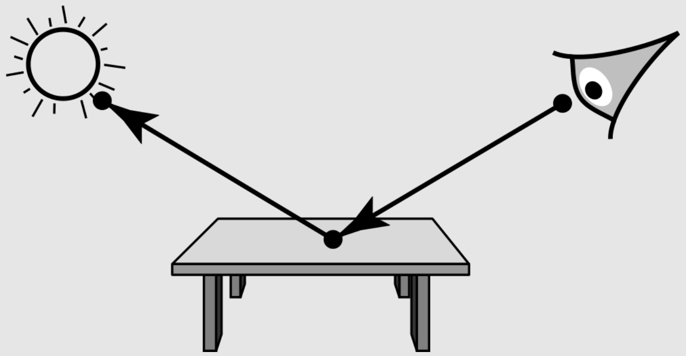
# More Variance Reduction and Non-Photorealistic Rendering

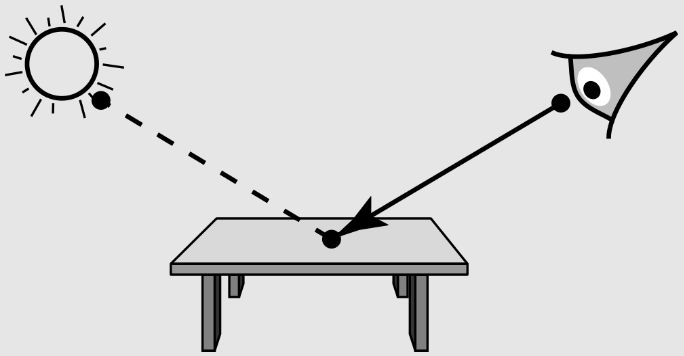
- ~~Monte-Carlo Sampling~~
- ~~Biased vs Unbiased Estimators~~
- Physically-Based Rendering Methods

# Previous Methods



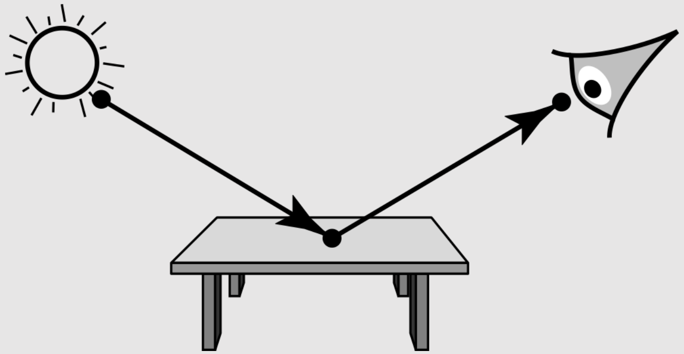
[ backward path tracing ]

Fails: cannot intersect point lights



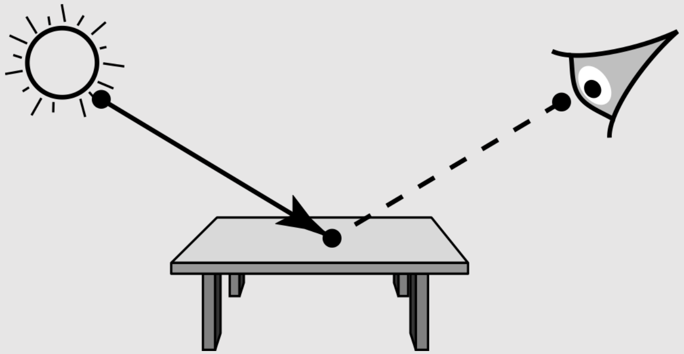
[ backward path tracing + connect to light ]

Works: reaches point lights



[ forward path tracing ]

Fails: cannot intersect pinhole camera

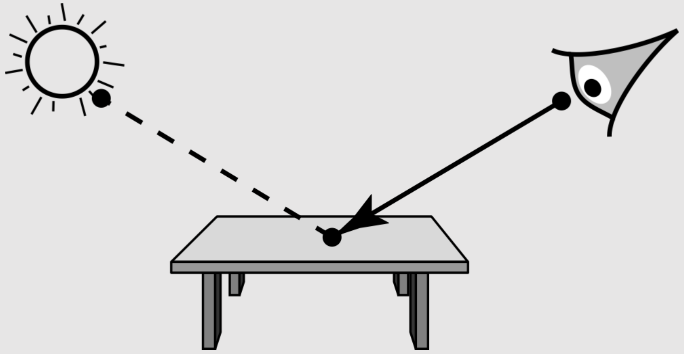


[ forward path tracing + connect to camera ]

Works: reaches pinhole camera

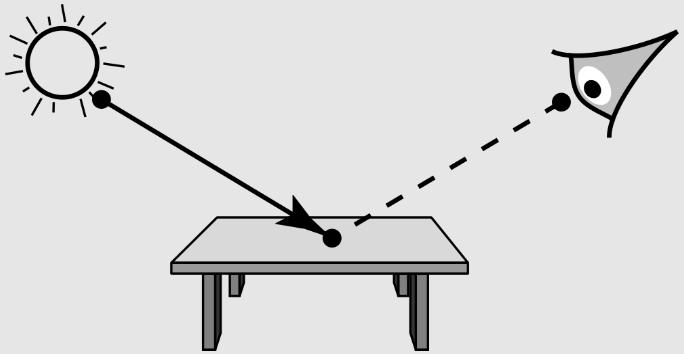
# Path Tracing Can Be Biased

- Deliberately connect terminating rays to light (forward) or camera (backward)
- Probability of sampling a ray that hits a non-volume source (point light, pinhole camera) is 0
    - We bias our renderer by choosing those rays



**[ backward path tracing + connect to light ]**

**works: reaches point lights**



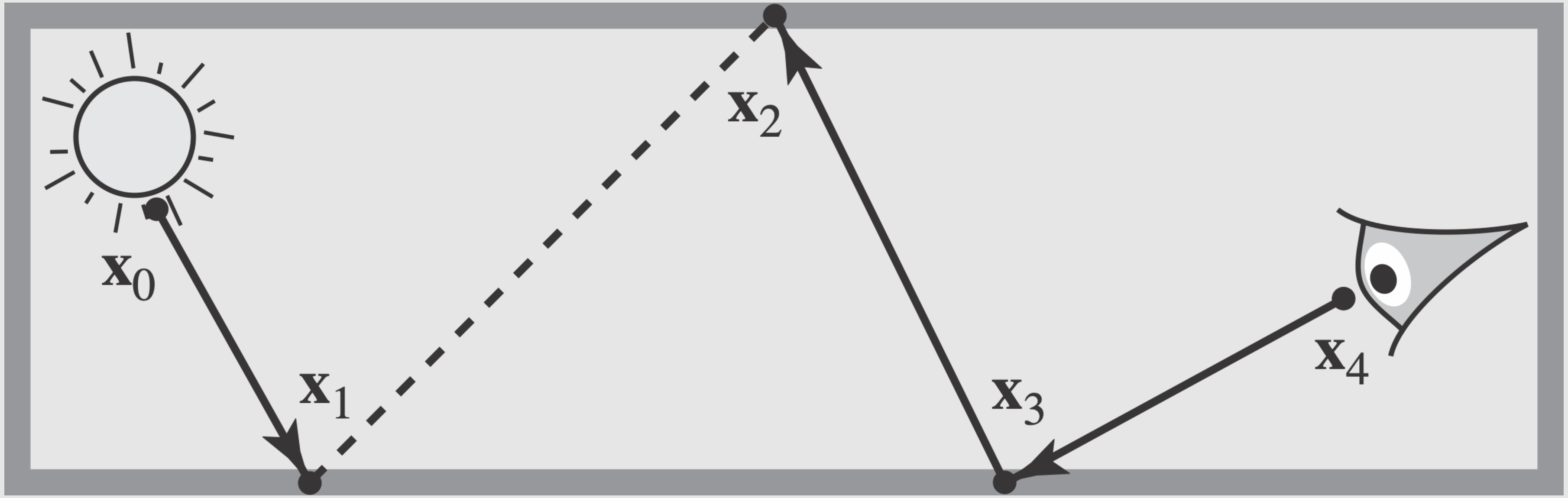
**[ forward path tracing + connect to camera ]**

**works: reaches pinhole camera**

# Bidirectional Path Tracing

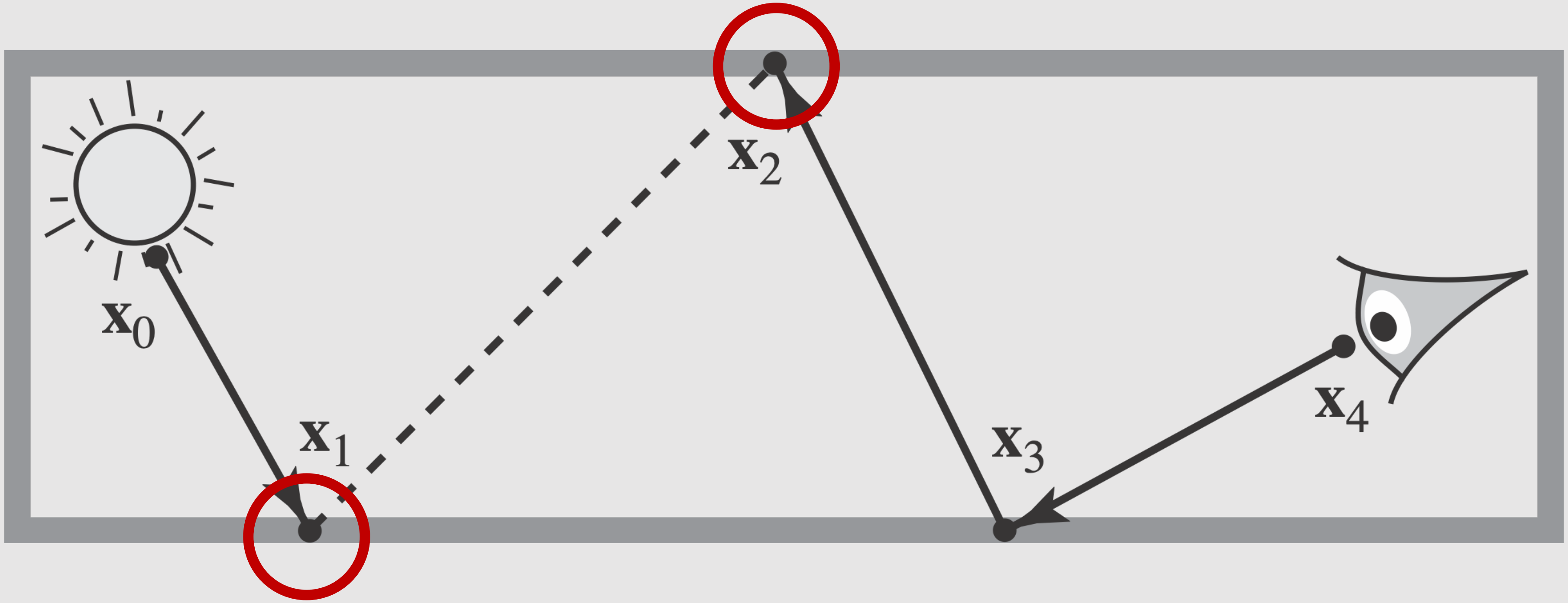
- If path-tracing is so great, why not do it **twice?**
  - Main idea of bidirectional!
- Trace a ray from the camera into the scene
- Trace a ray from the light into the scene
  - Connect the rays at the end

- Unbiased algorithm
  - No longer trying to connect rays through non-volume sources
- Can set different lengths per ray
  - Example: Forward m = 2, Backward m = 1



