Interactive Real-Time BRDF Editing under Environment Lighting

Ailie Fraser
Pixel Café
March 6th, 2015

Question

How do graphic artists and designers create a scene and edit its appearance?

Question

How do graphic artists and designers create a scene and edit its appearance?

- Draw objects
- Define material properties and reflectance of objects (BRDF)
- Lighting (point lights or environment maps)

Motivation

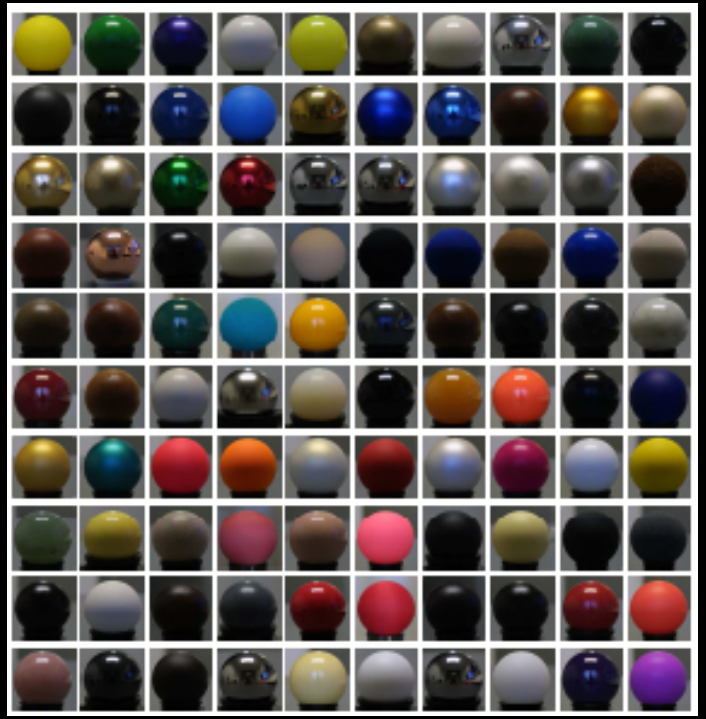
How do graphic artists and designers create a scene and edit its appearance?

- Draw objects
- Define material properties and reflectance of objects (BRDF)
- Lighting (point lights or environment maps)

Motivation

Material Reflectance

The BRDF (Bidirectional Reflectance Distribution Function) of a surface determines how that surface reflects light.



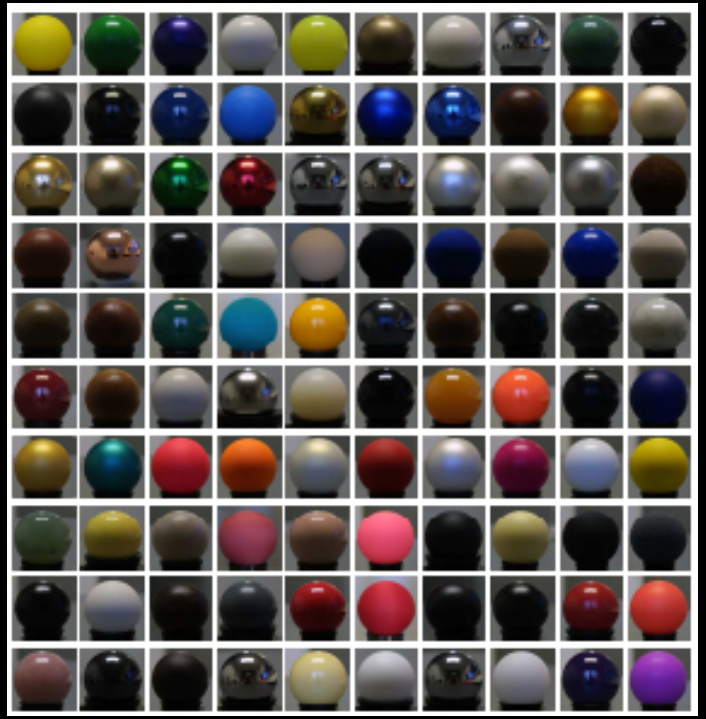
MERL Database, http://www.merl.com/brdf/

Motivation

Material Reflectance

The BRDF (Bidirectional Reflectance Distribution Function) of a surface determines how that surface reflects light.

How do we edit an object's BRDF?

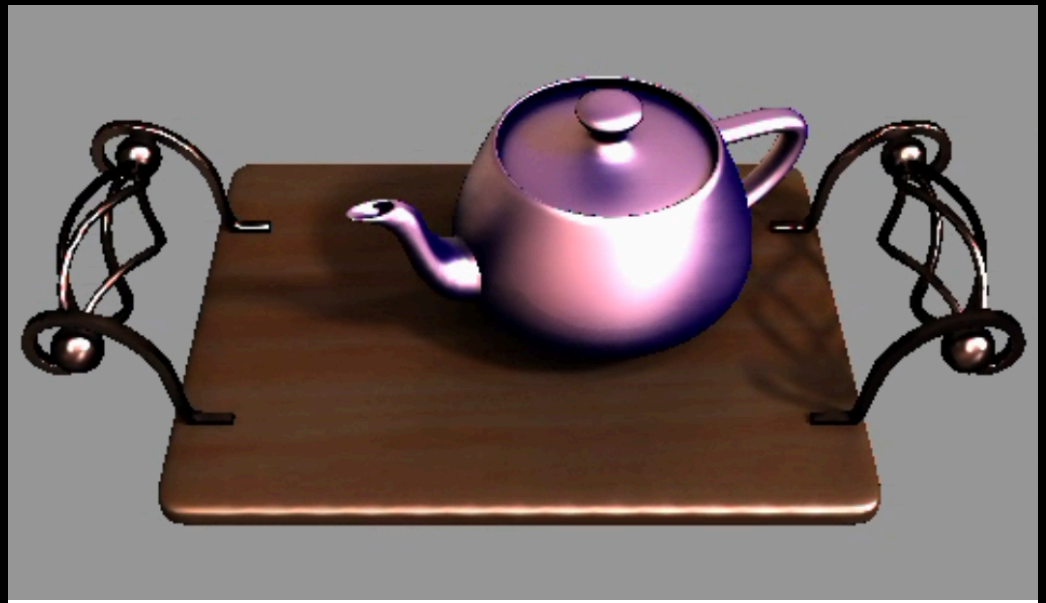


MERL Database, http://www.merl.com/brdf/

Motivation

How to make the teapot shinier?

