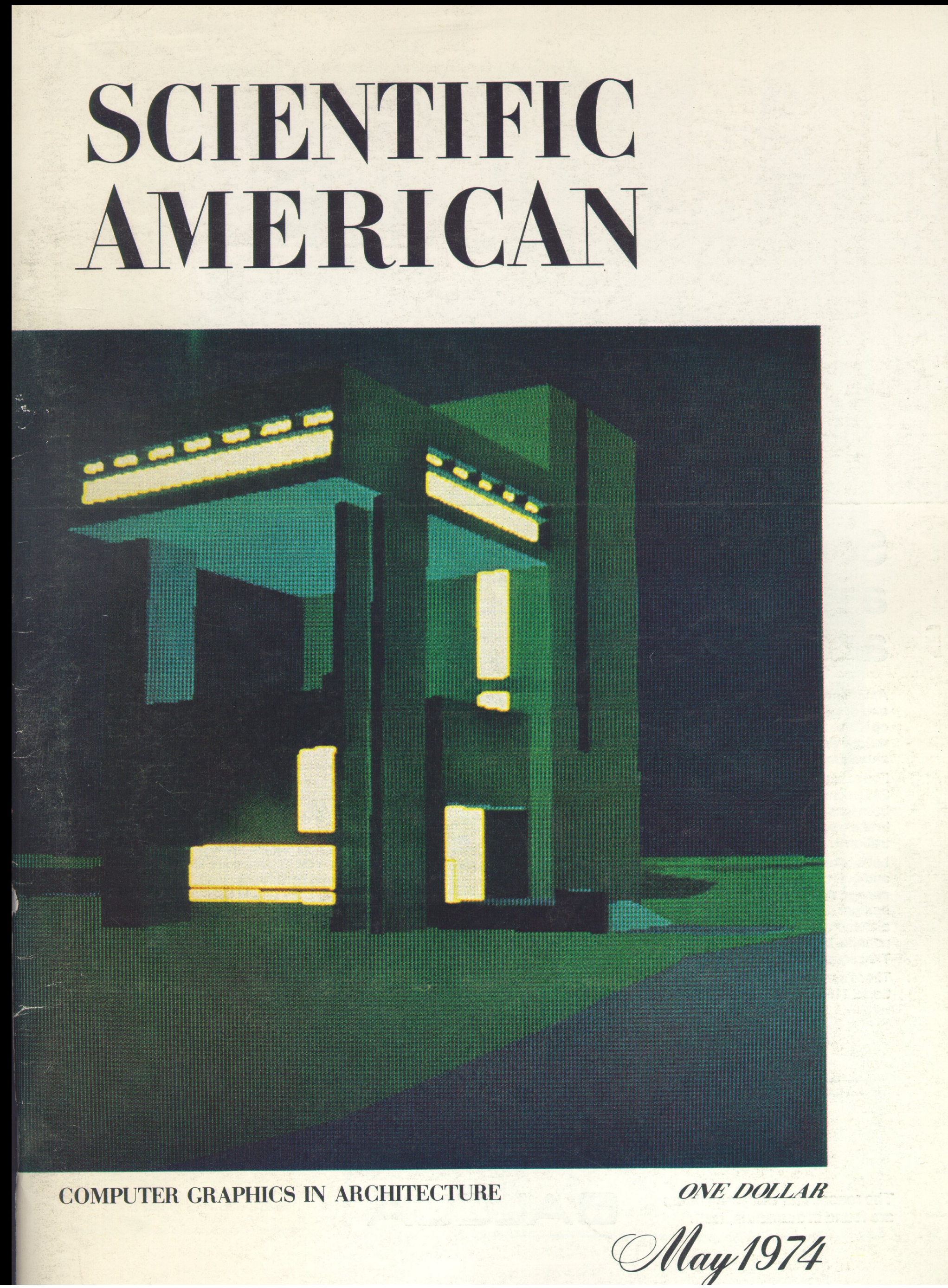


Cornell Program of Computer Graphics (PCG)

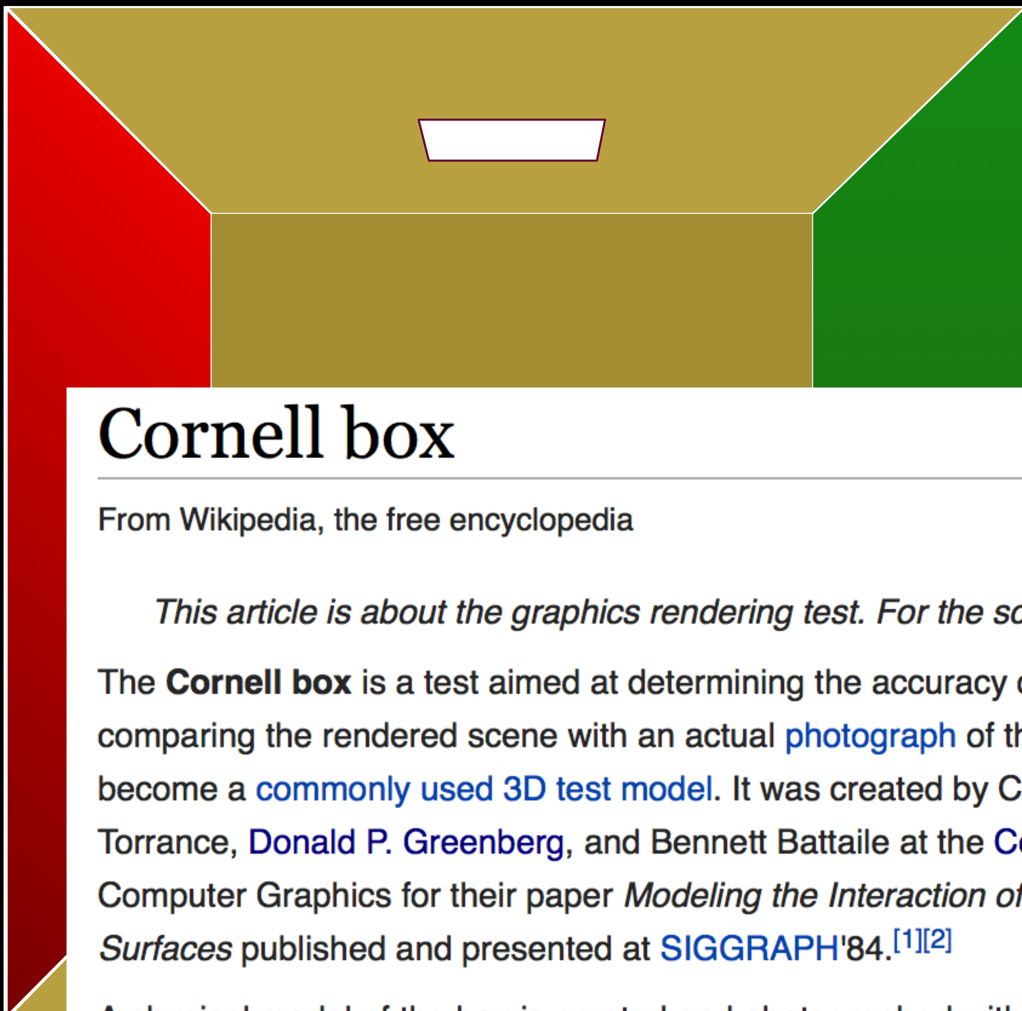
Computer Graphics in Architecture

A computer programmed to generate pictures or drawings can show a prospective building in various settings, enabling an observer to "walk" through the scene. It can also produce detailed plans

By Donald P. Greenberg



Computing light

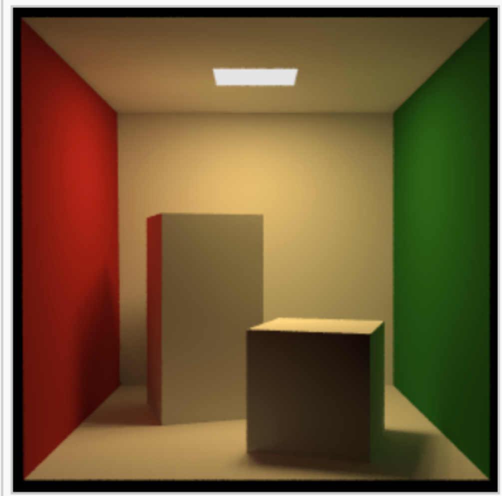


Cornell box

From Wikipedia, the free encyclopedia

This article is about the graphics rendering test. For the sculptures, see Joseph Cornell.