# Bidirectional Path Tracing

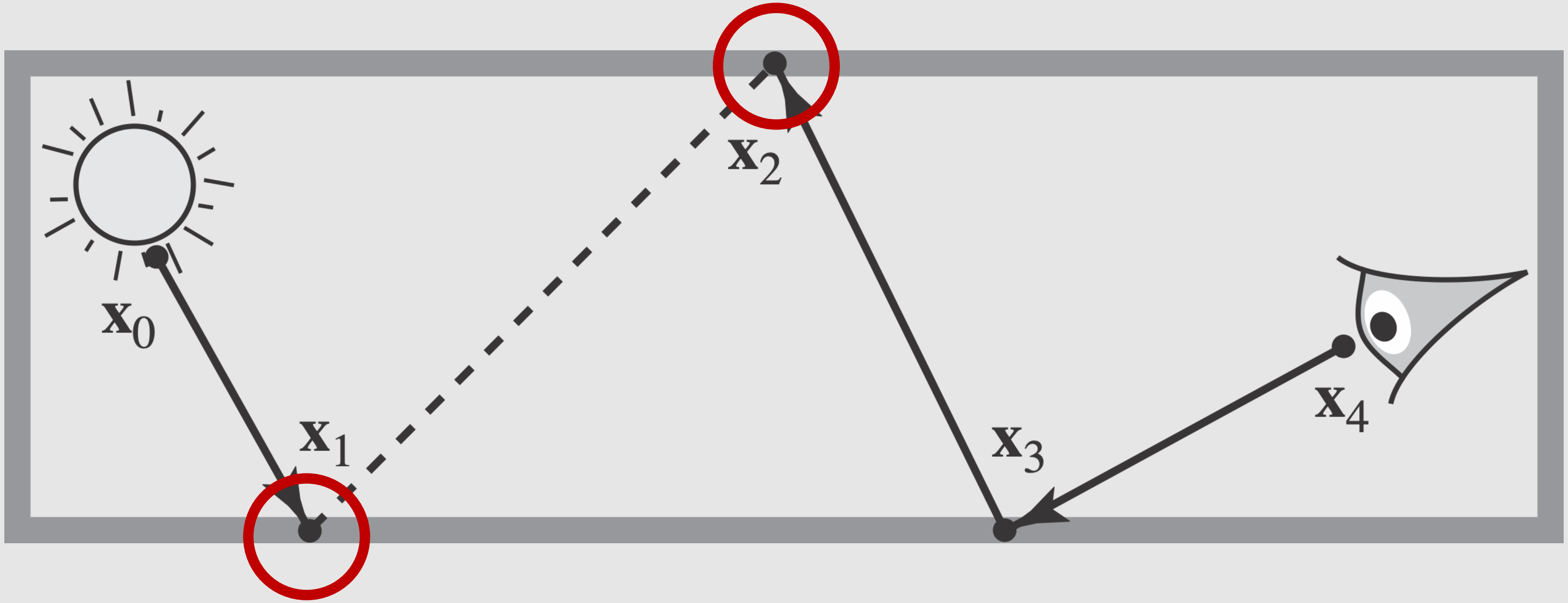
Issue: what if these are mirrors!



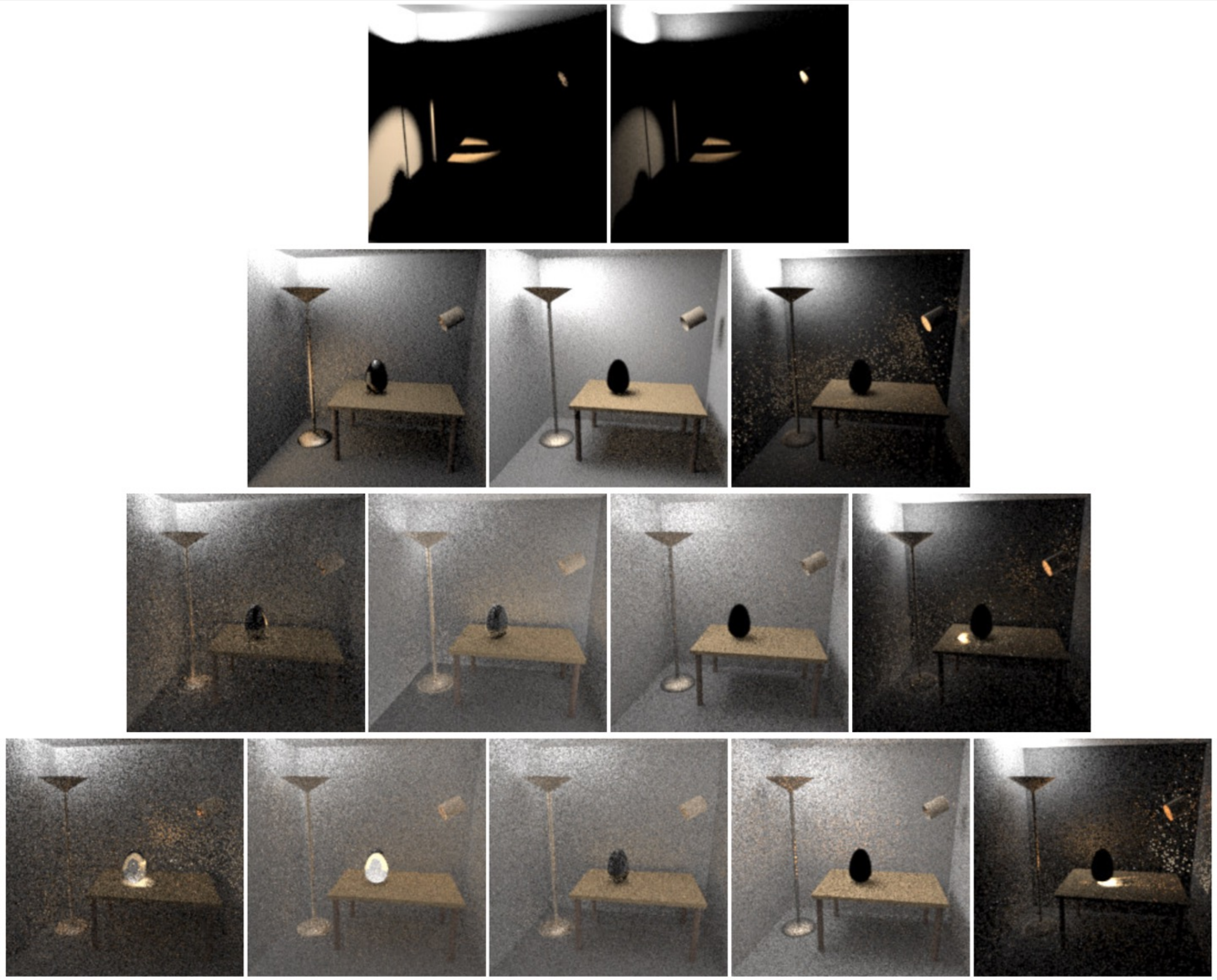
# Bidirectional Path Tracing

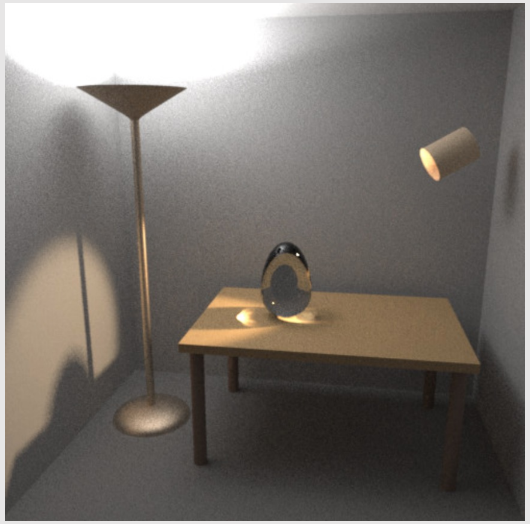
- In cases of mirrors, we cannot choose any ray path
- Instead, continue tracing rays until diffuse surfaces are reached on both rays

**Issue: what if these are mirrors!**



# Bidirectional Path Tracing

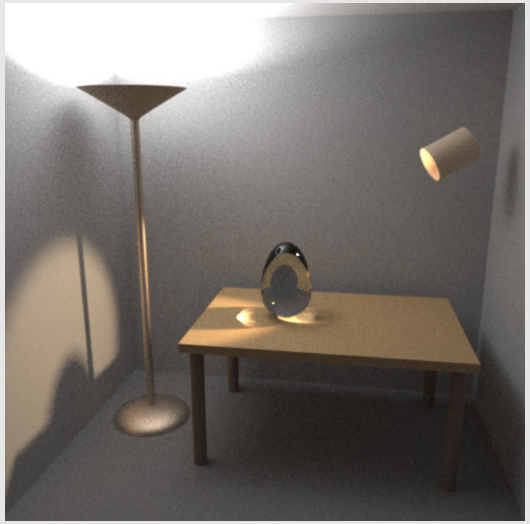




**[ final image ]**

- Each row shows path length
- As we move over images in a row, we decrease forward ray depth and increase a backward ray depth
    - Overall length kept constant per row

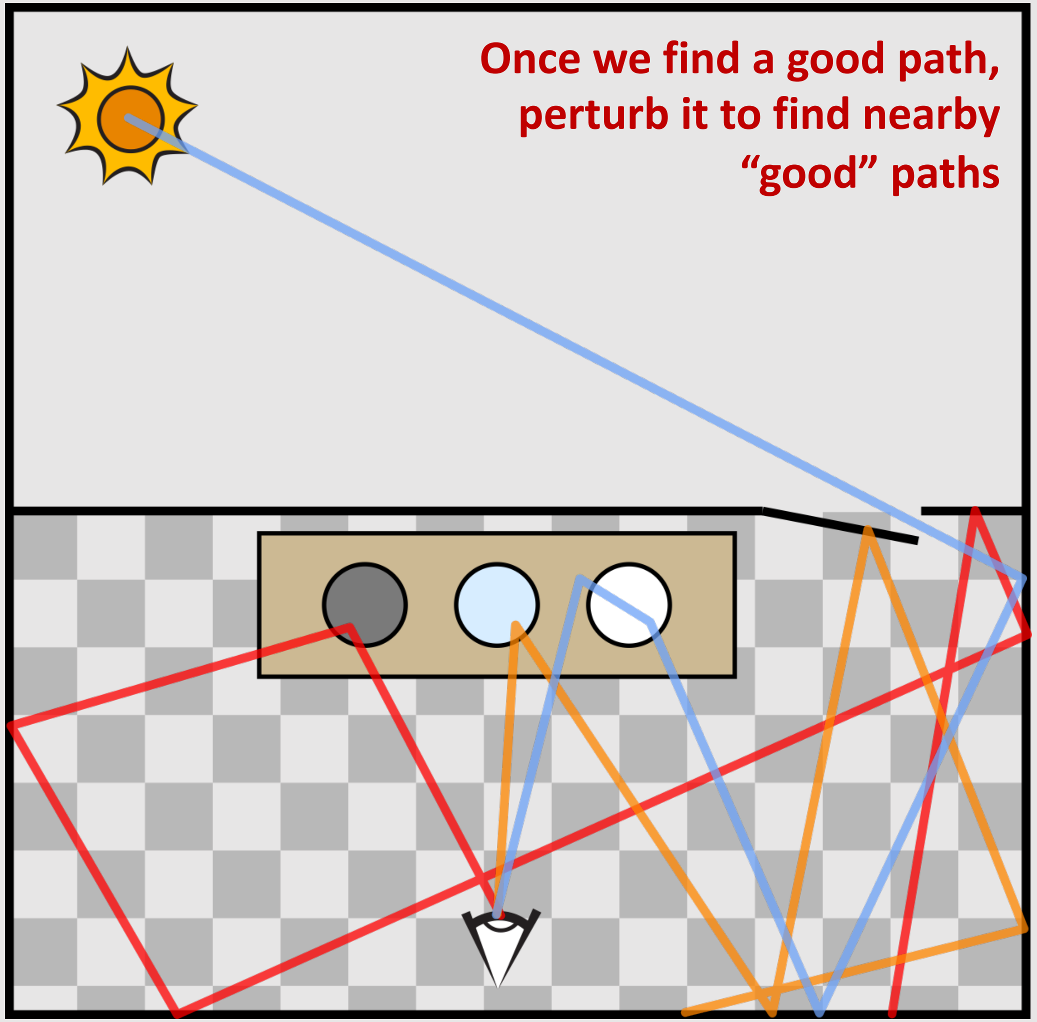
# Bidirectional Path Tracing

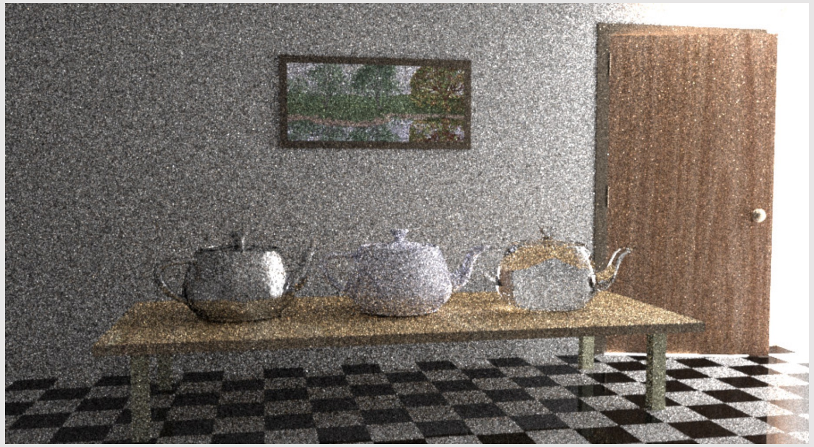


**[ final image ]**

- Not easy to tell which path lengths work well for a scene!
  - The glass egg is illuminated at specific path lengths for forward and backward rays