Or edit just the teapot's handle?



$$\rho_{AS} = \frac{\sqrt{(n_u+1)(n_v+1)}}{8\pi} \frac{(\cos\theta_h)^{n_u \cos^2\phi_h + n_v \sin^2\phi_h}}{\theta_d \max(\cos\theta_l, \cos\theta_v)} F(\theta_d)$$

BRDF:
(Ashikhmin-Shirley)

Ben-Artzi et al., 2006

Motivation

Goal:

- Real-time interactive BRDF editing system

Motivation

Goal:

- Real-time interactive BRDF editing system
- Paint edits directly on material

Motivation

Goal:

- Real-time interactive BRDF editing system
- Paint edits directly on material
- Show objects under environment lighting

Motivation

Goal:

- Real-time interactive BRDF editing system
- Paint edits directly on material
- Show objects under environment lighting

Motivation

Goal:

- Real-time interactive BRDF editing system
- Paint edits directly on material
- Show objects under environment lighting
- Edit intuitive material properties

Motivation

Goal:

- Real-time interactive BRDF editing system
- Paint edits directly on material
- Show objects under environment lighting
- Edit intuitive material properties
- Give artistic freedom while maintaining photorealism

Overview

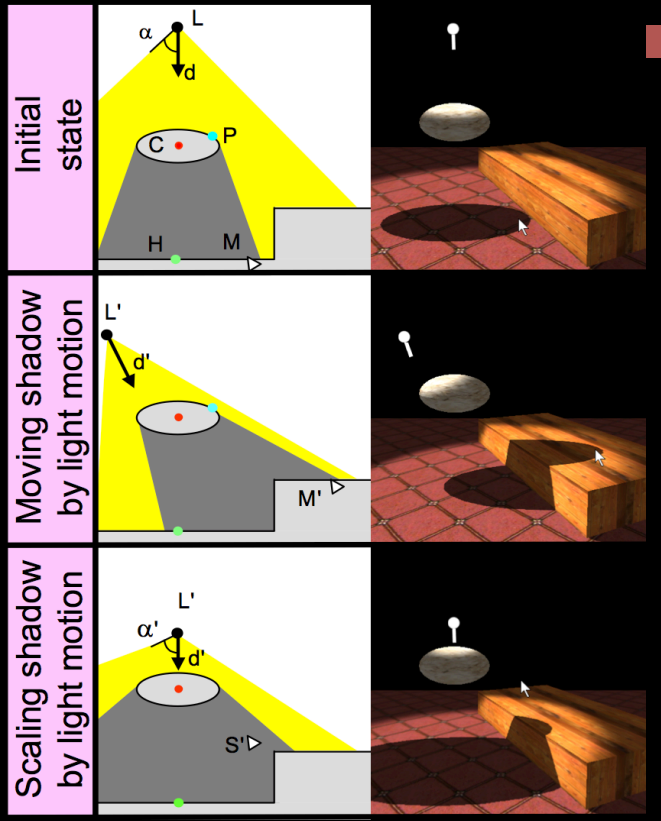
- Background
- Environment Mapping
- Implementation
- Next Steps

Background: Interactively Editing Lighting & Effects

Background: Interactively Editing Lighting & Effects

Interactive Shadow Editing

- Move, rescale or rotate shadows
- System inversely computes new light or object positions
- User can define constraints
- Uses shadow mapping

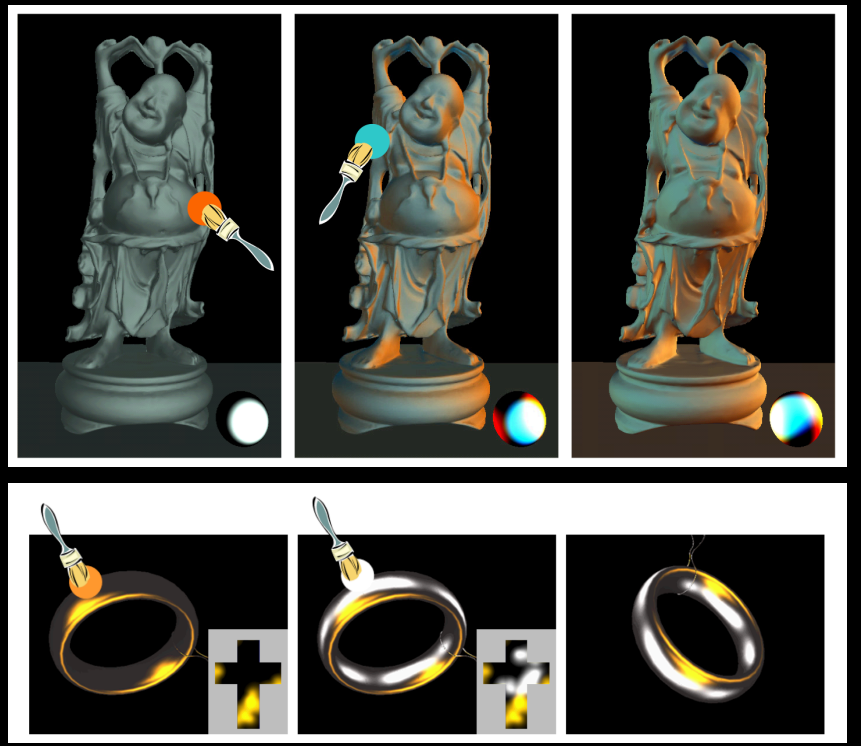


Pellacini et al., “A User Interface for Interactive Cinematic Shadow Design”, 2002