The **Cornell box** is a test aimed at determining the accuracy of rendering software by comparing the rendered scene with an actual photograph of the same scene, and has become a commonly used 3D test model. It was created by Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile at the Cornell University Program of Computer Graphics for their paper *Modeling the Interaction of Light Between Diffuse Surfaces* published and presented at SIGGRAPH'84.[1][2]

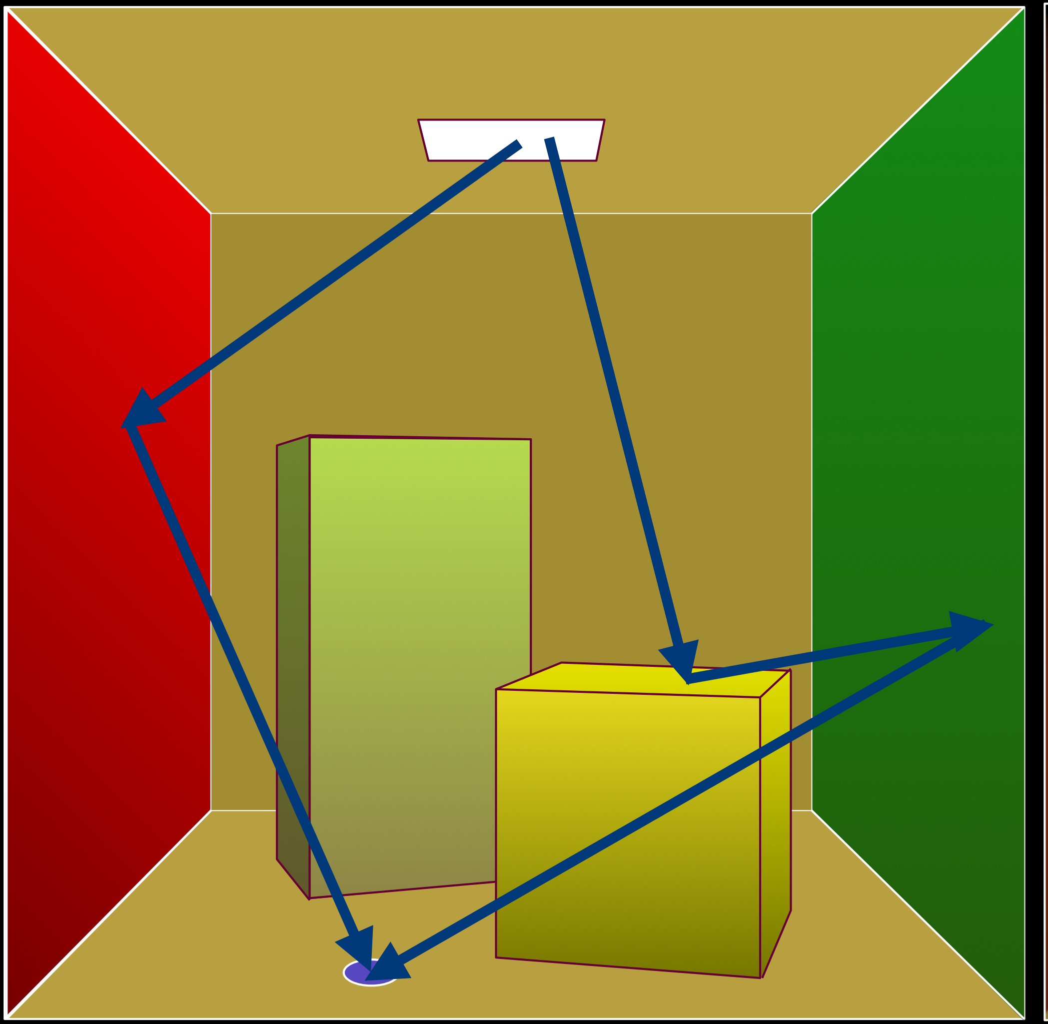


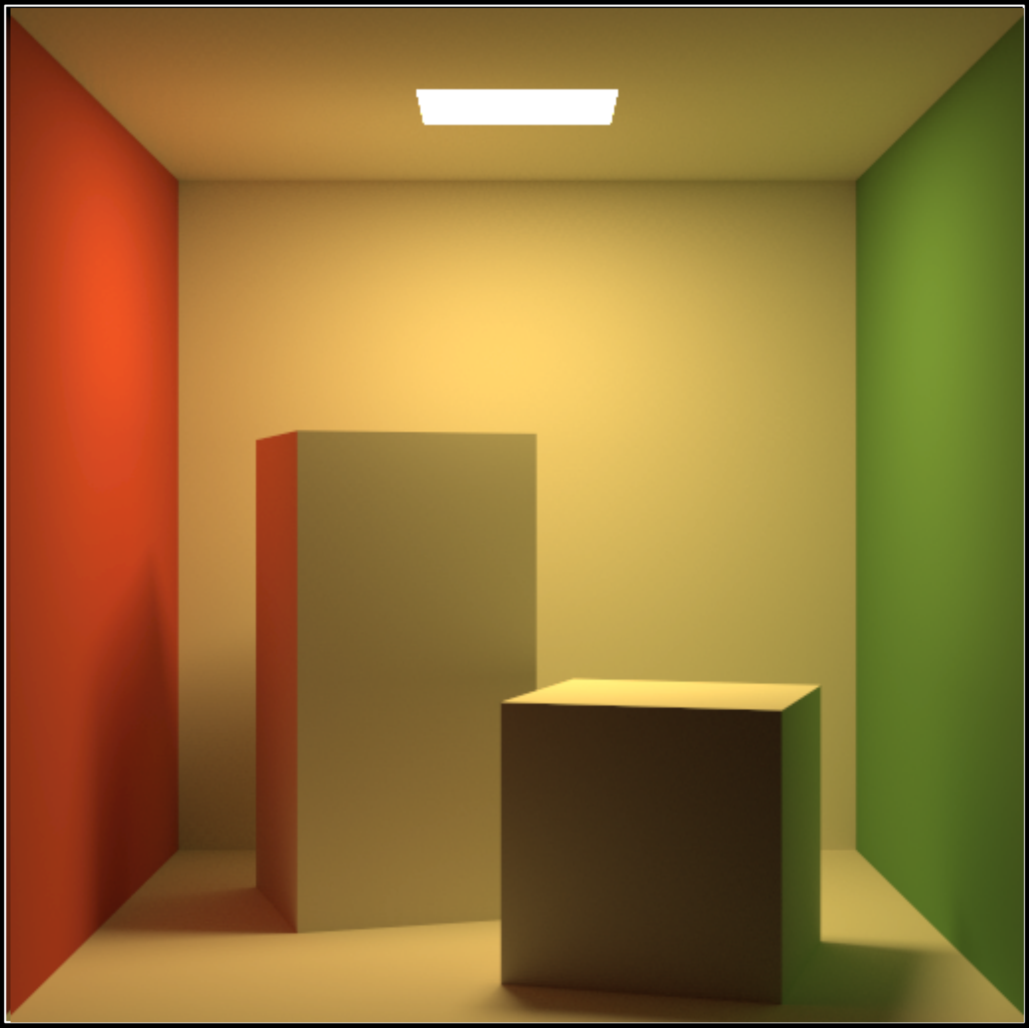
Standard Cornell box rendered with POV-Ray

A physical model of the box is created and photographed with a CCD camera. The exact settings are then measured from the scene: emission spectrum of the light source, reflectance spectra of all the surfaces, exact position and size of all objects, walls, light source and camera.