# Good Paths Are Hard To Find



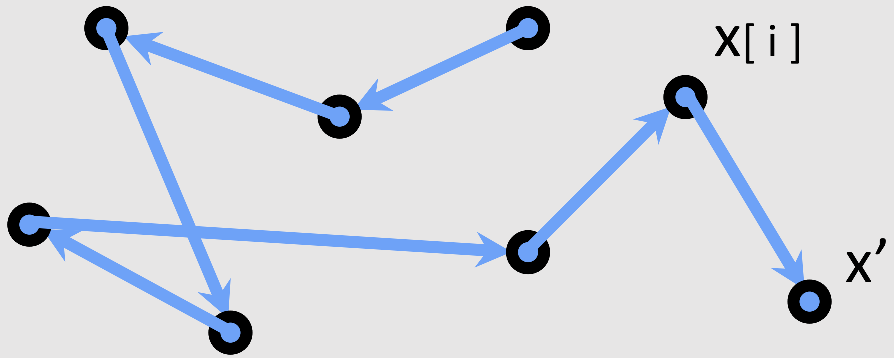
[ Bidirectional Path Tracing ]

[ Metropolis Light Transport ]

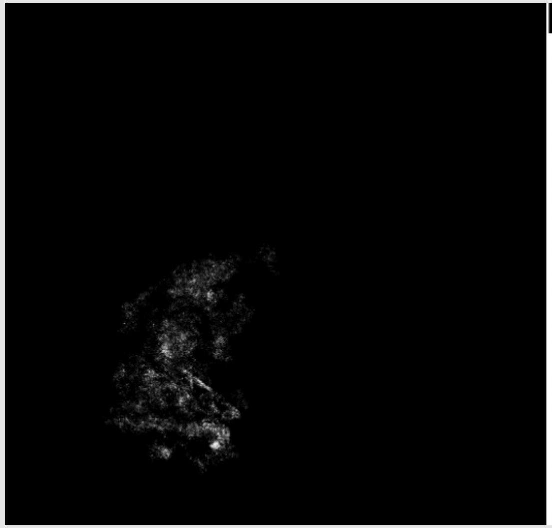
# Metropolis Hasting Algorithm

- *"Once we find a good path, perturb it to find nearby 'good' paths"* – previous slide
- **Algorithm:** take random walk of dependent samples
  - If in an area where sampling yields high values, stay in or near the area
    - Otherwise move far away
- Sample distribution should be proportional to integrand
  - Make sure mutations are "ergodic" (reach whole space)
  - Need to take a long walk, so initial point doesn't matter

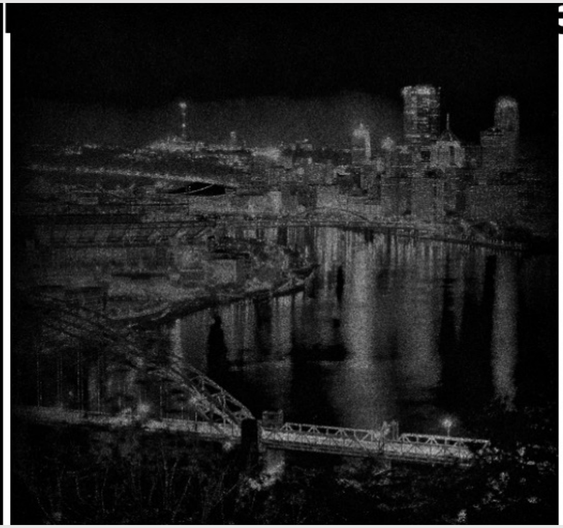
```
float r = rand();
// if f(x') >> f(x[i]), then a is large
// and we increase chances of moving to x'
// if f(x') << f(x[i]), then a is small
// and we increase chances of staying at x
float a = f(x')/f(x[i]);
if (r < a)
  x[i+1] = x';
else
  x[i+1] = x;
```



# Metropolis Hasting: Sampling An Image



[ short walk ]



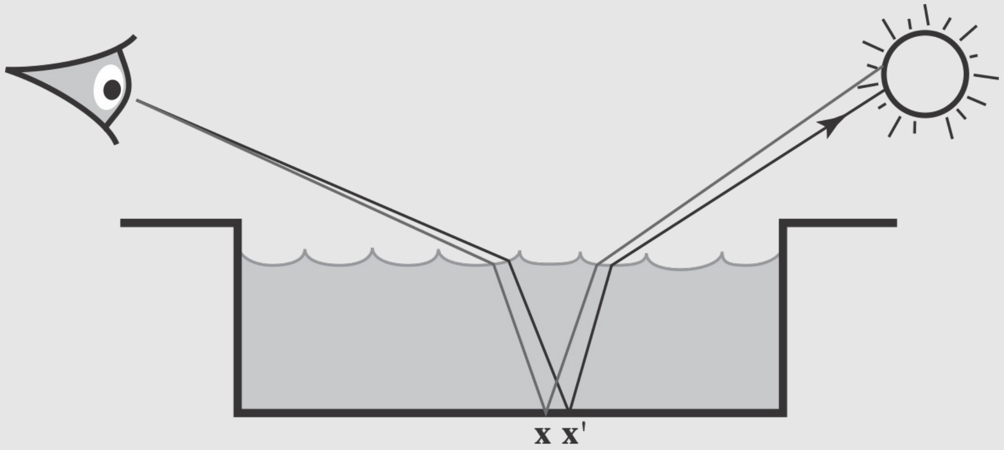
[ long walk ]

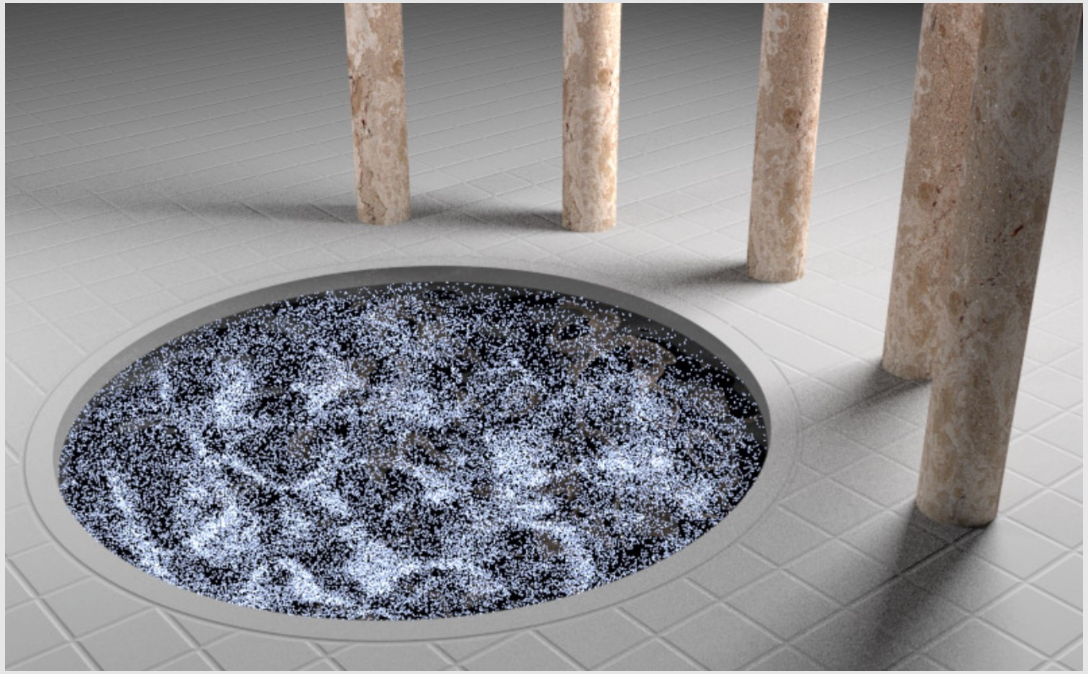
[ original image ]

- Want to take samples proportional to image density $f$
- Occasionally jump to a random point (ergodicity)
- Transition probability is ’relative darkness’
  - $f(x')/f(x_i)$

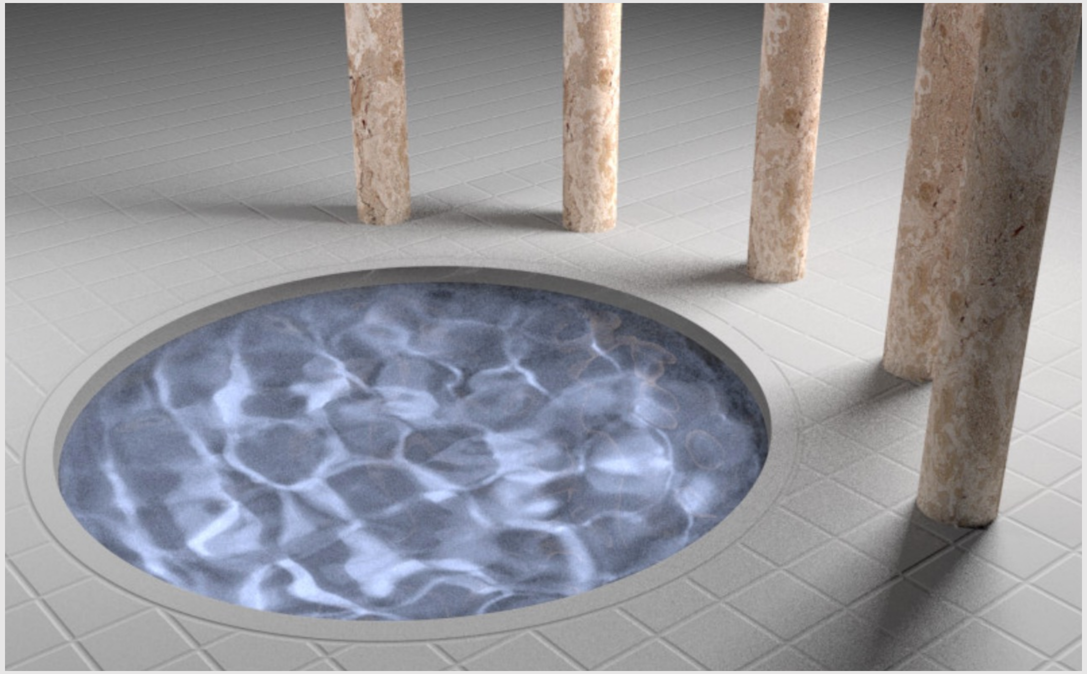
# Metropolis Light Transport

- **Similar idea:** mutate good paths
- Water causes paths to refract a lot
  - Small mutations allows renderer to find contributions faster
- Path Tracing and MLT rendered in the same time