Background: Interactively Editing Lighting & Effects

Illumination Brush

- Draw colour on object
- Inversely calculate lighting
- Separate brushes for diffuse and specular
- Lights are synthetic and low-frequency

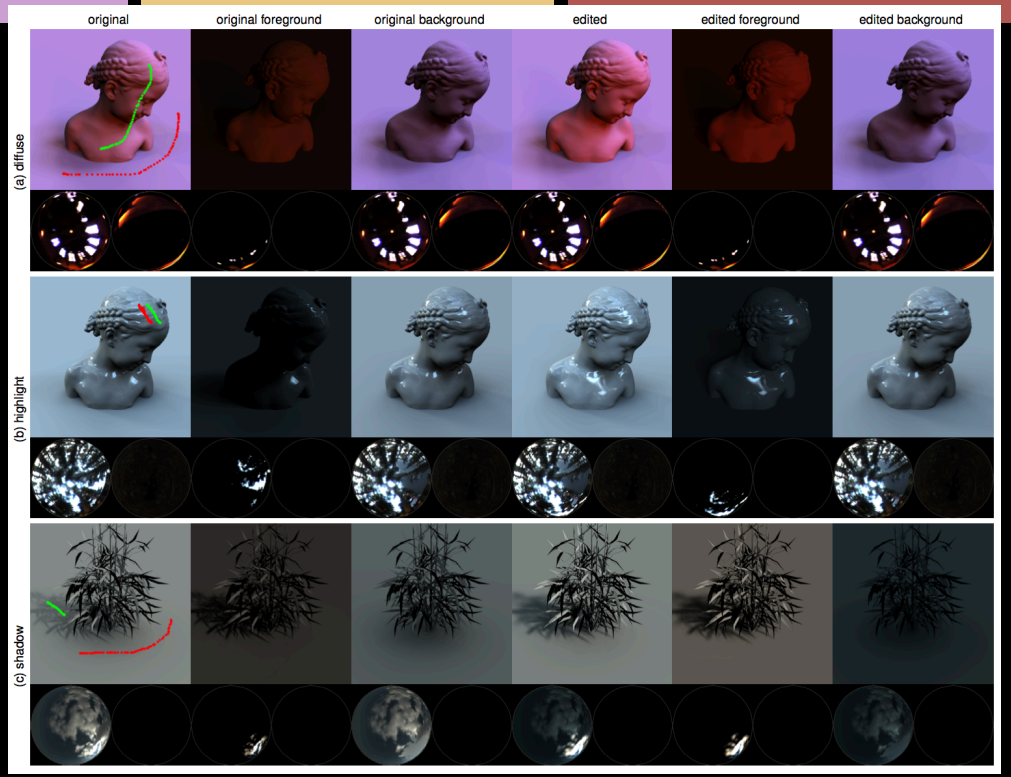


Okabe et al., “Illumination Brush: Interactive Design of Image-Based Lighting”, 2006

Background: Interactively Editing Lighting & Effects

envyLight

- Editing real-world illumination
- Change contrast, position or blur of lighting effects
- Splits environment map into foreground and background

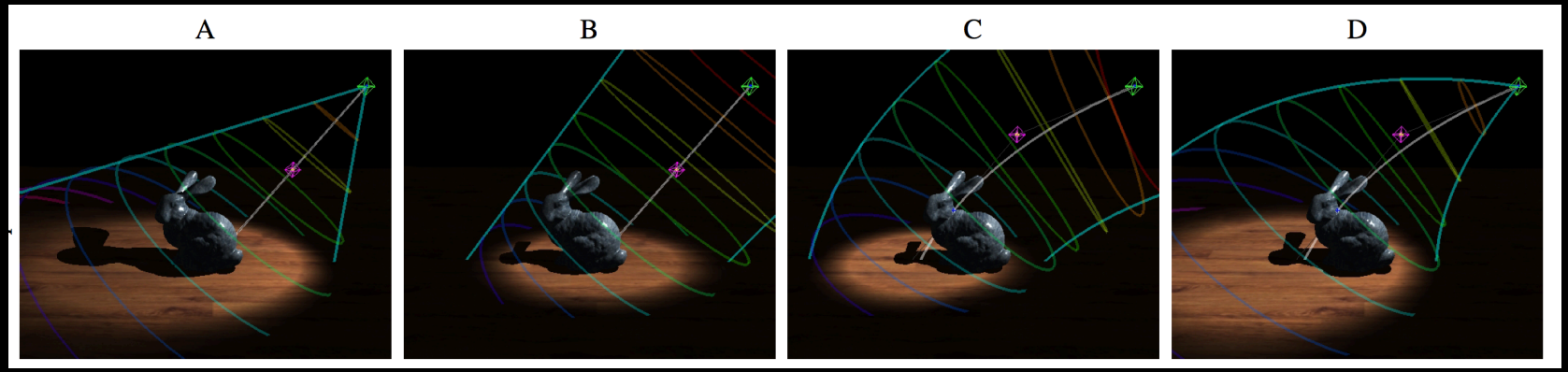


Pellacini, “envyLight: An Interface for Editing Natural Illumination”, 2010

Background: Interactively Editing Lighting & Effects

Bending/Redirecting Light Rays

- BendyLights: lets light rays be nonlinear
- User can drag the light “tube” to edit



Kerr et al., “BendyLights: Artistic Control of Direct Illumination by Curving Light Rays”, 2010

Background: Interactively Editing Lighting & Effects

Bending/Redirecting Light Rays

- Edit complex lighting effects interactively

Schmidt et al., “Path-Space Manipulation of Physically-Based Light Transport”, 2013

Background: Interactively Editing Lighting & Effects

Bending/Redirecting Light Rays

- Interactive reflection editing
- Redirect the mirror reflection direction in real-time