The same scene is then reproduced in the renderer, and the output file is compared with the photograph.

Computing light

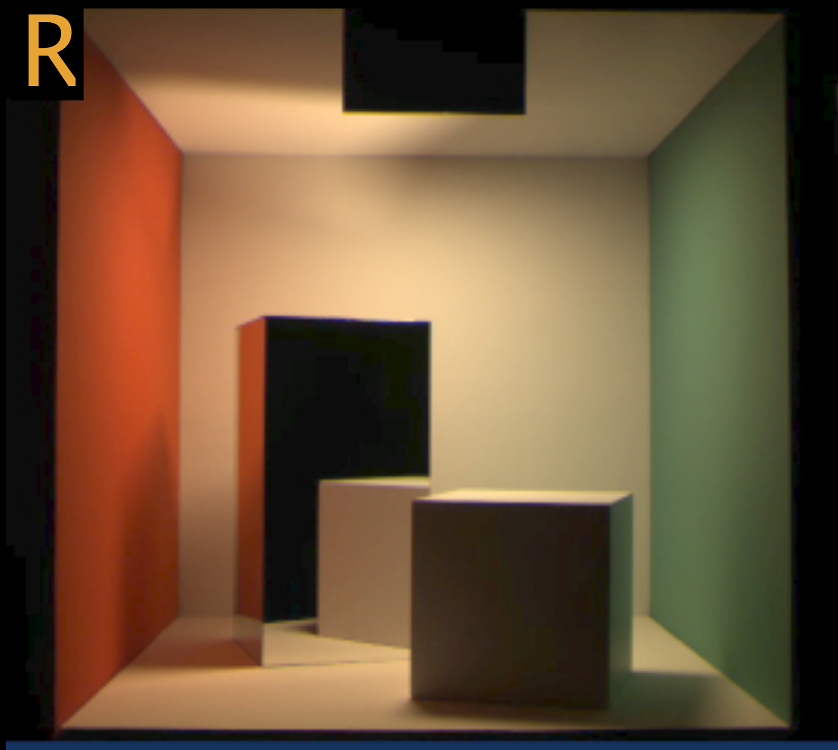




Global illumination

Which is the virtual image, which is the real Cornell Box?

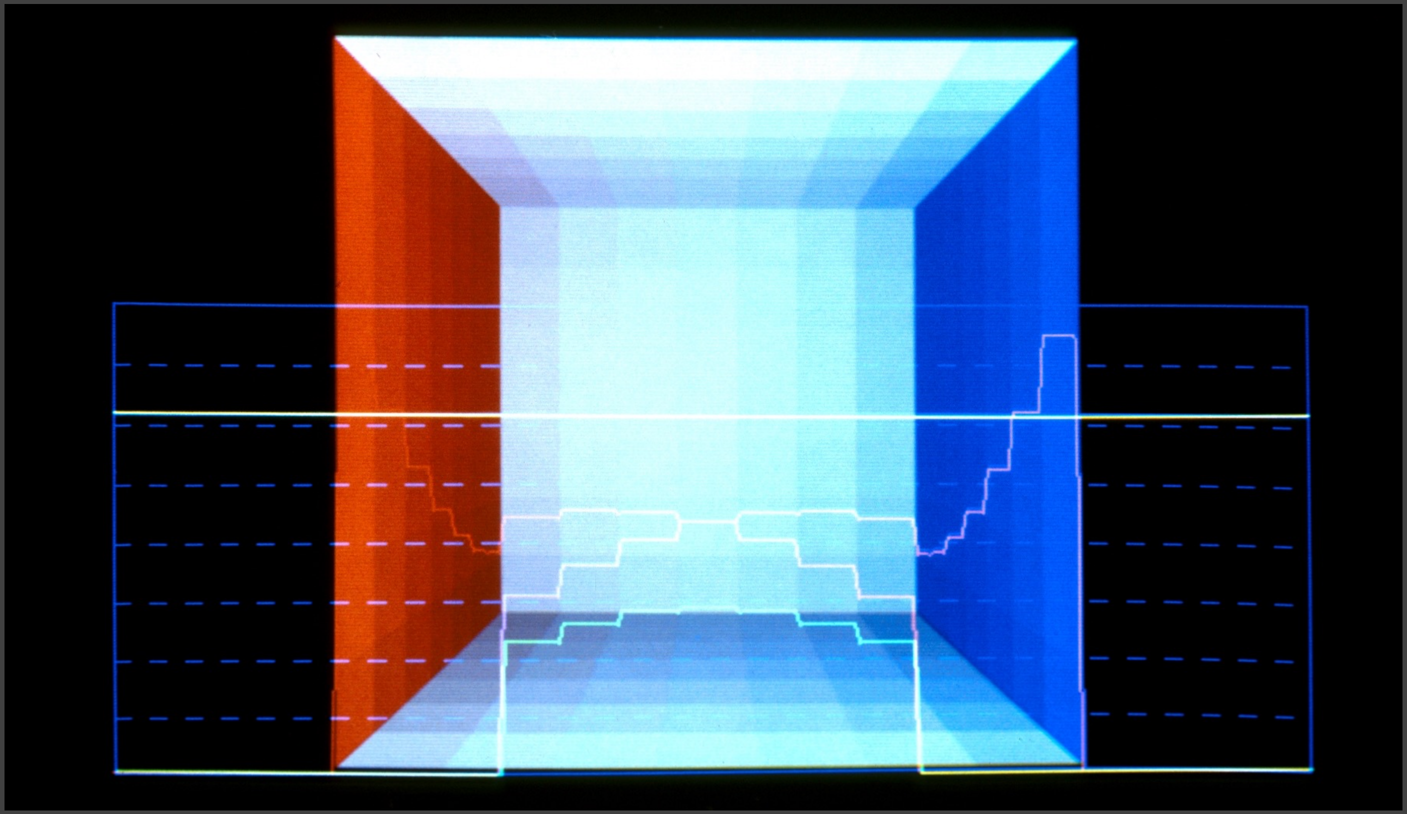
R



V

Radiosity '84

Radiosity



Goral, Torrance, Greenberg, Battaile, 1984

Notable commercial radiosity engines are Enlighten by Geomerics, used for games including Battlefield 3 and Need for Speed: The Run, 3D Studio Max, form•Z, LightWave 3D and the Electric Image Animation System.

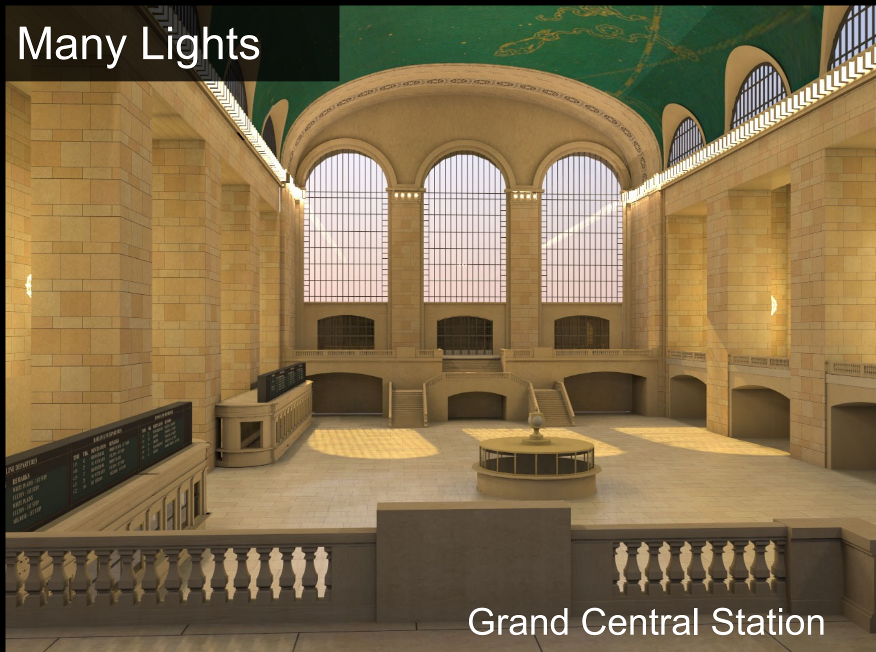
My research: Modeling and Rendering World

- Challenge: world is complex
  - Complex datasets: micron resolution data to building size data
  - Complex lighting: volumetric scattering, global illumination

What do I do?