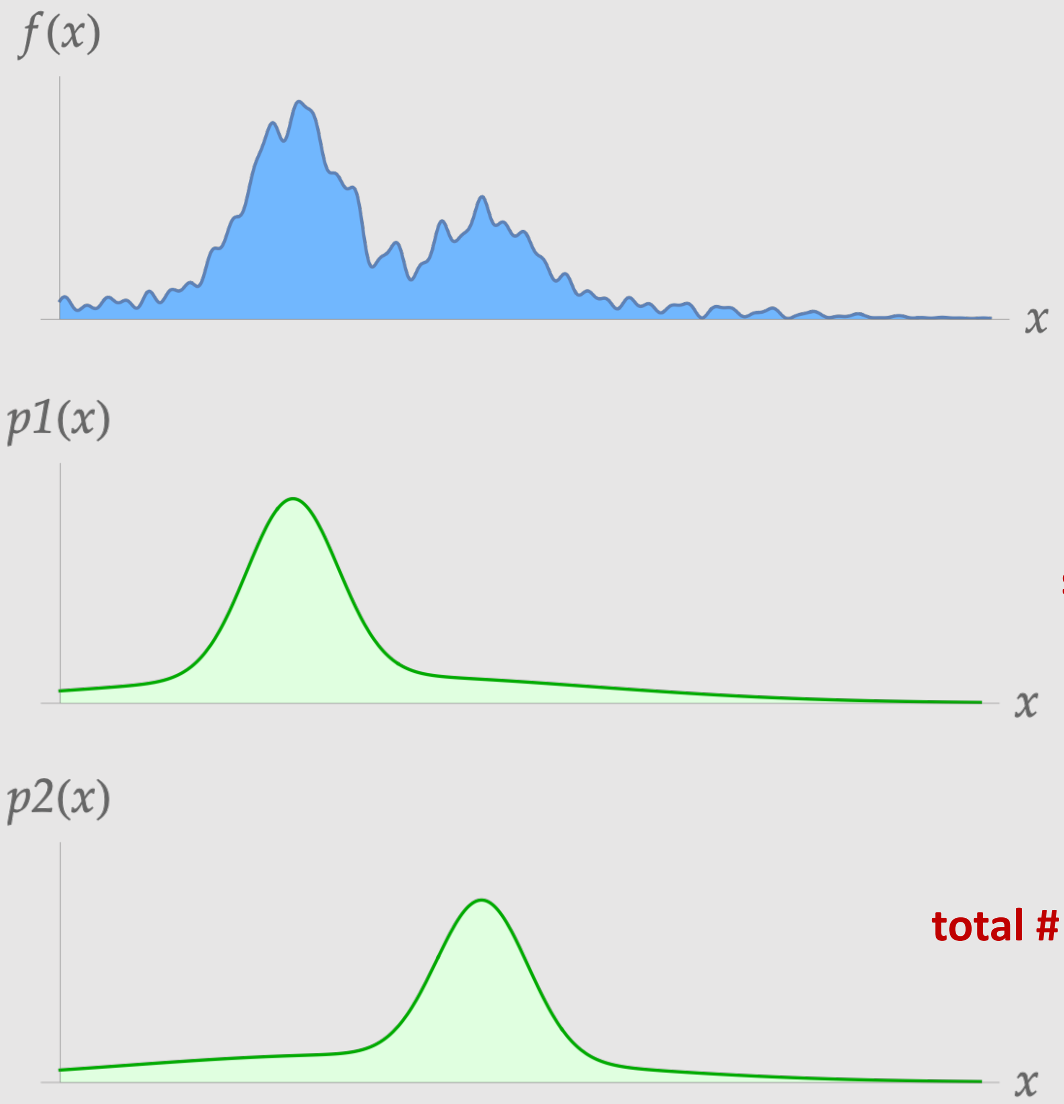
**[ Path Tracing ]**



**[ Metropolis Light Transport ]**

If there are so many good sampling methods,
why not combine them?

# Multiple Importance Sampling




- **Multiple Importance Sampling:** combine strategies to preserve strengths of all of them
  - Think of it as taking multiple rays/samples at each bounce

$$\frac{1}{N}\sum_{i=1}^{n}\sum_{j=1}^{n_i}\frac{f(x_{ij})}{\sum_k c_k p_k(x_{ij})}$$

**sum over strategies**

**sum over samples**

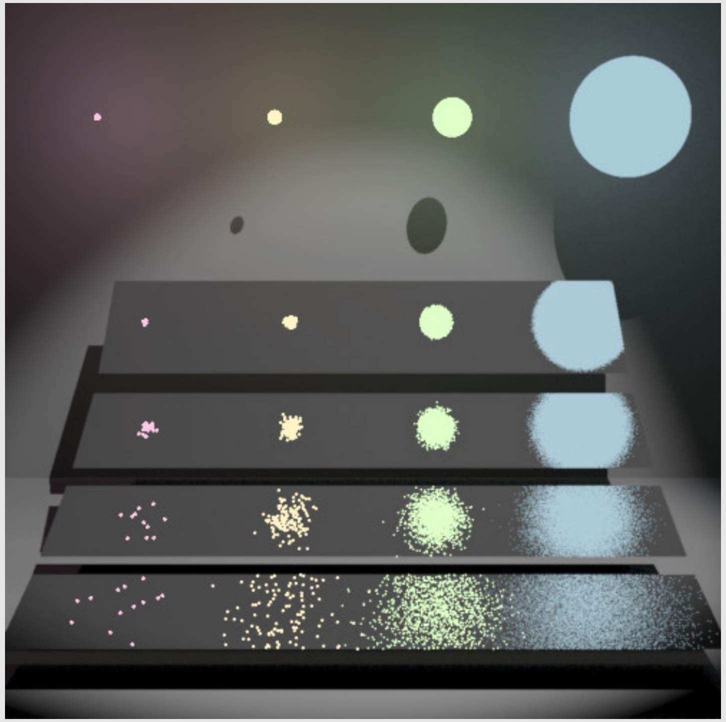
**$j^{th}$ sample taken with $i^{th}$ strategy**

**total # of samples**

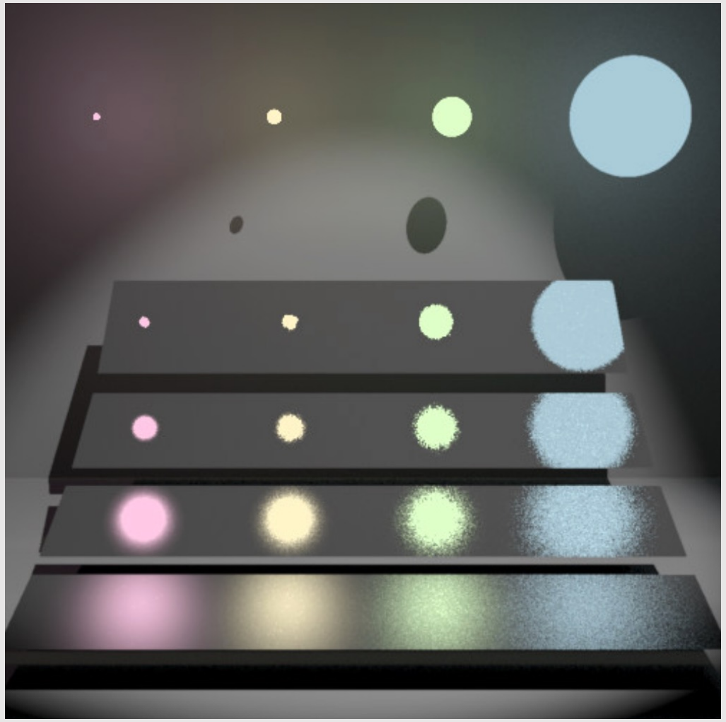
**fraction of samples taken with $k^{th}$ strategy**

**$k^{th}$ strategy PDF**

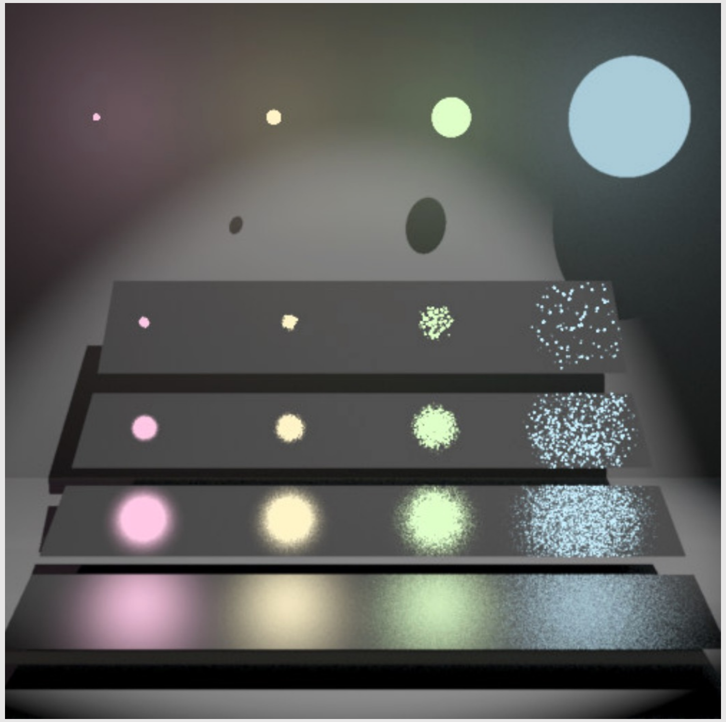
# Multiple Importance Sampling



[ sample materials ]



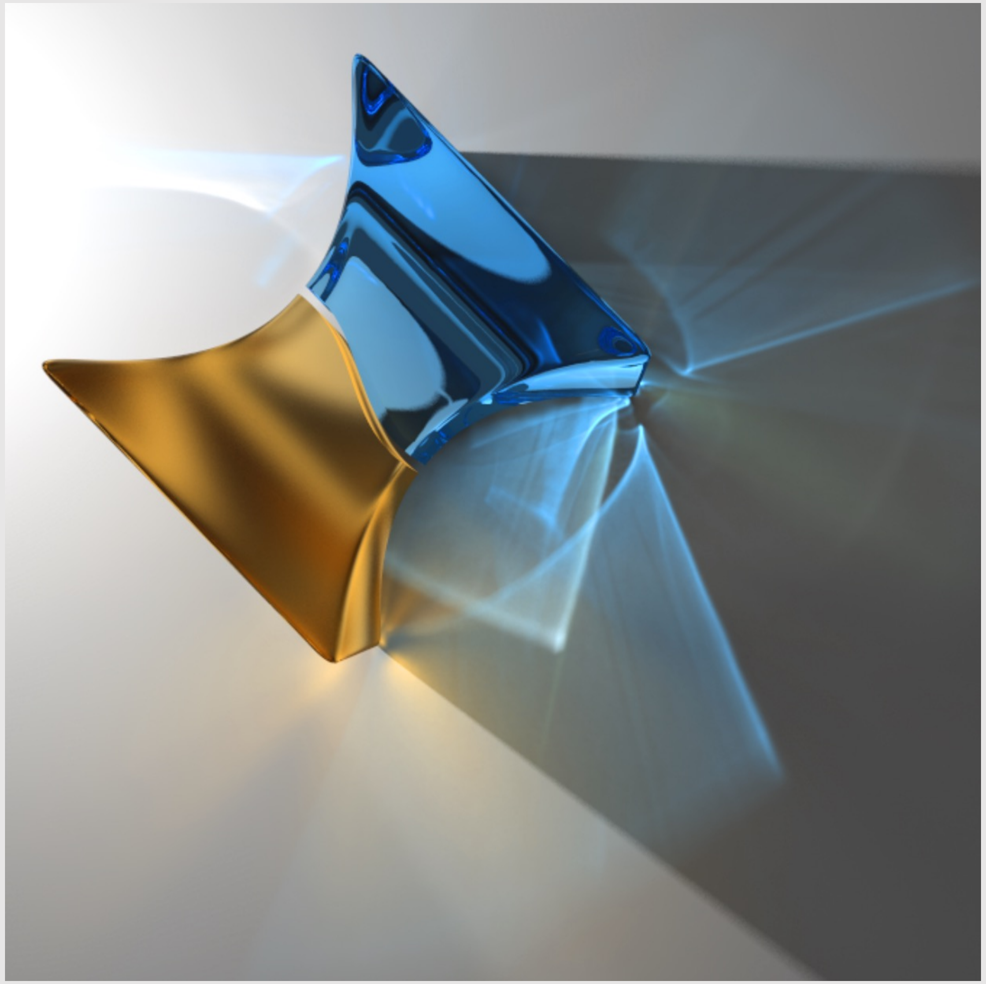
[ sample both ]



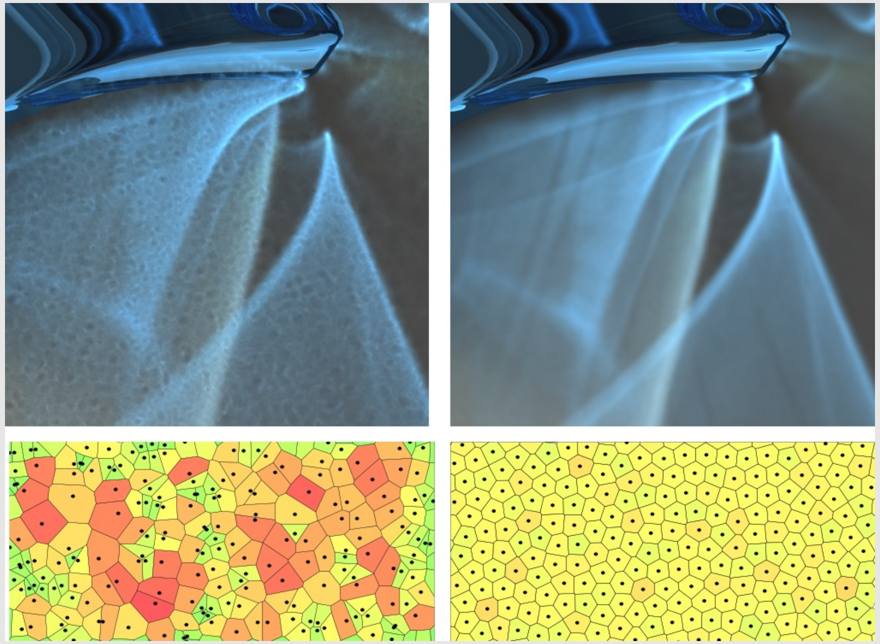
[ sample lights ]