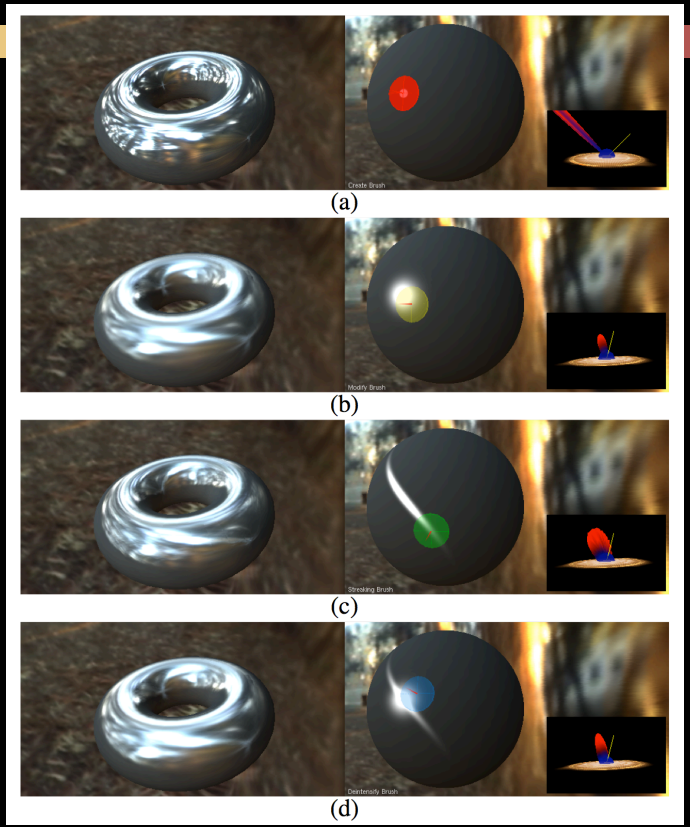
Ritschel et al., “Interactive Reflection Editing”, 2009

Background: Interactive BRDF Editing

Background: Interactive BRDF Editing

BRDF-Shop

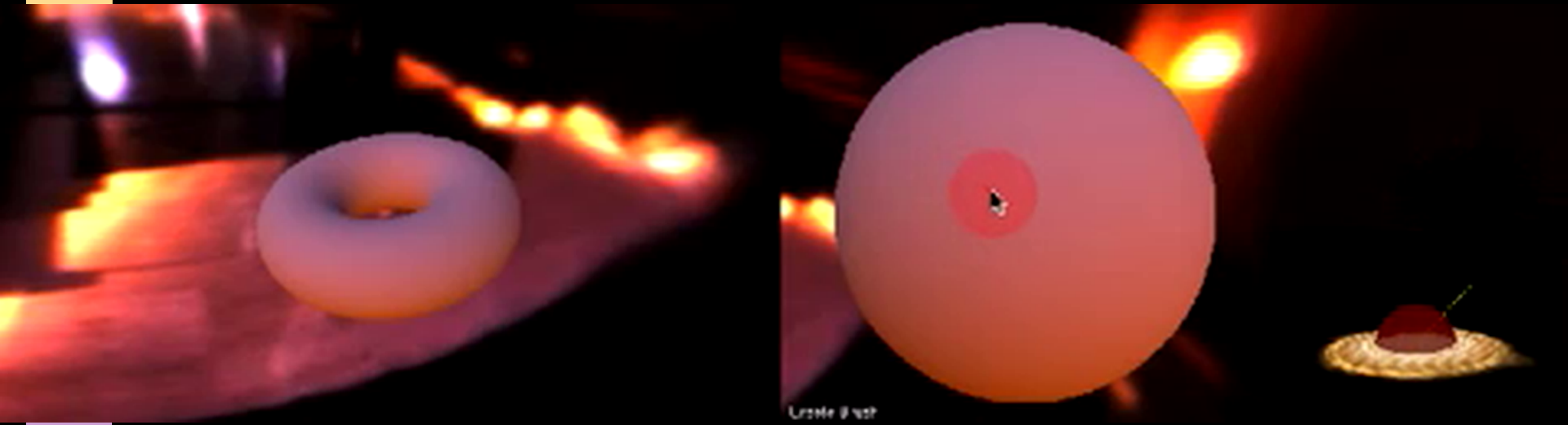
- Interactively position and manipulate specular highlights
- Brush modes: create, edit roughness, streaking, intensify
- Edit on sphere, scene updates in real-time



Colbert & Pattanaik, “BRDF-Shop: An artistic tool for creating physically correct BRDFs”, 2006

Background: Interactive BRDF Editing

BRDF-Shop



Colbert & Pattanaik, “BRDF-Shop: An artistic tool for creating physically correct BRDFs”, 2006

Background: Interactive BRDF Editing

BRDF-Shop

Advantages:

- Intuitive editing options
- Allows editing right on the surface

Disadvantages:

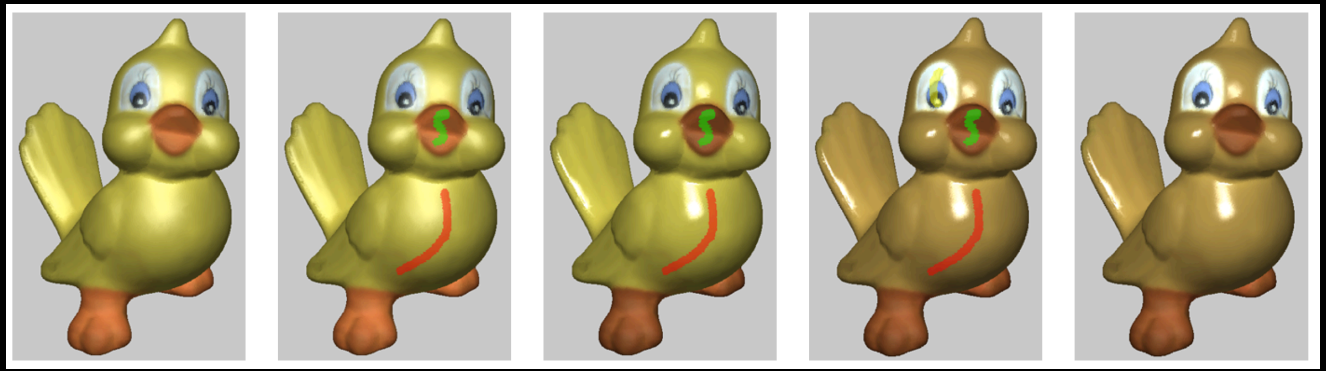
- Cannot paint on the object, must paint on sphere
- Editing limited to positioning and manipulating highlights