- Predictive Rendering for high complexity

The Complexity Challenge

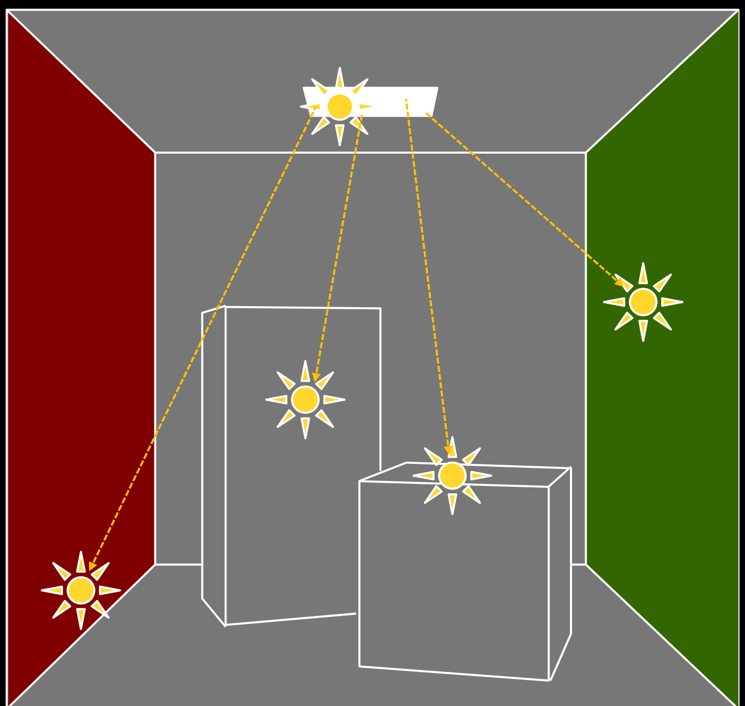


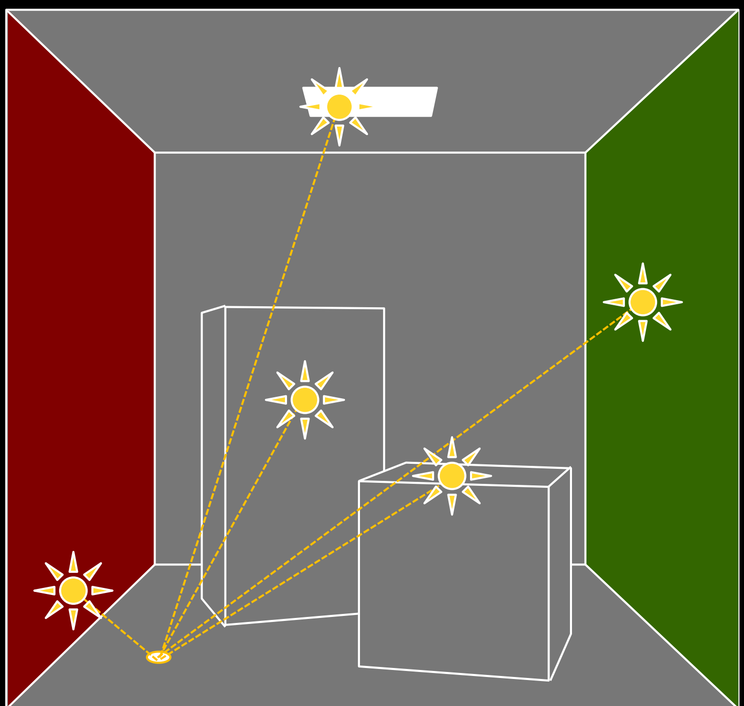
Rendering Problem Formulation

- **Based on Instant Radiosity** ***[Keller 1997]***
- Approximate indirect illumination by

  Virtual Point Lights (VPLs)

1. Generate VPLs
2. Render with VPLs

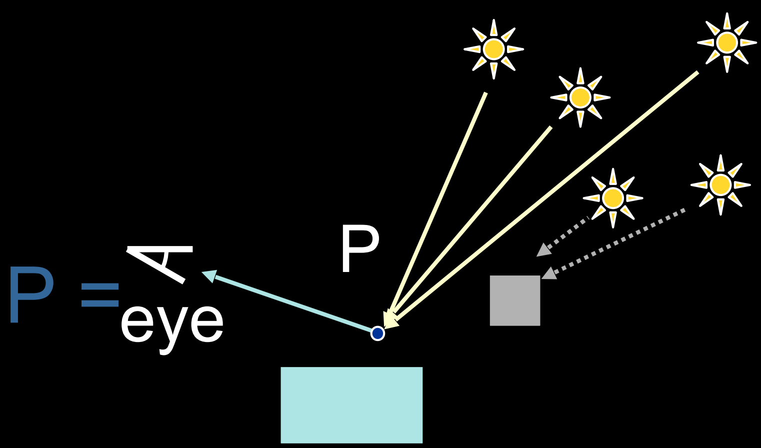




Problem: Many-light formulation

- Unify illumination
- Convert to point lights

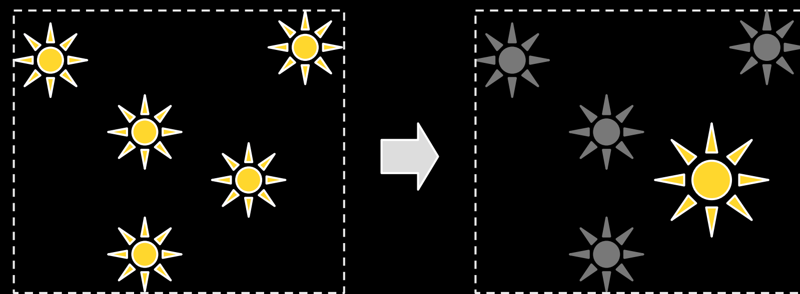
$$\mathrm{P} = \sum_{i \in \mathrm{Lights}} k_i I_i$$

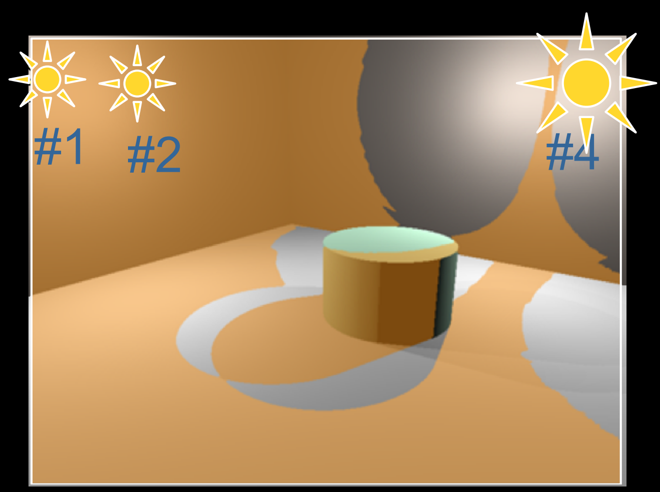


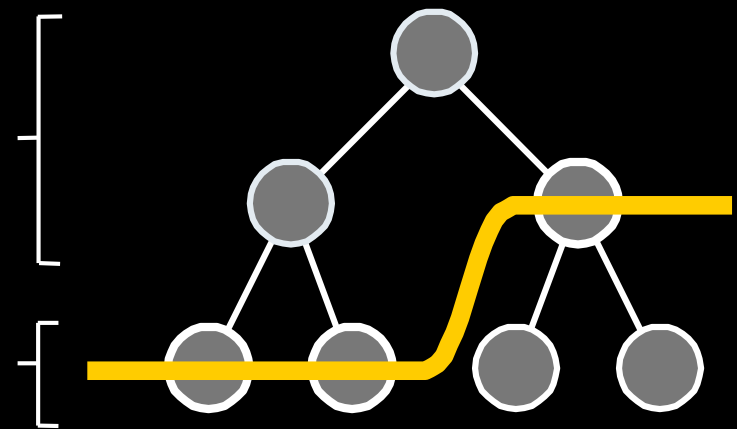
Kitchen light: area, sun/sky, indirect

Lightcuts [SIG'05,'06,SIG'12]