- Normally need to pick next ray bounce as hitting a material or hitting light
  - MIS allows us to take both rays and average them together
  - At each bounce, trace a ray as normal, and another ray to the light

# Photon Mapping

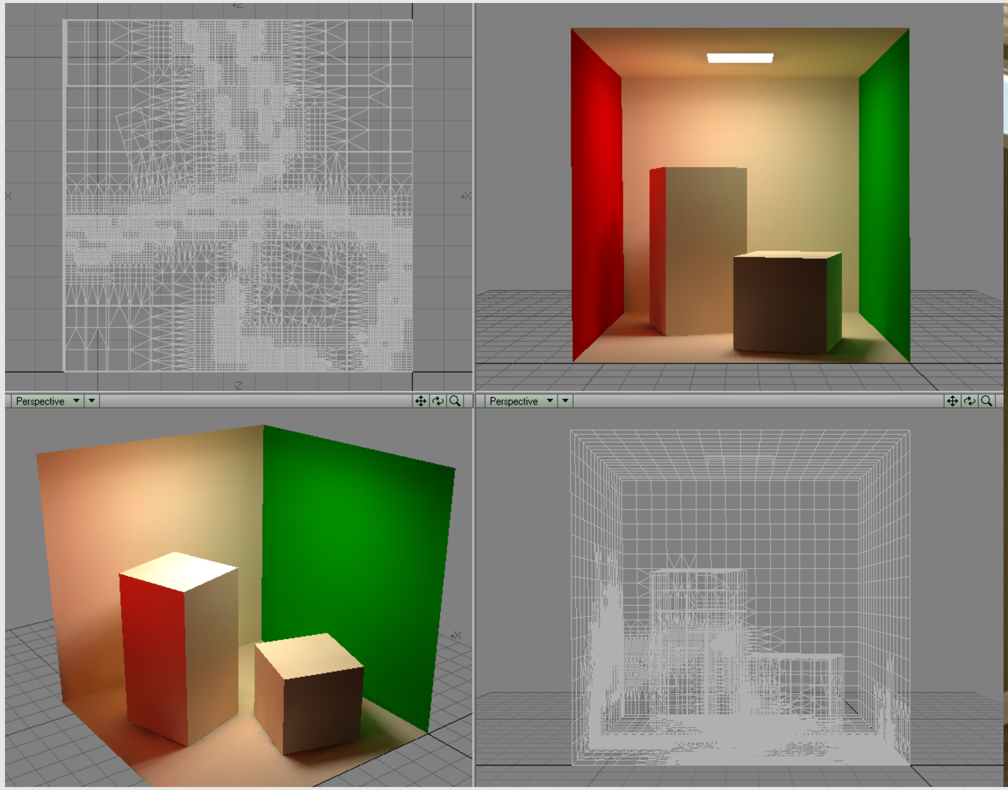


- Trace particles from light, deposit “photons” in KD-tree
  - Useful for, e.g., caustics, fog
- Voronoi diagrams can improve photon distribution
  - **Careful:** poor Voronoi resolution causes aliasing!



# Finite Element Radiosity

- Transport light between patches in scene
- Solve large linear system for equilibrium distribution
    - Good for diffuse lighting; hard to capture other light paths
        - Light paths travel in groups
        - Difficult when light diverges




# Rendering Algorithm Chart

| method | consistent? | unbiased? |
| --- | --- | --- |
| Rasterization | no | no |
| Path Tracing | almost | almost |
| Bidirectional Path Tracing | yes | yes |
| Metropolis Light Transport | yes | yes |
| Photon Mapping | yes | no |
| Finite Element Radiosity | no | no |

# Special Topics in A3:
# Non-Photorealistic Rendering

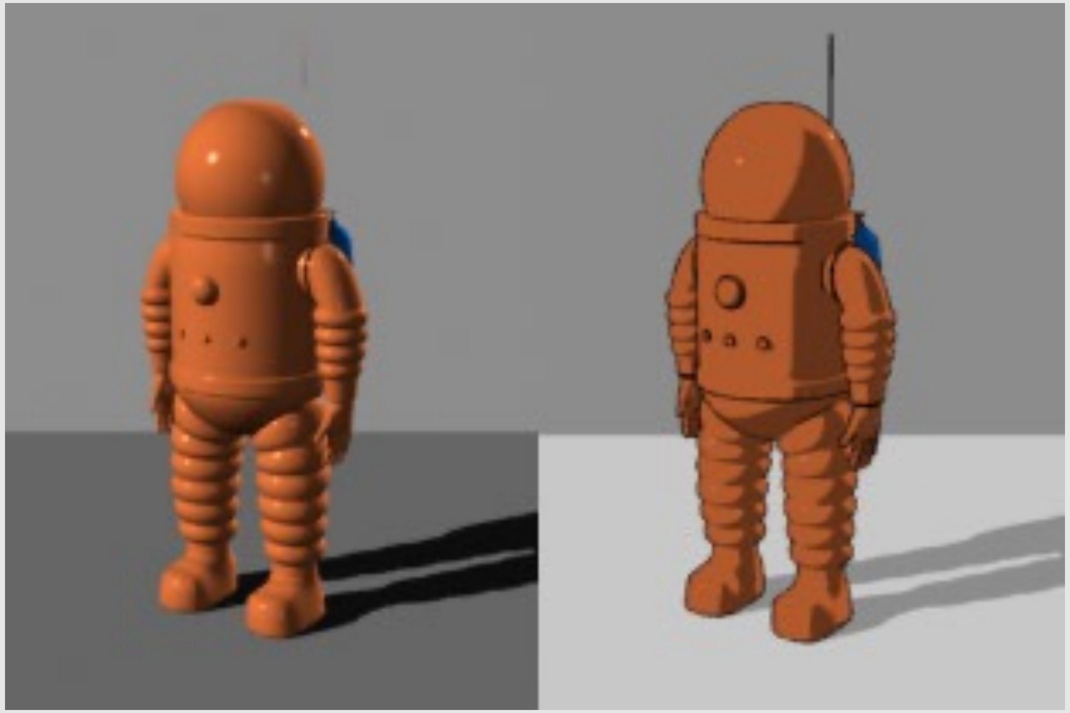
- NPR Introduction
- NPR Features

# Non-Photorealistic Rendering

- Focuses on creativity/expressiveness in rendering
  - Not concerned with making **physically-accurate results**
  - Other names:
    - Toon-shading
    - Cel-shading
    - Stylized Rendering

- Famous for using 3D graphics to create 2D stylings
  - Paint-like
  - Sketch-like
  - Comic-like
  - Cartoon-like

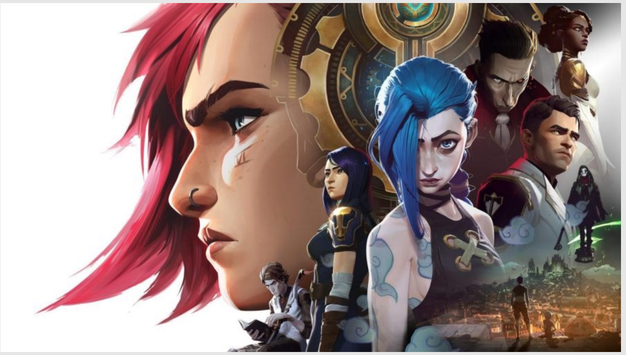
- **Idea:** change the way light interacts with geometry
  - NPR BRDFs



# NPR In Media





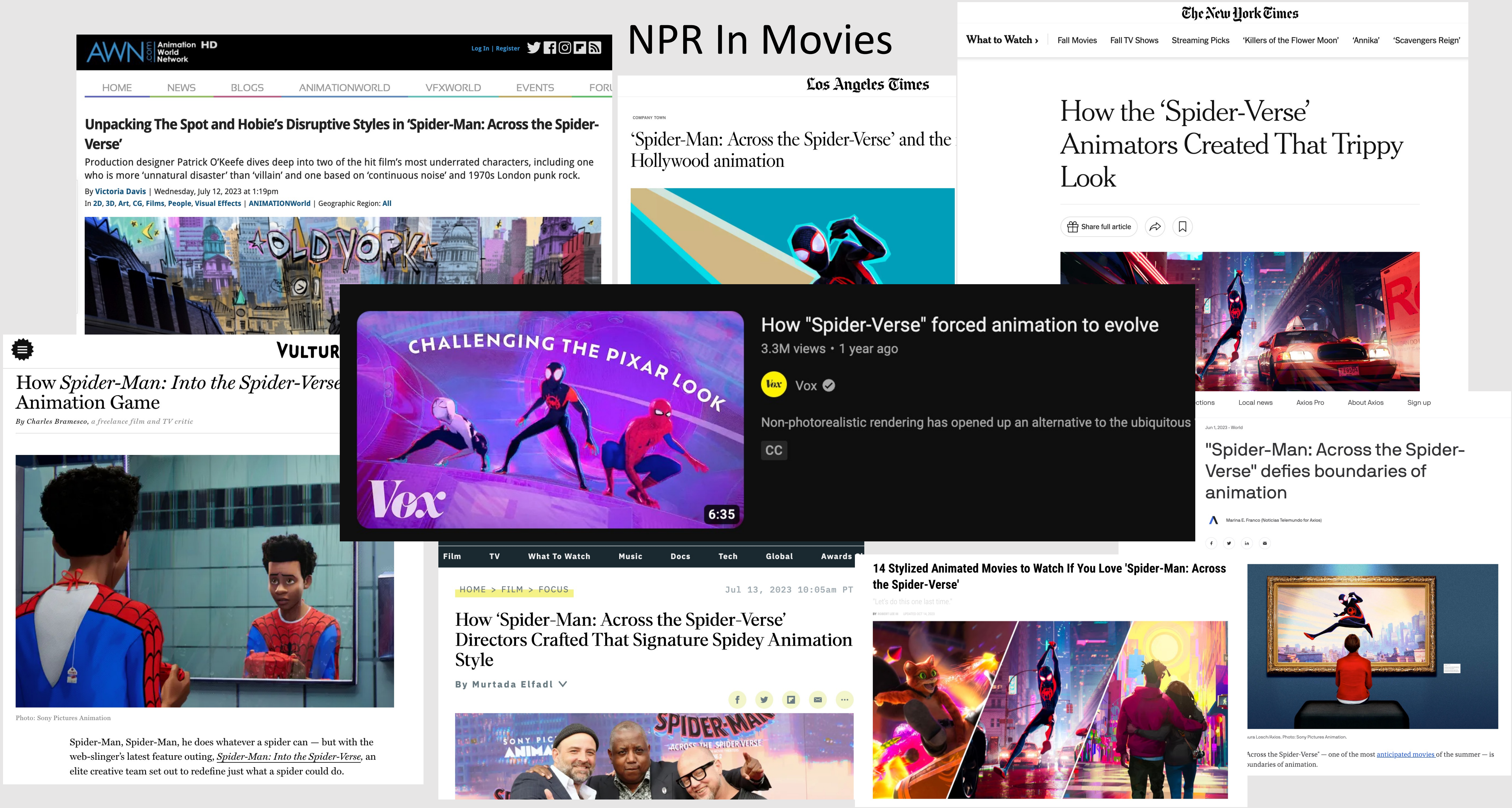






# NPR In Games


Legend of Genshin or something, idk I don’t play video games (2017) Nintendo

# NPR In Movies



AWN.com Animation World Network HD
Log In | Register
HOME NEWS BLOGS ANIMATIONWORLD VFXWORLD EVENTS FORU
Unpacking The Spot and Hobie's Disruptive Styles in 'Spider-Man: Across the Spider-Verse'
Production designer Patrick O'Keefe dives deep into two of the hit film's most underrated characters, including one who is more 'unnatural disaster' than 'villain' and one based on 'continuous noise' and 1970s London punk rock.
By Victoria Davis | Wednesday, July 12, 2023 at 1:19pm
In 2D, 3D, Art, CG, Films, People, Visual Effects | ANIMATIONWorld | Geographic Region: All

Los Angeles Times
COMPANY TOWN
'Spider-Man: Across the Spider-Verse' and the
Hollywood animation

The New York Times
What to Watch › Fall Movies Fall TV Shows Streaming Picks 'Killers of the Flower Moon' 'Annika' 'Scavengers Reign'
How the 'Spider-Verse' Animators Created That Trippy Look
Share full article

How "Spider-Verse" forced animation to evolve
3.3M views • 1 year ago
Vox
Non-photorealistic rendering has opened up an alternative to the ubiquitous
CC
CHALLENGING THE PIXAR LOOK
Vox
6:35

VULTUR
How *Spider-Man: Into the Spider-Verse* Animation Game
By Charles Bramesco, a freelance film and TV critic
Photo: Sony Pictures Animation
Spider-Man, Spider-Man, he does whatever a spider can — but with the web-slinger's latest feature outing, *Spider-Man: Into the Spider-Verse*, an elite creative team set out to redefine just what a spider could do.