Colbert & Pattanaik, “BRDF-Shop: An artistic tool for creating physically correct BRDFs”, 2006

Background: Interactive BRDF Editing

Appwand

- User draws a stroke on the object to edit
- Changes are automatically propagated to similar regions
- Edit parameters of any BRDF model



Pellacini & Lawrence, “AppWand: Editing Measured Materials using Appearance-Driven Optimization”, 2007

Background: Interactive BRDF Editing

Appwand

Advantages:

- Allows for many different BRDF models
- User can draw directly on the object

Disadvantages:

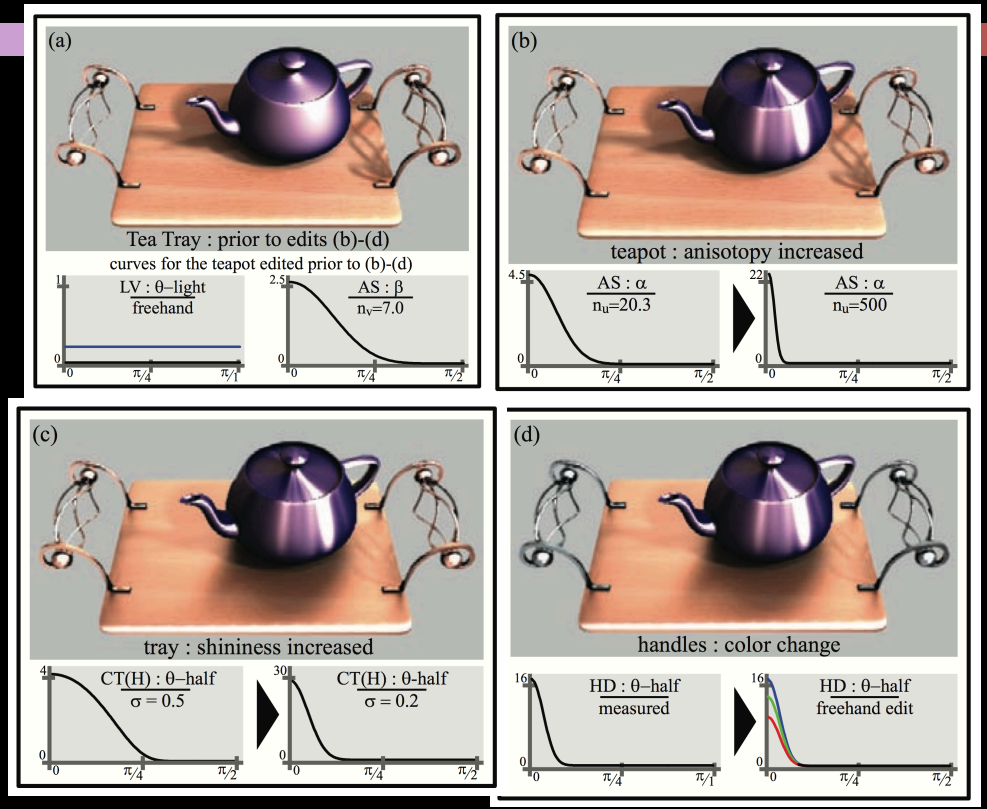
- Still need to specify values of BRDF parameters
- Mainly for global edits – local editing less intuitive

Pellacini & Lawrence, “AppWand: Editing Measured Materials using Appearance-Driven Optimization”, 2007

Background: Interactive BRDF Editing

Real-time BRDF Editing

- Use any BRDF
- User edits curves to specify parameters
- Rendered scene updates in real-time



Ben-Artzi et al., “Real-Time BRDF Editing in Complex Lighting”, 2006

Background: Interactive BRDF Editing

Ben-Artzi et al., “Real-Time BRDF Editing in Complex Lighting”, 2006

Background: Interactive BRDF Editing

Real-time BRDF Editing

Advantages:

- Support for any kind of BRDF and complex lighting
- Can parameterize BRDF in physically meaningful way

Disadvantages:

- Have to know what parameters mean or use trial-and-error
- Cannot paint directly on scene