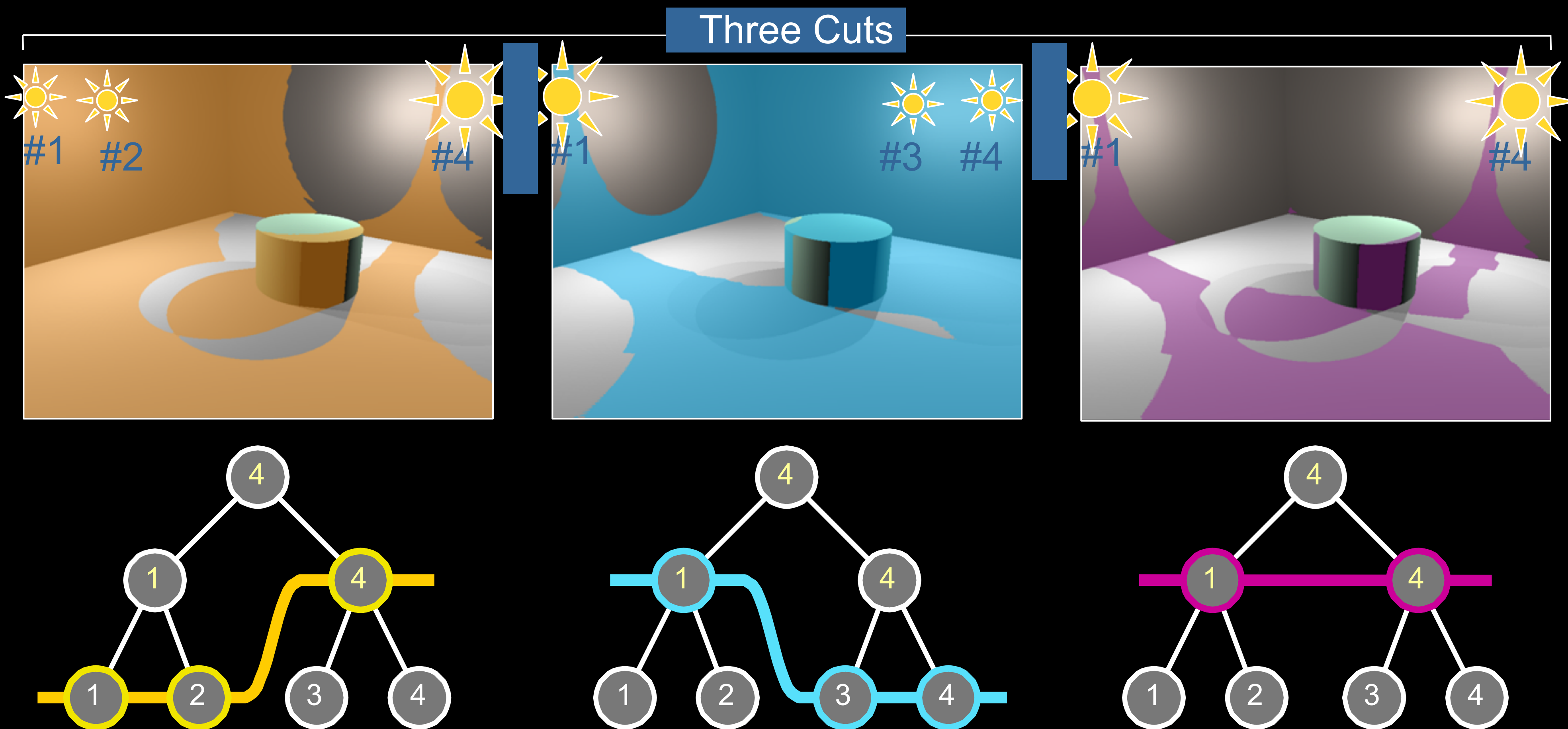
- Scalable rendering of many lights
- Light Cluster
- Light Tree
- Cut
  - Set of nodes that partition L into clusters
    - (a representative per cluster)





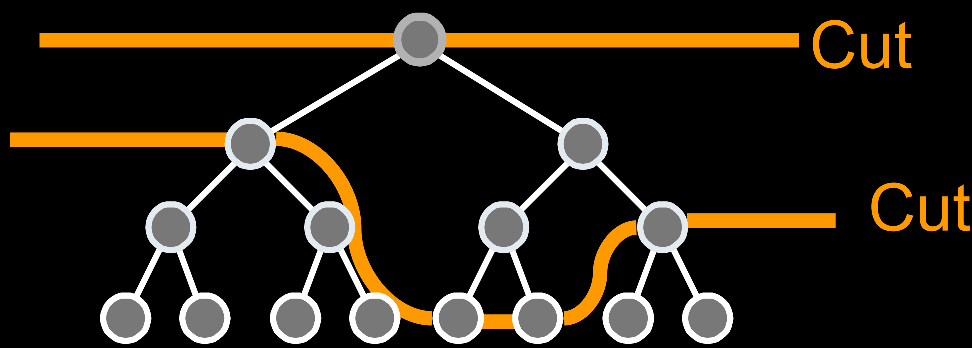


Cuts and Representatives

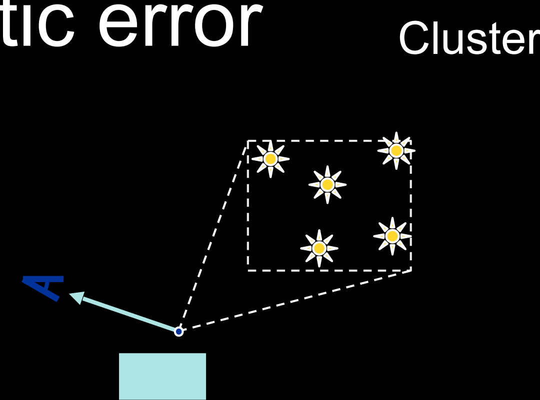


Cut Selection Algorithm

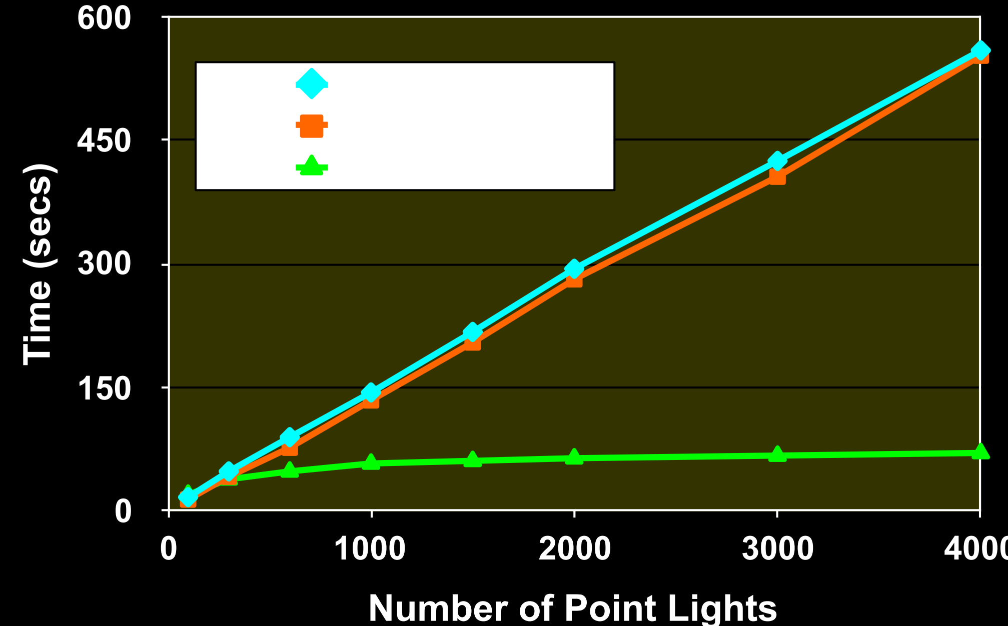
- For each point, find cut that is perceptually good
- Start with coarse cut (eg, root node)