Film TV What To Watch Music Docs Tech Global Awards
HOME > FILM > FOCUS Jul 13, 2023 10:05am PT
How 'Spider-Man: Across the Spider-Verse' Directors Crafted That Signature Spidey Animation Style
By Murtada Elfadl

14 Stylized Animated Movies to Watch If You Love 'Spider-Man: Across the Spider-Verse'

ctions Local news Axios Pro About Axios Sign up
Jun 1, 2023 - World
"Spider-Man: Across the Spider-Verse" defies boundaries of animation
Marina E. Franco (Noticias Telemundo for Axios)
Across the Spider-Verse" — one of the most anticipated movies of the summer — is
oundaries of animation.

# NPR Styles: Low Poly

Lonely Mountains Downhill (2019) Megagon Industries

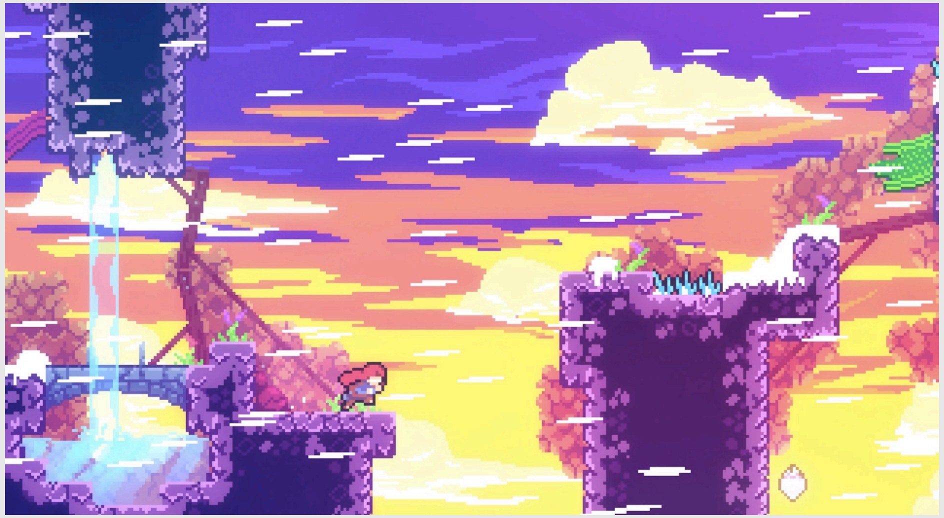
SuperHot (2016) SuperHot Team

What The Golf (2019) Triband

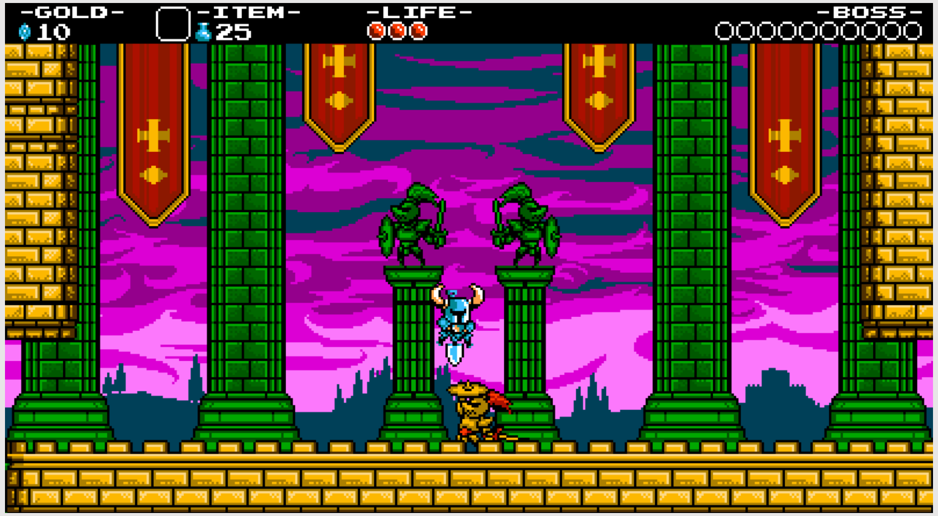
Minecraft (2011) Mojang

# NPR Styles: Pixel Art

Enter the Gungeon (2016) Devolver Digital



Celeste (2018) Maddy Makes Games




Shovel Knight (2014) Yacht Club Games

Stardew Valley (2016) Eric Barone

# NPR Styles: 2D Animation

Snow White (1937) Walt Disney Animation Studio

Klaus (2019) SPA Studios

Moana (2016) Walt Disney Animation Studio

The Bad Guys (2022) Dreamworks

# NPR Styles: Ghibli Animation

My Neighbor Totoro (1988) Studio Ghibli

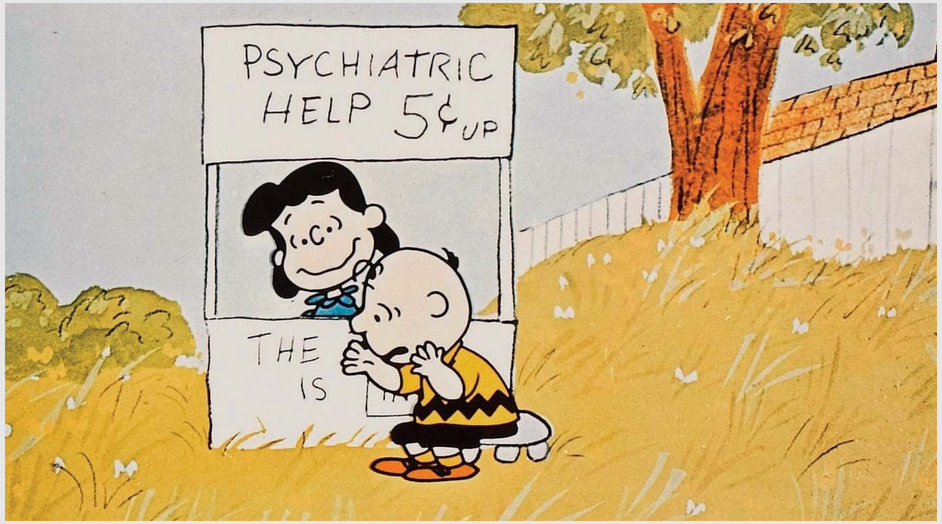
Ponyo (2009) Studio Ghibli

Spirited Away (2001) Studio Ghibli

Earwig And The Witch (2020) Studio Ghibli

# NPR Styles: Comic Book

Spiderman vs Superman



Peanuts

One Piece Netflix Adaptation


Lore Olympus

# Halftone Printing

- Also referred to as **stippling**
- Printing used to require carving out blocks and dipping them in ink to press onto sheets of paper
  - Printing many small dots was an easier way to print images
    - Can create gradients with varying dot sizes
- Subtractive color scheme: print colors on top of each other on white paper to produce darker colors
  - Example: CMYK
- Printing many small CMKY dots produced color prints
  - The alignment offset caused by printing error became known as the 'comic book' style


- ~~NPR Introduction~~

- NPR Features

# Non-Photorealistic Rendering

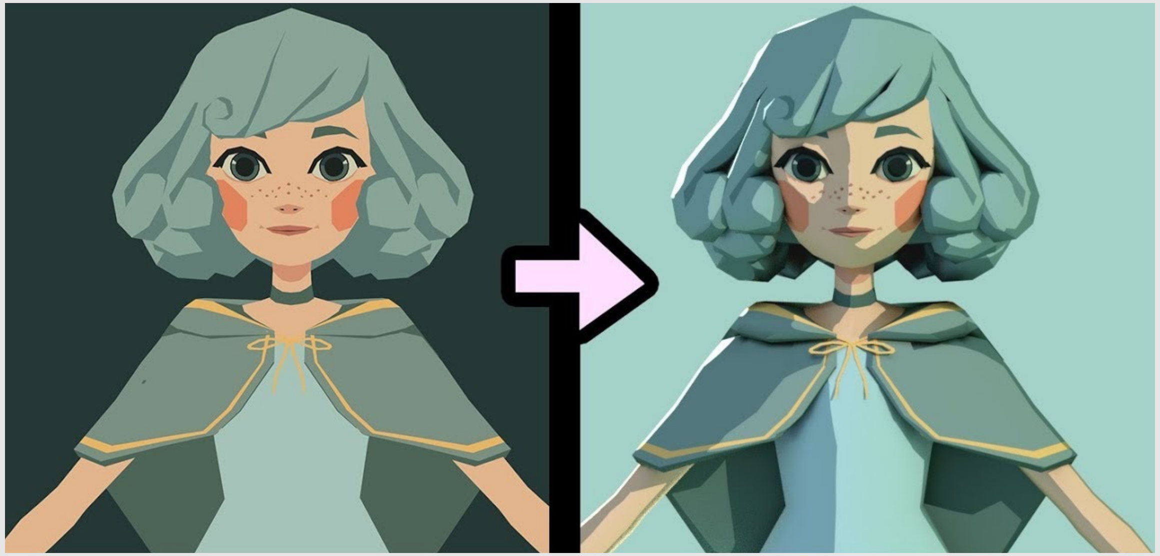
- NPR rendering is heavily based off of 2D art
  - What are the components of 2D art?
    - Solid color
    - Hard shadows
    - Outlines

- **Goal:** achieve these components in our renderer
  - We want our 3D graphics to look like 2D graphics

Wish (2023) Walt Disney Pictures

# Solid Colors

- The foundation of NPR is **solid colors**
  - Rather than interpolating colors at vertices, set solid colors for entire regions
- Emphasize bold, contrasting colors
  - Want to be able to tell apart different elements from their colors
  - Some shadows are baked in
    - Ex: hair
- Can add hard shadows for volumetric effect

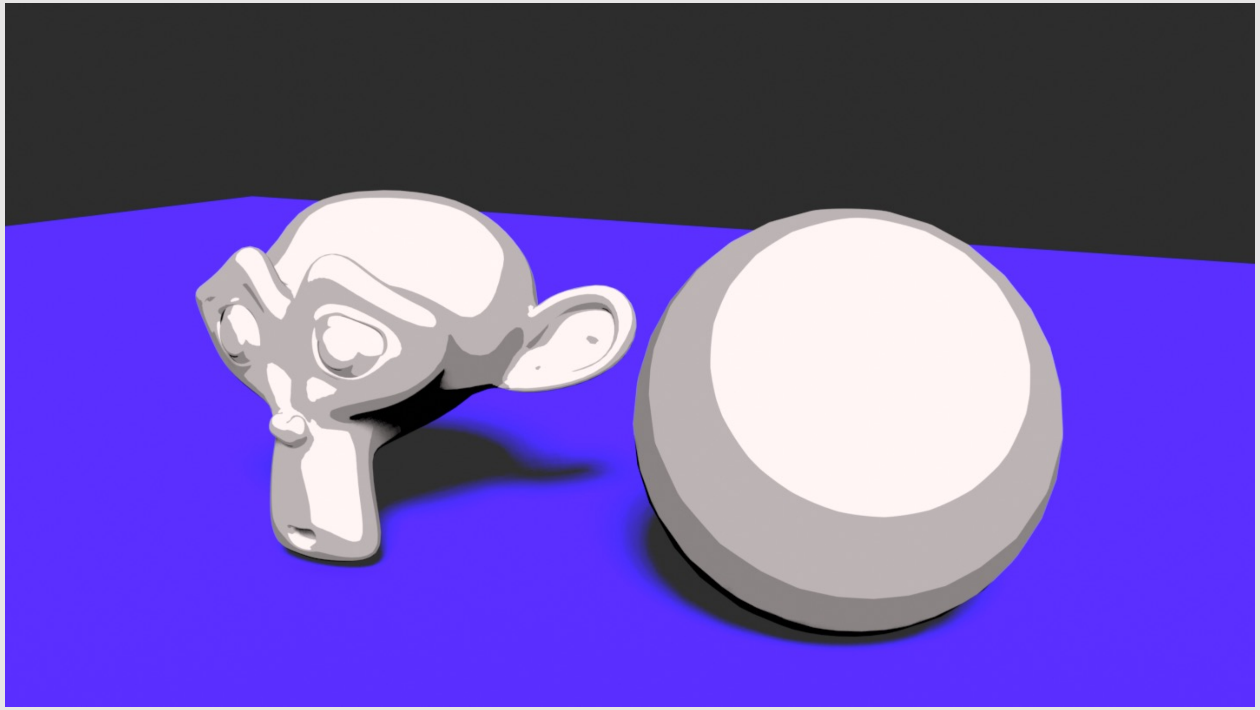


The Legend of Zelda: The Wind Waker (2002) Nintendo

# Hard Shadows

- Shading was an expensive task in 2D animation
  - Saved time by drawing hard shadow layers instead of shadow gradients
  - Replicated in NPR for the '2D look'