Ben-Artzi et al., “Real-Time BRDF Editing in Complex Lighting”, 2006

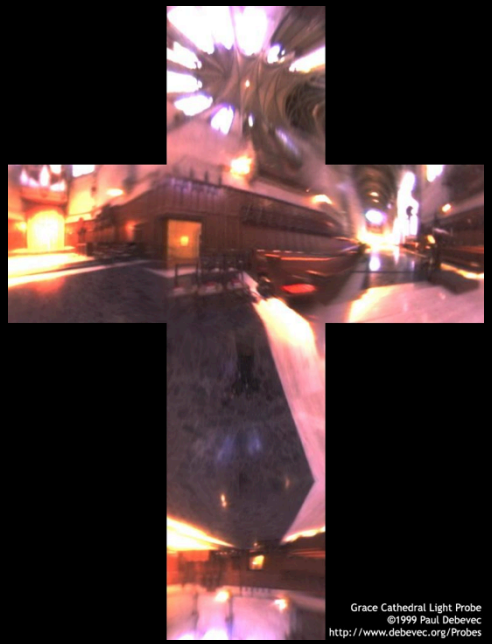
Overview

- Background
- Environment Mapping
- Implementation
- Next Steps

Environment Mapping

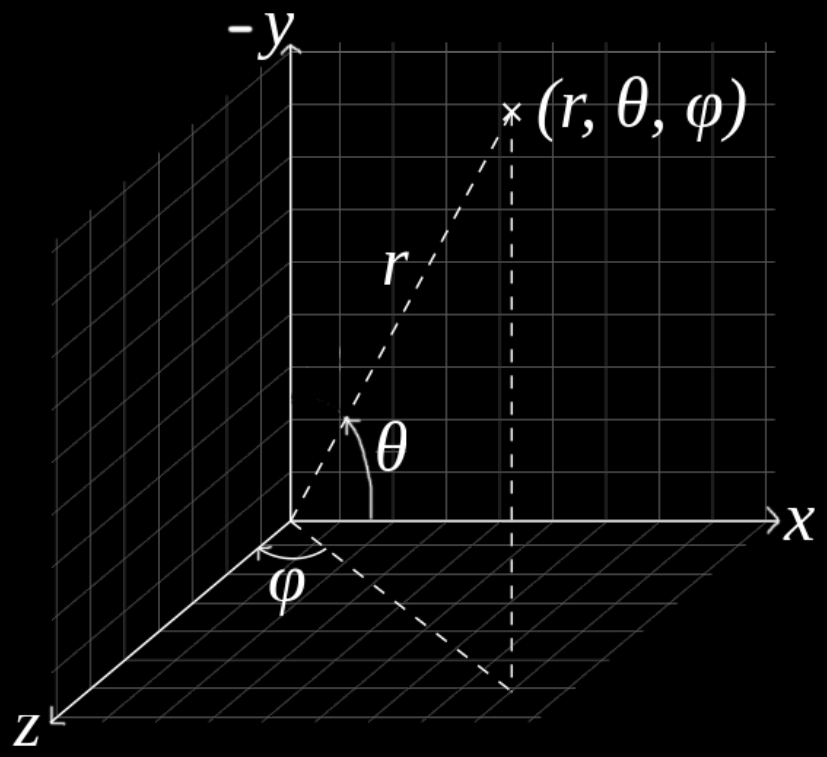
Miller & Hoffman, 1984

Environment Mapping

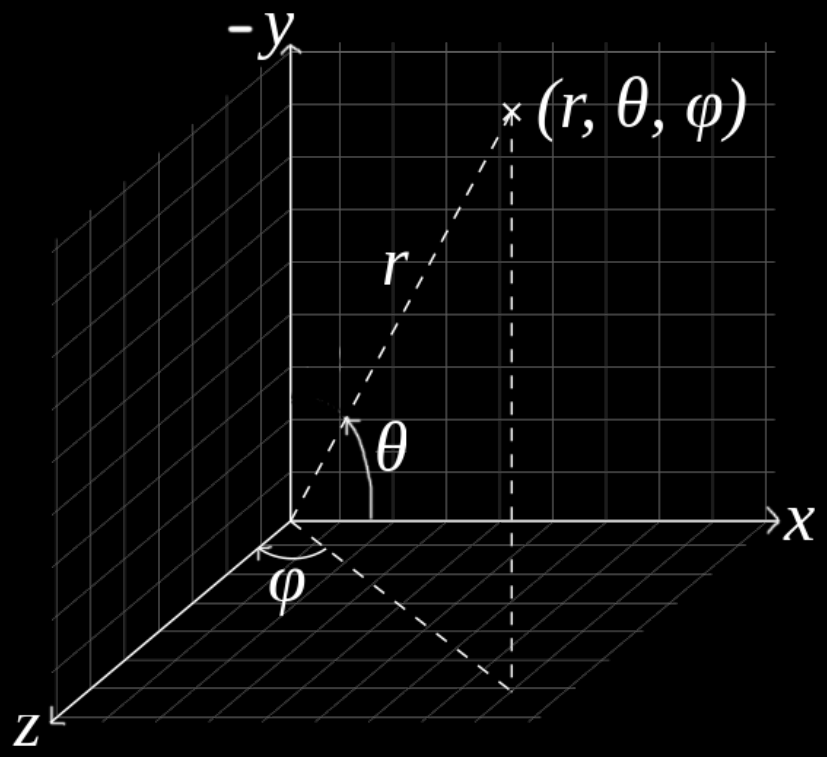


Paul Debevec, http://www.pauldebevec.com

Environment Mapping: Spherical coordinates



Environment Mapping: Spherical coordinates



$$x = -\sin\theta \sin\varphi$$
$$y = -\cos\theta$$
$$z = \sin\theta \cos\varphi$$

$$\theta = \arccos(-y) \qquad \in [0, \pi]$$
$$\varphi = \arctan(\frac{x}{-z}) + \pi \qquad \in [0, 2\pi]$$

$$u = \frac{\varphi N}{2\pi} \quad \in [0, N]$$
$$v = \frac{\theta M}{\pi} \quad \in [0, M]$$

Environment Mapping: Perfect Mirror

$$\vec{m} = -\vec{s} + (2\vec{s} \cdot \vec{n})\vec{n}$$
$$= (x, y, z)$$

$$\theta = \arccos(-y) \qquad \in [0, \pi]$$
$$\varphi = \arctan(\frac{x}{-z}) + \pi \qquad \in [0, 2\pi]$$

$$u = \frac{\varphi N}{2\pi} \quad \in [0, N]$$
$$v = \frac{\theta M}{\pi} \quad \in [0, M]$$

Environment Mapping: The Reflection Equation

$$I(\mathbf{x}, \vec{\omega}_o) = \int_\Omega L(\mathbf{x}, \vec{\omega}_i)\rho(\vec{\omega}_i, \vec{\omega}_o)\max(0, \vec{\omega}_i \cdot \vec{n})d\vec{\omega}_i$$