- Test visibility of cluster representative (and estimate contribution)
- Estimate error using conservative analytic error bounds
  - Refine if error bound > perceptual metric
  - Perceptual metric
    - Weber's law: 2% of total energy



Lightcuts Results



Grand Central

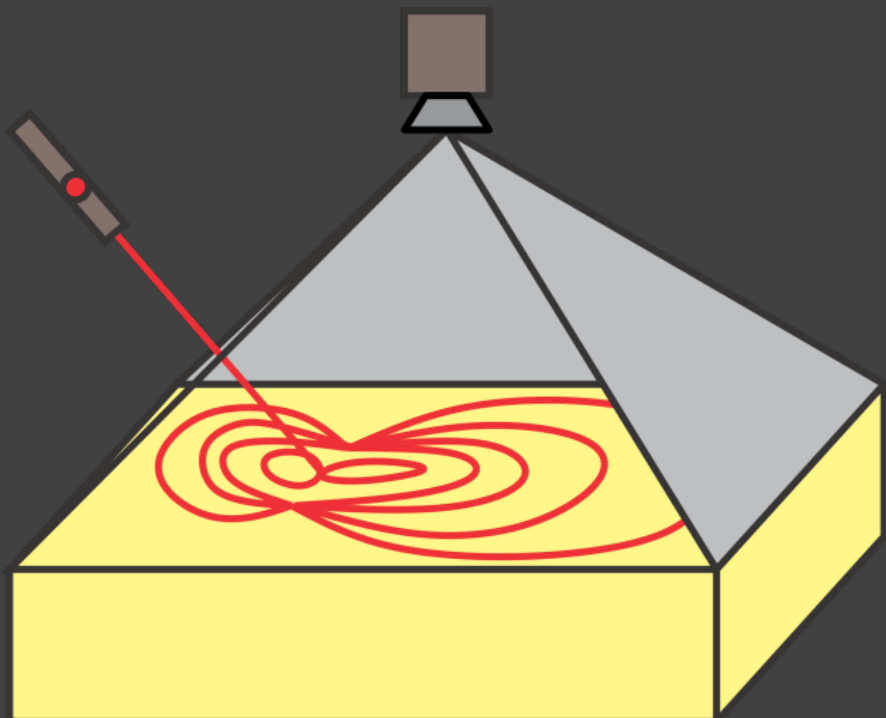
1.46M polygons, 143464 lights

Area+Sun/sky+Indirect

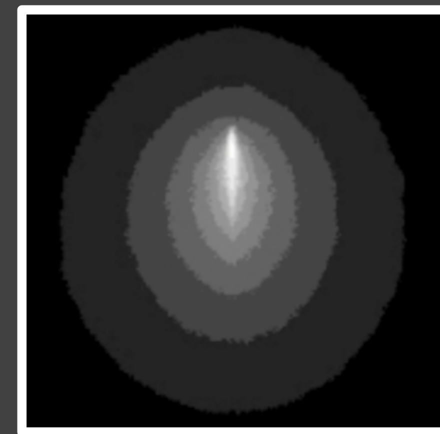
Shadow rays/ point: 46 (0.03%)

What makes
jade look like jade?