- **Algorithm:**
  - Extract world-space normal
  - Take dot-product with camera look-at direction
  - Threshold values, mapping to 0 or 1
    - Creates a binary mask
    - This is now your hard shadow
    - The larger the threshold, the larger the shadows
  - Repeat, changing threshold and shadow opacity for multiple hard shadows



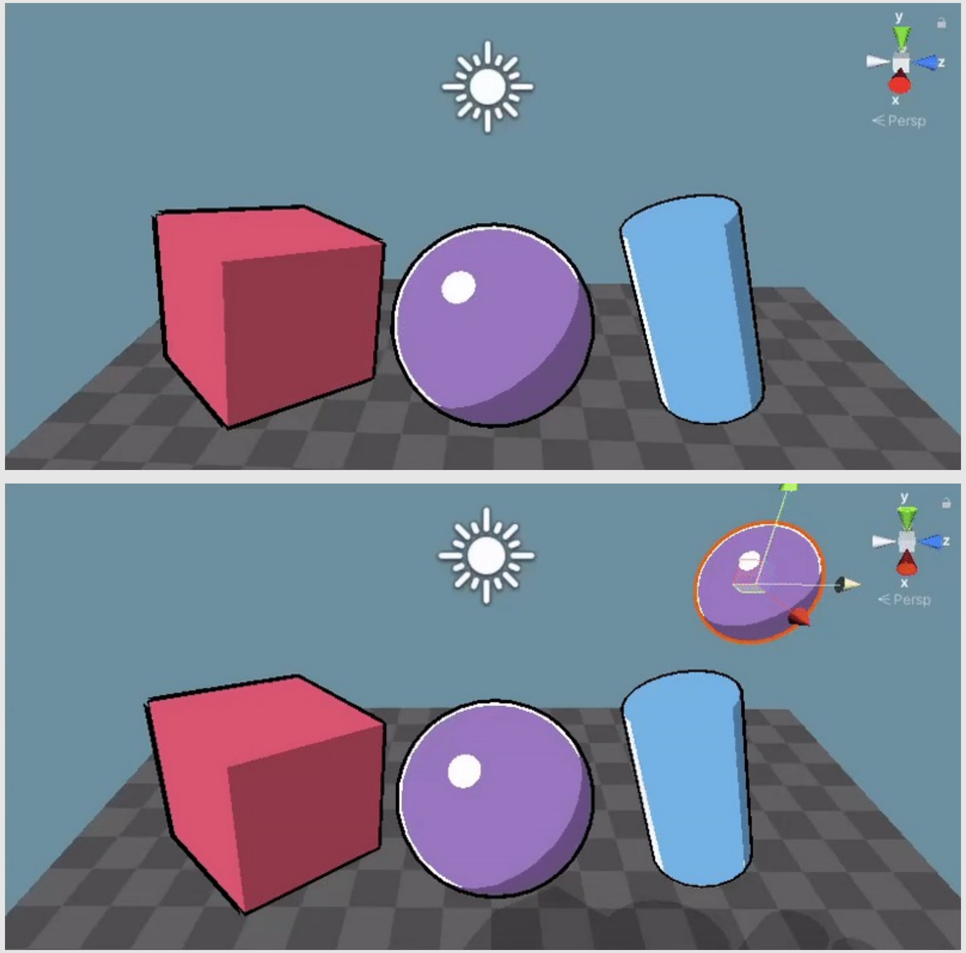
Making A NPR Shader In Blender (2021) Maxime Garcia

# Hard Shadows

- **Problem:** Hard shadows change as camera moves around
  - What if we instead want to add hard shadows from a light source?

- **Algorithm:**
  - Extract world-space normal
  - **Take dot-product with ~~camera look-at direction~~ world-space light vector**
  - Threshold values, mapping to 0 or 1
    - Creates a binary mask
    - This is now your hard shadow
    - The larger the threshold, the larger the shadows
  - Repeat, changing threshold and shadow opacity for multiple hard shadows

- Can also use for specular highlights + casting shadows!



Cel Shader (2021) Lewis Gadsby

# N-dot-V Outline

- **Problem:** Want to identify where the outlines are in our model
- **Idea:** outlines are areas in the mesh that sit in between areas that face towards us and areas that face away from us
- **Algorithm:**
  - Extract world-space normal
  - Take dot-product with camera look-at direction
  - Look at areas where the dot product is close to 0
    - These are the edges to your model
    - Shade them darker


Ryner (2015) Lucas Gogol

# Inverted Hull Outline

- **Problem:** Want to identify where the outlines are in our model
- **Idea:** outlines are areas in the mesh that sit in between areas that face towards us and areas that face away from us
- **Algorithm:**
  - Create geometry
  - Duplicate geometry
    - Create outline
    - Flip normals/Invert hull
    - Enable backface culling
- Outlines are now a separate geometry that sit on top of the original geometry

Cel-Shading (2020) Andrey Torchinsky

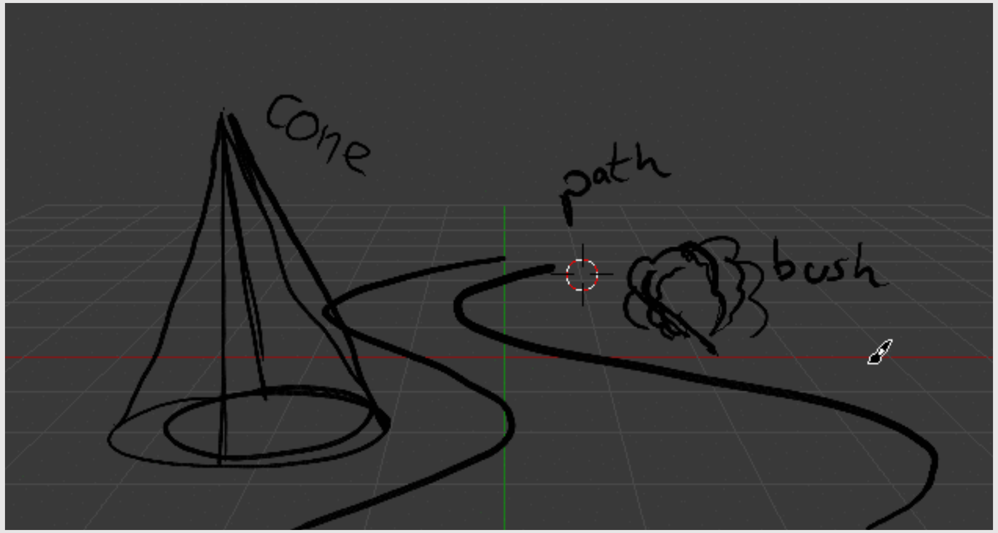
# Painting

The Mitchells vs. the Machines (2021) Sony Pictures Animation

- **Problem:** want to get the hand-drawn look in animation
- Sometimes it is more work to automate a process than just doing the process
  - More work to write a shader than just draw it
- **Idea:** Draw 2D brushes on top of 3D frames
  - **Issue:** Where do those brush strokes sit in the scene?

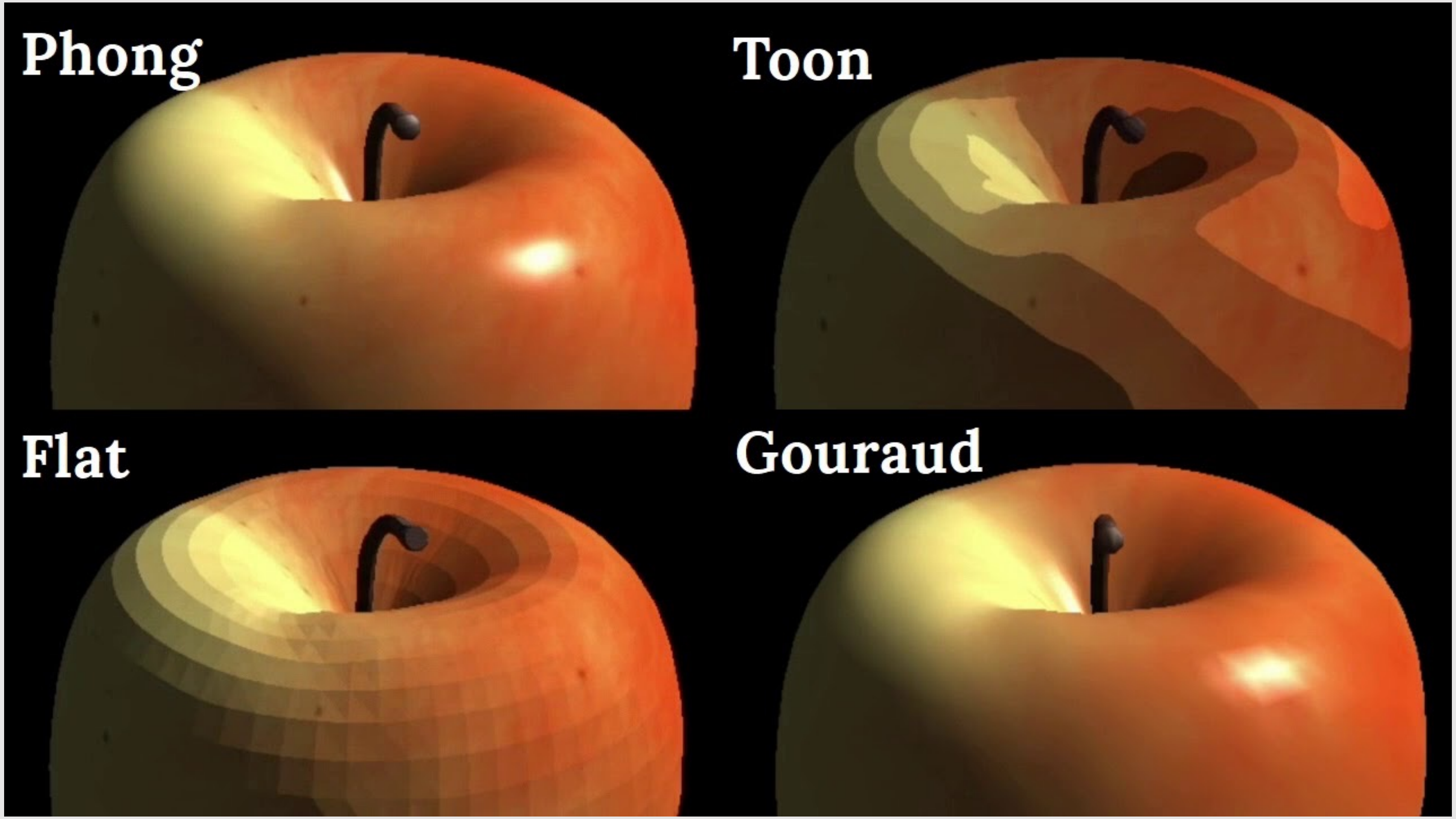
# Grease Pencil

- Released in Blender in 2008
- Draw strokes on screen
  - Saved as a collection of points with vector paths passing through
  - Can be warped or manipulated
- Strokes saved to the XZ plane
  - Planes can be transformed in 3D
  - Changing the view of the camera changed where new strokes would be generated
- **Workflow:**
  - Draw strokes
  - Move strokes
  - Repeat