L = Incident lighting

ρ = BRDF

$\vec{n}$ = Surface normal at $\mathbf{x}$

$\vec{\omega}_i, \vec{\omega}_o$ = Incident and outgoing directions

Ramamoorthi & Hanrahan, 2001

Environment Mapping: Implementation

$$I(\mathbf{x}, \vec{\omega}_o) = \int_\Omega L(\mathbf{x}, \vec{\omega}_i)\rho(\vec{\omega}_i, \vec{\omega}_o)\max(0, \vec{\omega}_i \cdot \vec{n})\underbrace{d\vec{\omega}_i}_{=\sin\theta d\theta d\varphi}$$

$E = M$ by N environment map

$$I(\mathbf{x}) = \sum_{u\in[0,N]}\sum_{v\in[0,M]} E(u,v)\rho(\vec{\omega}_i, \vec{\omega}_o)\max(0, \vec{\omega}_i \cdot \vec{n})\underbrace{\sin(\frac{v\pi}{M})\frac{\pi}{M}\frac{2\pi}{N}}$$

Environment Mapping: BRDFs

Diffuse

$$\rho(\vec{\omega}_i, \vec{\omega}_o) = k_d$$

Environment Mapping: BRDFs

Diffuse

$$\rho(\vec{\omega}_i, \vec{\omega}_o) = k_d$$

Environment Map:

Environment Mapping: BRDFs

Specular: Normalized Phong

(Ramamoorthi & Hanrahan, 2001)

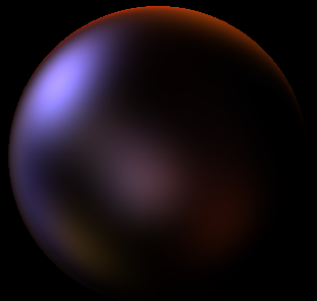
$$\rho(\vec{\omega}_i, \vec{\omega}_o) = \left(\frac{1+\sigma}{2\pi}\right) \cdot k_s \cdot \frac{\max(0, \vec{\omega}_o \cdot \vec{m})^\sigma}{\vec{\omega}_i \cdot \vec{n}}$$

Environment Mapping: BRDFs

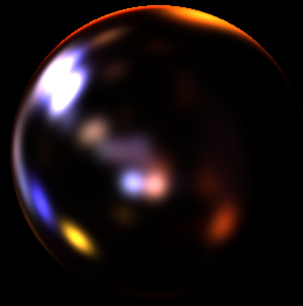
Specular: Normalized Phong

(Ramamoorthi & Hanrahan, 2001)

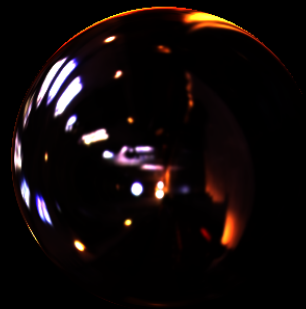
$$\rho(\vec{\omega}_i, \vec{\omega}_o) = \left(\frac{1+\sigma}{2\pi}\right) \cdot k_s \cdot \frac{\max(0, \vec{\omega}_o \cdot \vec{m})^\sigma}{\vec{\omega}_i \cdot \vec{n}}$$



σ = 10



σ = 100



σ = 5000

Environment Mapping: BRDFs

Specular: Torrance-Sparrow (simplified)

(Ramamoorthi & Hanrahan, 2001)

$$\rho(\theta_i, \theta_o) = k_s \frac{S}{4 \cos\theta_i \cos\theta_o} \quad , \quad S = \frac{1}{\pi\sigma^2} e^{-(\frac{\theta_h}{\sigma})^2}$$

$$\theta_h = \text{half angle between } \theta_i \text{ and } \theta_o$$

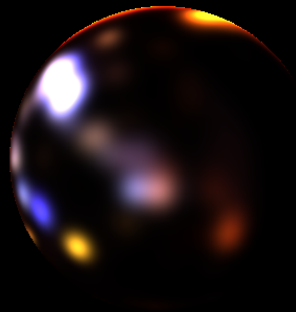
Environment Mapping: BRDFs

Specular: Torrance-Sparrow (simplified)

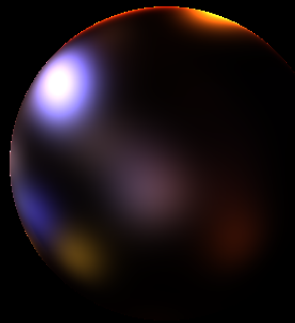
(Ramamoorthi & Hanrahan, 2001)

$$\rho(\theta_i, \theta_o) = k_s \frac{S}{4 \cos\theta_i \cos\theta_o} \quad , \quad S = \frac{1}{\pi\sigma^2} e^{-(\frac{\theta_h}{\sigma})^2}$$

σ = 0.01



σ = 0.1



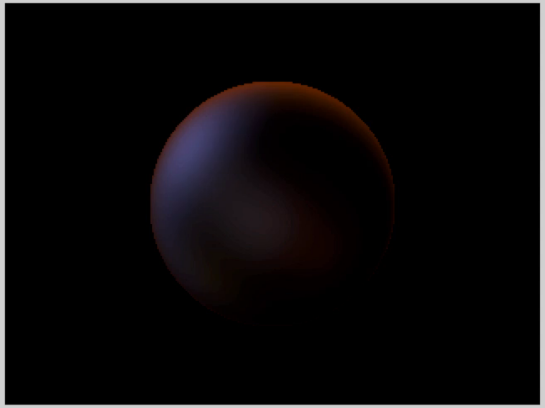
σ = 0.2

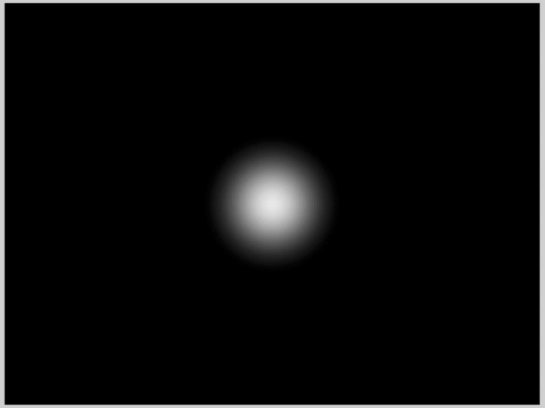
Overview

- Background
- Environment Mapping
- Implementation
- Next Steps

Implementation: Real-time editing

- Pre-render images of the sphere under environment and point source lighting
- Vary values of k_d, k_s and σ
- At runtime, add pre-rendered components together
- Interface to interactively edit diffuse colour and specular properties