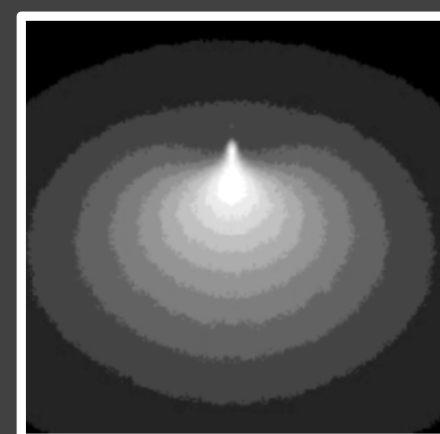
Answer: how light scatters matters



Shine laser and capture photo



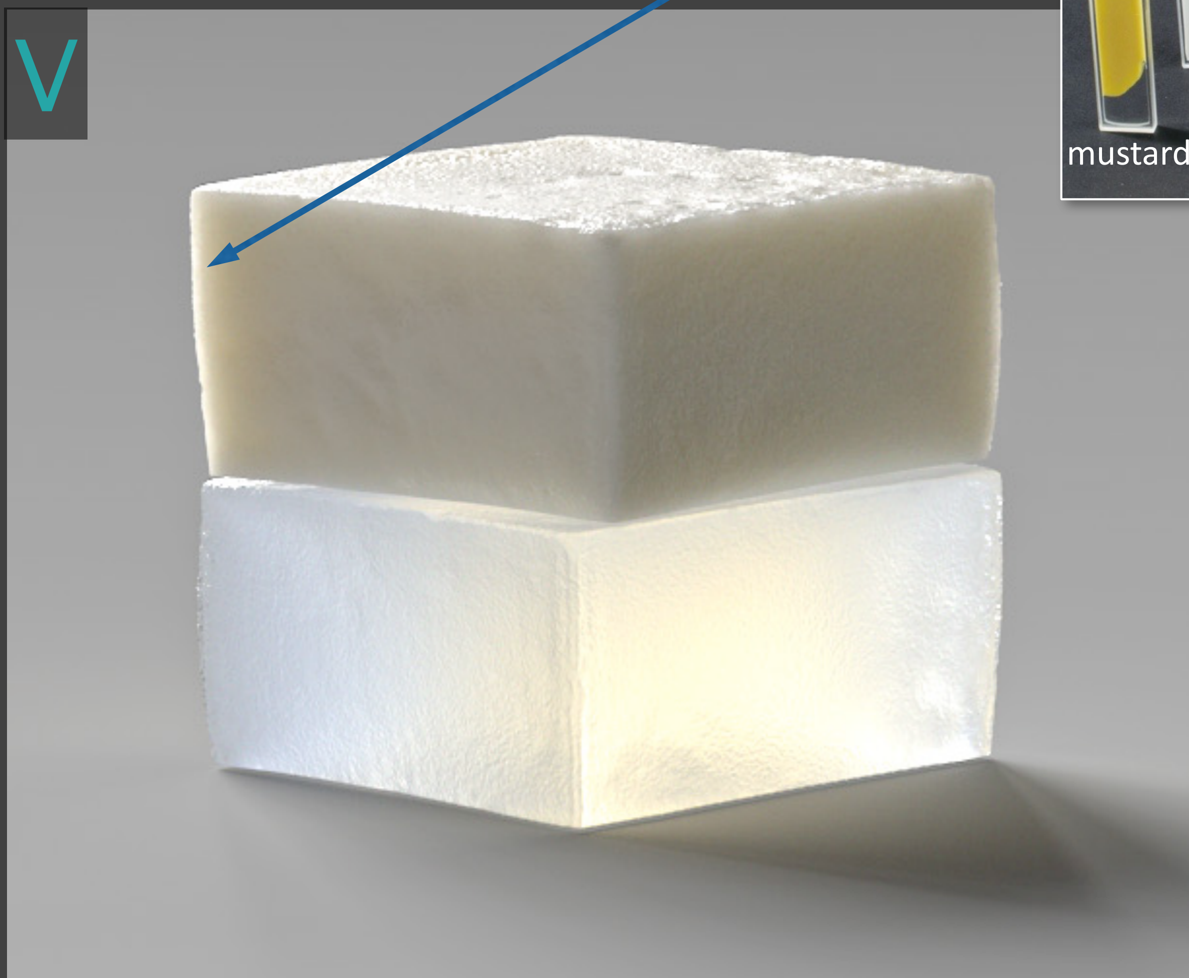
Soap



Wax

w/ Adelson, Gkioulekas, Xiao, Zhao, Zickler

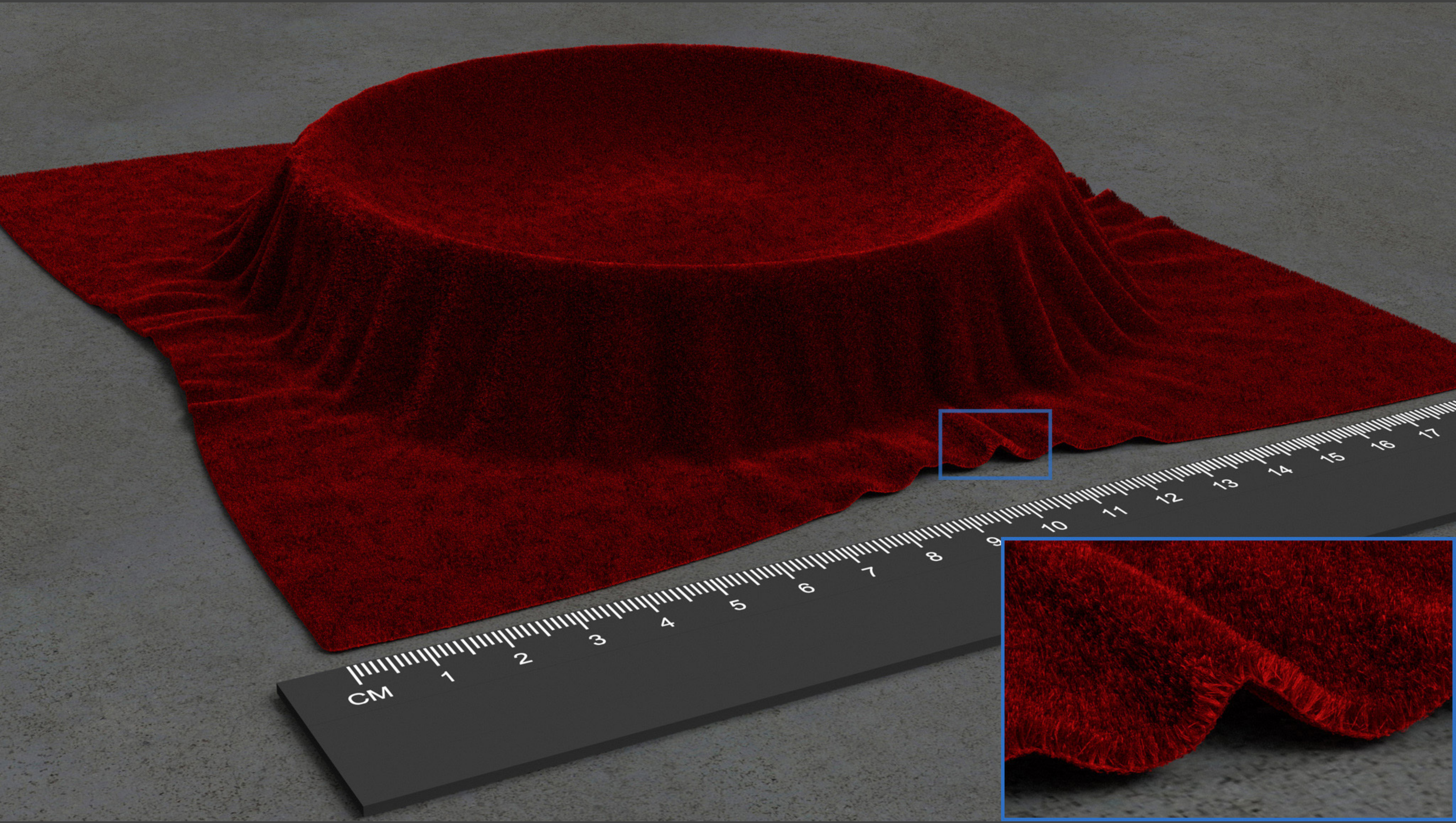
Answer: how light scatters matters



w/ Adelson, Gkioulekas, Xiao, Zhao, Zickler

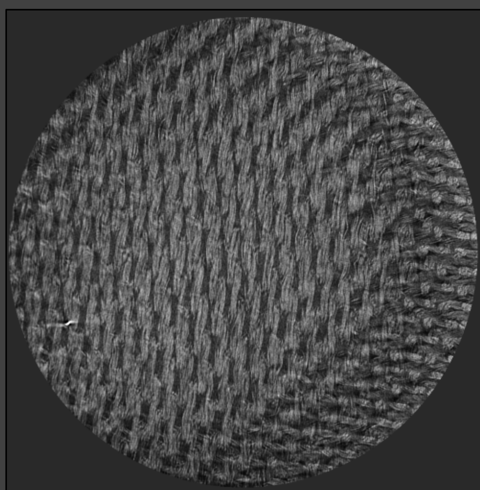
What makes
silk look like silk?
velvet look like velvet?

What makes velvet look like velvet?



[SIG11, SIG12, SIG13, CACM Research Highlight 2013,TOG15]