Blender Grease Pencil (2019) Nicola Sap, gskinner, Dédouze

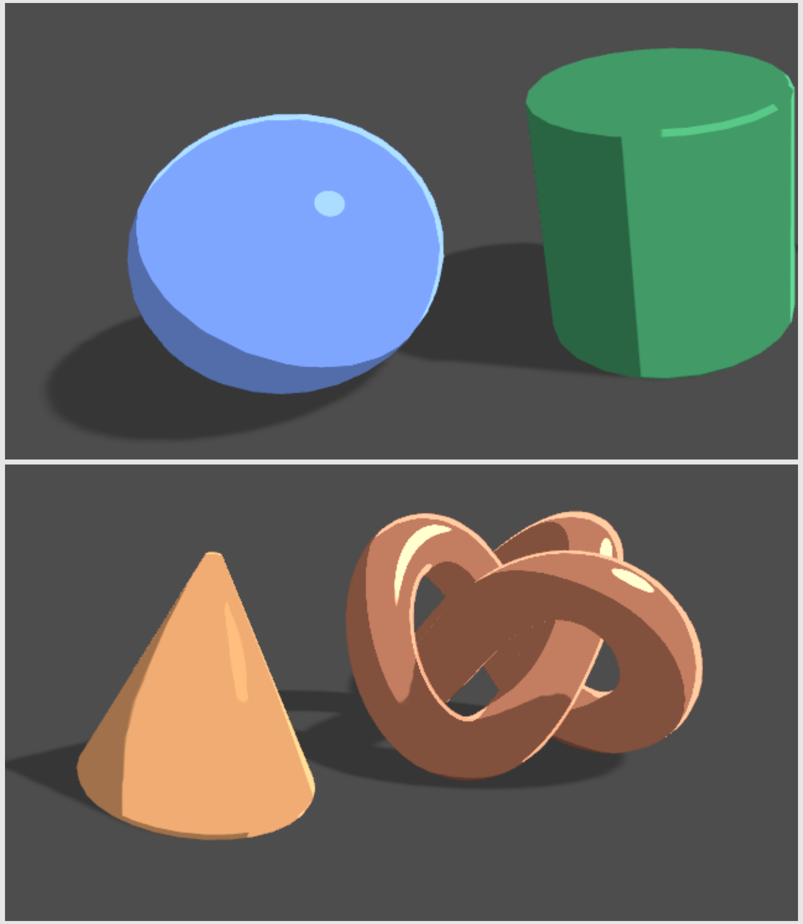
# Shading Methods




Phong, Gouraud, Flat, and Toon shading in OpenGL (2017) Chih-Chun Hsu

# Shading Methods

- **Flat Shading**
  - Shade entire primitive with the same surface normal
- **Phong Shading**
  - Interpolate normals at vertices using barycentric coordinates, then shade with interpolated normal
- **Gouraud Shading**
  - Shade at each vertex with its vertex normal, then interpolate vertex colors using barycentric coordinates
- **Toon Shading**
  - Flat, Phong, or Gouraud, with thresholding on lighting/color



What rules of rendering still apply in non-photorealistic rendering

# The Rendering Equation

$$L_o(\mathbf{p}, \omega_o) \;=\; L_e(\mathbf{p}, \omega_o) + \int_{\mathcal{H}^2} f_r(\mathbf{p}, \omega_i \to \omega_o) L_i(\mathbf{p}, \omega_i) \cos\theta \, d\omega_i$$

$L_o(\mathbf{p}, \omega_o)$ outgoing radiance at point $\mathbf{p}$ in outgoing direction $\omega_o$

$L_e(\mathbf{p}, \omega_o)$ emitted radiance at point $\mathbf{p}$ in outgoing direction $\omega_o$

$f_r(\mathbf{p}, \omega_i \to \omega_o)$ scattering function at point $\mathbf{p}$ from incoming direction $\omega_i$ to outgoing direction $\omega_o$

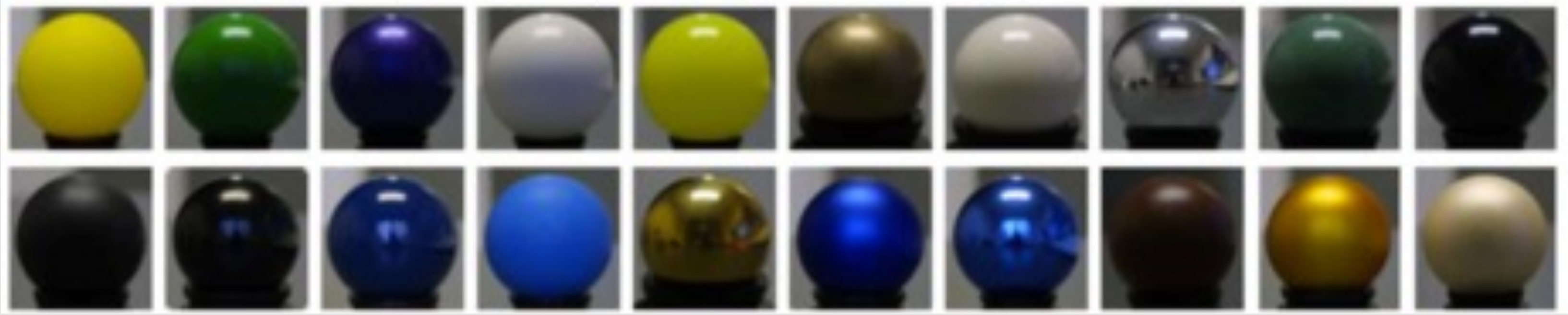
$L_i(\mathbf{p}, \omega_i)$ incoming radiance to point $\mathbf{p}$ from direction $\omega_i$

**still the same rendering equation!**

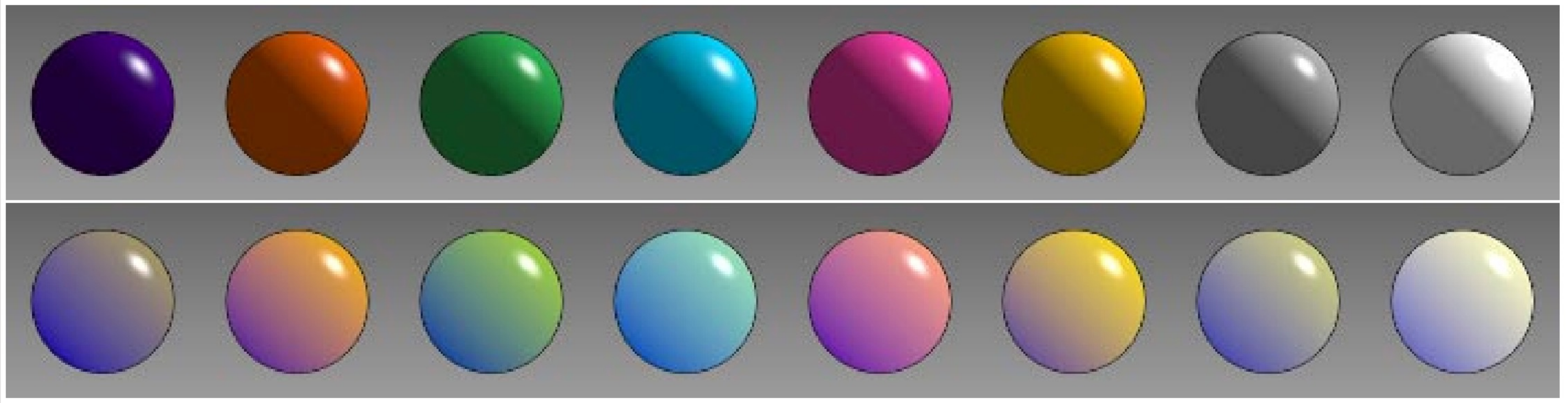
**Rendering equation describes how light interacts in the volume between objects, not the objects themselves!**

# Materials

**[ Photorealistic Materials ]**



**[ Non-Photorealistic Materials ]**



A Non-Photorealistic Lighting Model For Automatic Technical Illustration (1998) A. Gooch, B. Gooch, P. Shirley, E. Cohen

# Number Of Ray Samples

- **Number of Rays**
  - How many rays we trace into the scene
    - Measured as samples (rays) per pixel [spp]

- Increasing the number of rays increases the quality of the image
  - Anti-aliasing
  - Reduces black spots from terminating emission occlusion

[ 1 spp ]

[ 16 spp ]

# Bias & Consistency

- An estimator is **consistent** if it converges to the correct answer:

$$\lim_{n\to\infty} P(|I - \hat{I}_n| > 0) = 0$$

**near infinite # of samples**

- An estimator is **unbiased** if it is correct on average:

$$E[I - \hat{I}_n] = 0$$

**even if just 1 sample**

- **consistent != unbiased**


[ biased ] [ unbiased ]

# Bias & Consistency

- An estimator is **consistent** if it converges to the correct answer:

$$\lim_{n \to \infty} P(|I - \hat{I}_n| > 0) = 0$$

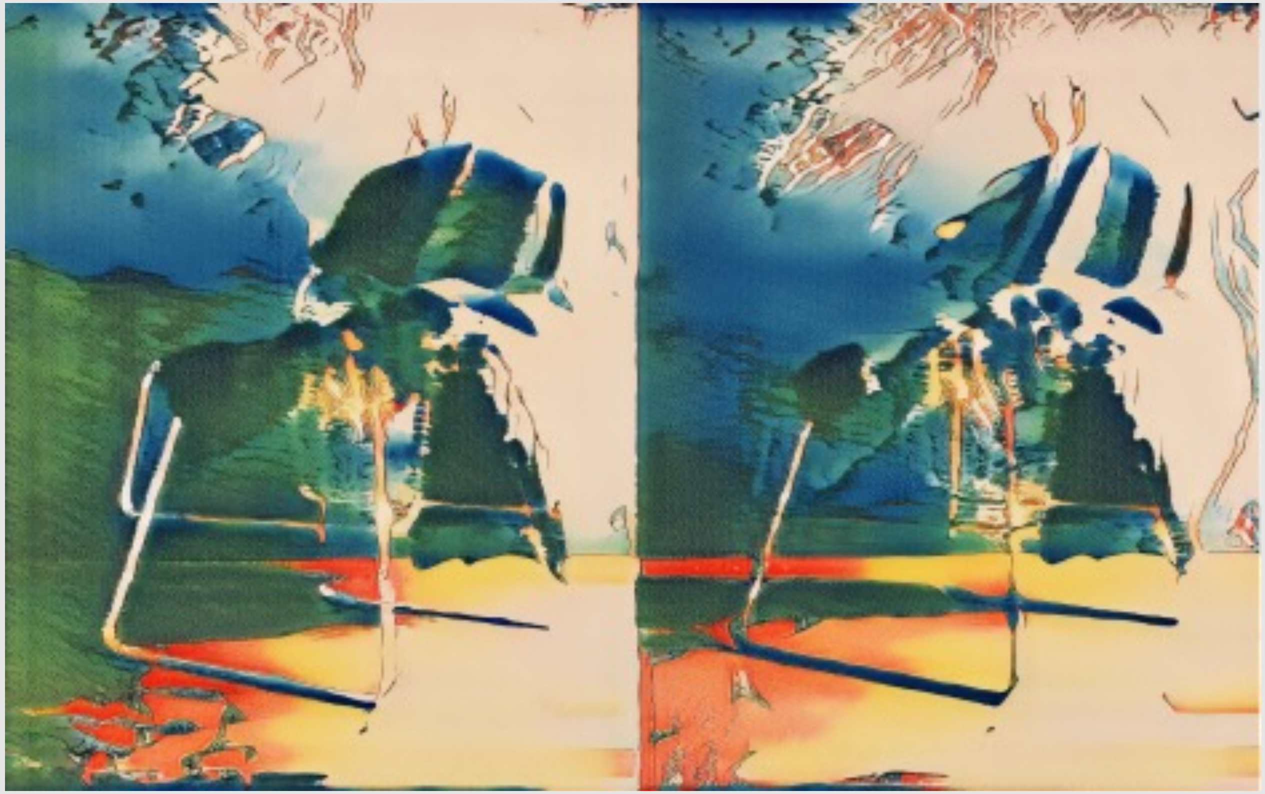
**replace 'correct answer'**

- An estimator is **unbiased** if it is correct on average:

$$E[I - \hat{I}_n] = 0$$

**replace 'correct answer'**

- **consistent != unbiased**



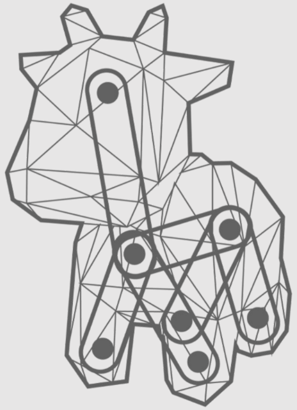
**[ biased ]** **[ unbiased ]**

# Course Roadmap


[ A1: Rasterization ]


[ A2: MeshEdit ]


[ A3: PathTracer ]



**Next Time**

[ A4: Animation ]