Specular roughness

5

Adjust Highlight Spread

Draw

Diffuse Red

0

Specular Intensity

1

Blur/Sharpen Brushes

Blur Sharpen

Diffuse Green

0

Overall Scaling

1

Highlight Intensity Brush

Increase Decrease

Diffuse Blue

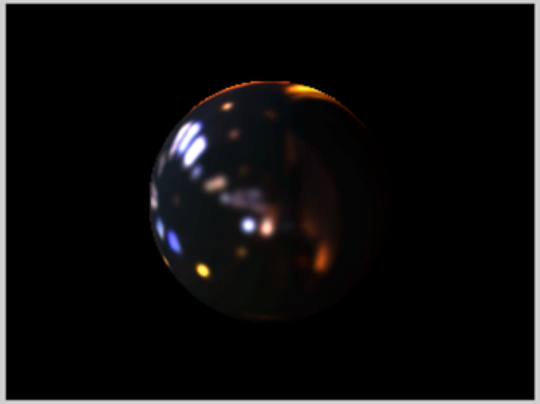
0

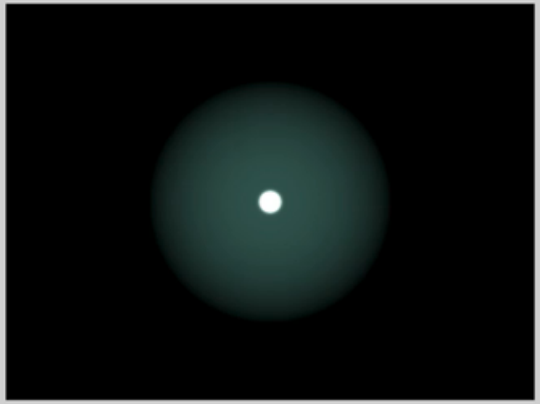
BRDF Model

Phong Torrance-Sparrow

Edit Diffuse Color

Change Overall Draw Colour





Specular roughness

0.04

Adjust Highlight Spread

Draw

Diffuse Red

0.2

Specular Intensity

1

Blur/Sharpen Brushes

Blur Sharpen

Diffuse Green

0.35

Overall Scaling

1

Highlight Intensity Brush

Increase Decrease

Diffuse Blue

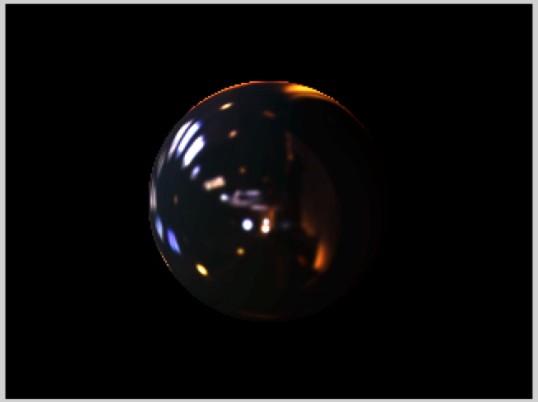
0.3

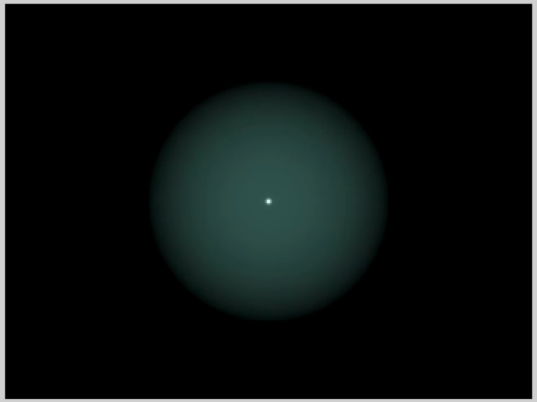
BRDF Model

Phong Torrance-Sparrow

Edit Diffuse Color

Change Overall Draw Colour





Specular roughness

1500

Adjust Highlight Spread

Draw

Diffuse Red

0.2

Specular Intensity

1

Blur/Sharpen Brushes

Blur Sharpen

Diffuse Green

0.35

Overall Scaling

1

Highlight Intensity Brush

Increase Decrease

Diffuse Blue

0.3

BRDF Model

Phong Torrance-Sparrow

Edit Diffuse Color

Change Overall Draw Colour

Overview

- Background
- Environment Mapping
- Implementation
- Next Steps

Next Steps

Goal:

- Real-time interactive BRDF editing system
- Paint edits directly on material
- Show objects under environment lighting
- Edit intuitive material properties
- Give artistic freedom while maintaining photorealism

Next Steps

Give artistic freedom while maintaining photorealism

- Artistic freedom: allow the user to paint whatever they want
- How to guarantee this will be photorealistic?
- How to solve for BRDF based on the painted edits?

Next Steps

Give artistic freedom

- Simple BRDFs are limited in the kinds of materials they can represent, and the edits that can be done
- The image with the “closest matching” BRDF parameters may not look like what the user painted

Next Steps

Edit intuitive material properties

- How to extend such a system to arbitrary BRDFs?
- What editing options should the user have?

Next Steps

Edit intuitive material properties

Example: Real-Time BRDF Editing

- Factor BRDF such that coefficients have meaningful effects when edited

Next Steps

Edit intuitive material properties

Example: A data-driven reflectance model (Matusik, 2003)

- Treat each BRDF as a high-dimensional vector
- Define a set of descriptive parameters (e.g. shiny, plastic)
- Classify a large set of BRDFs based on these
- Derive “trait vector” for each category
- Make edits based on these traits

Thank you!

Questions?