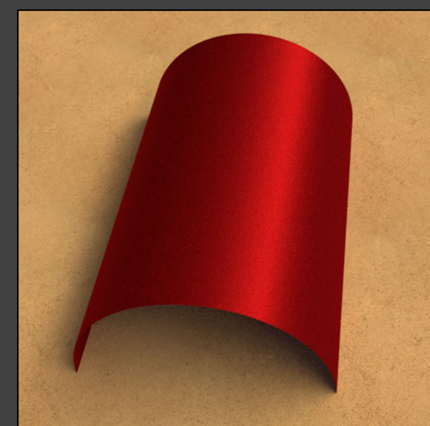
Model = structure + photos



Micro CT image

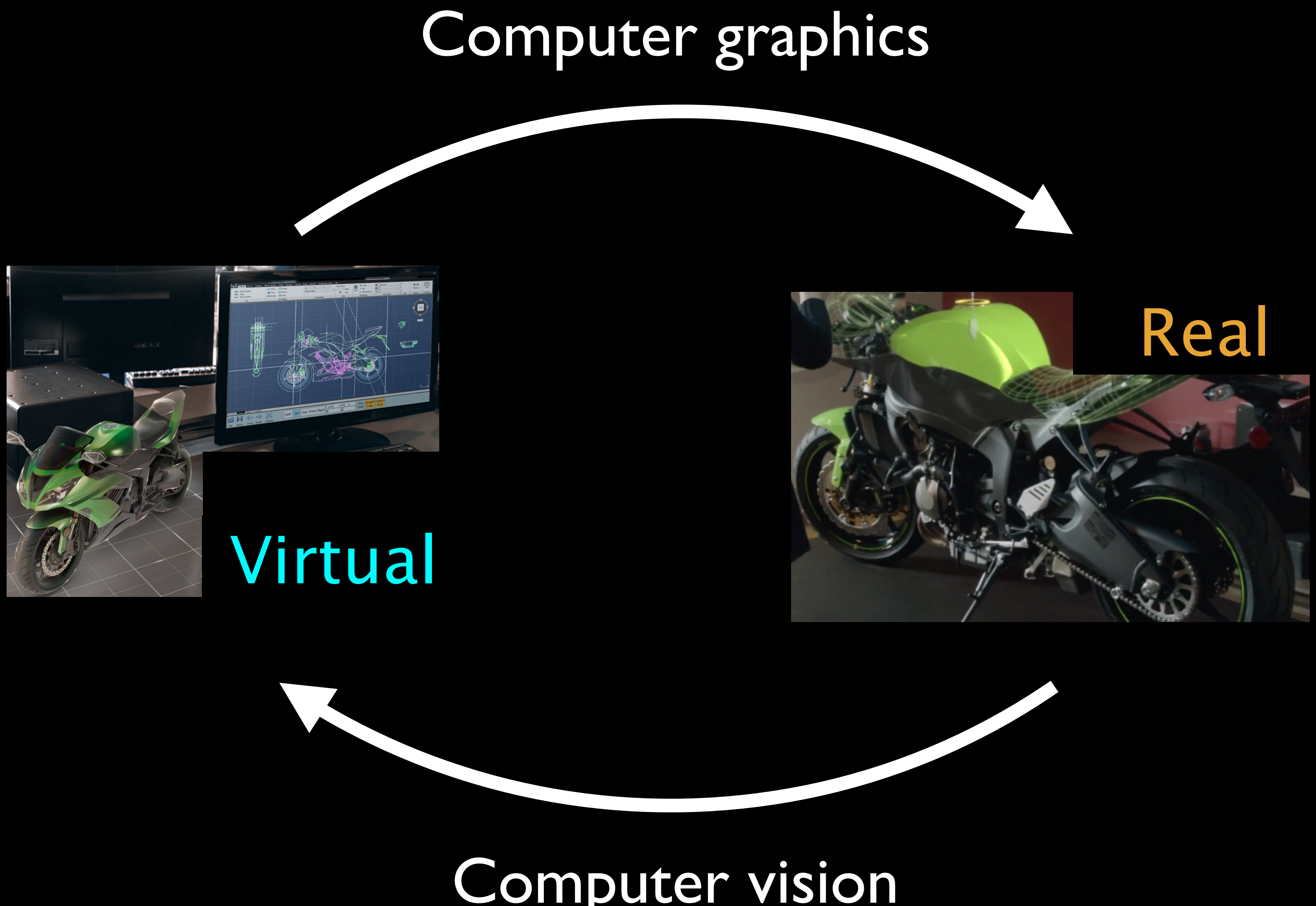
Extend

+ photo to match optical properties



Fabric model

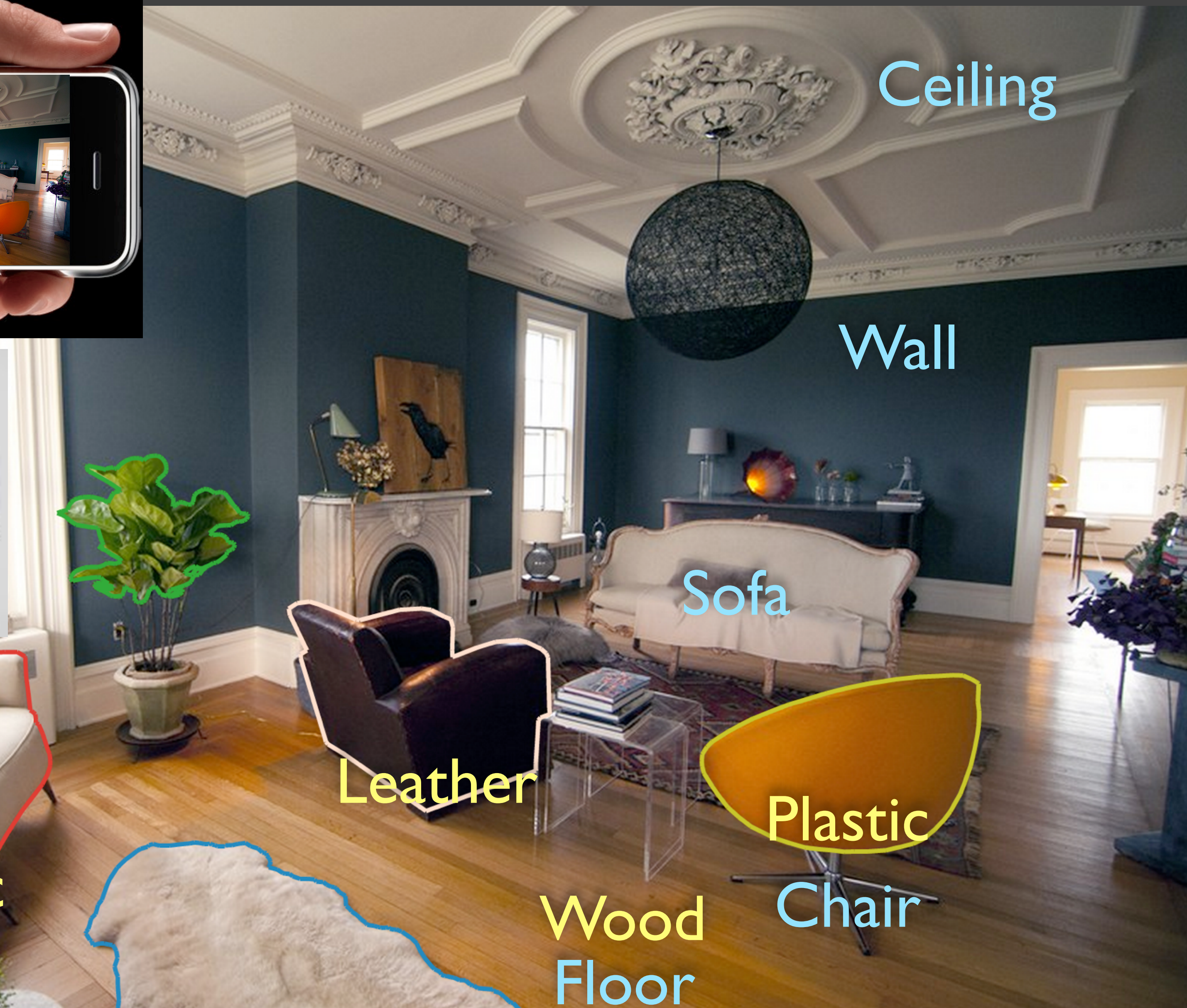
w/ Zhao, Marschner, Jakob


Computer graphics
Real
Virtual
Computer vision

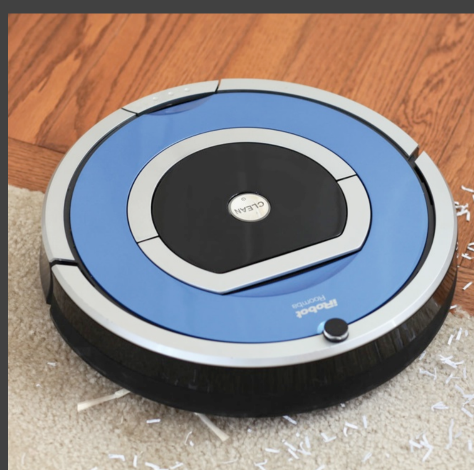
Materials in the Wild

Goal: to recognize, model and render materials in the wild

Materials in the Wild: understanding



Materials in the Wild: recognition



Want to learn more?

- 4670: Computer vision
  - images
  - 3D reconstruction
  - deep learning for scene understanding

- 5625: Interactive computer graphics
  - Shadow maps/shadow volumes
  - Texturing: theory, advanced
  - Subdivision surfaces
  - Some animation

Where do we go from here?

- Industry
  - VR: Oculus, Valve, …
  - Tech: Intel, NVidia, Microsoft,…
  - Movies: Dreamworks, Pixar, Disney…
  - Games: EA, Epic, Bungie, …
  - CAD/CAM: Boeing, Autodesk,…

- Graduate school

Thank you!