Multi-Textured BRDF-based Lighting

Chris Wynn

Overview

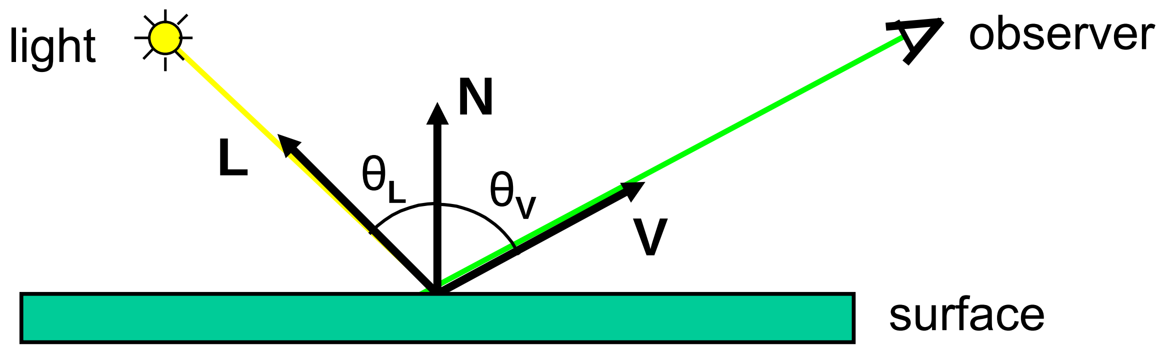
- What is a BRDF?
- The BRDF Lighting Equation
- Multi-texture BRDF Approximations
  - Single term approximations
  - Multi-term approximations
- Cool stuff with BRDFs
- Demos and Discussion

What is a BRDF?

- BRDF = Bi-directional reflectance distribution function
- Describes how light is reflected when it interacts with a surface material
- Physically-based
  - Analytical
  - Measured Data
- *Offers increased level of lighting "realism" for computer graphics*

What is a BRDF?

- **BRDF is a function of incoming light direction and outgoing view direction**



- **In 3D, a direction D can be represented in spherical coordinates (θ_D, ϕ_D)**
- **A BRDF is a 4D function: BRDF(θ_L, ϕ_L, θ_V, ϕ_V)**

The BRDF Lighting Equation

- Derived using properties of calculus

- Single Point Light Source

$$I_V = I_L * BRDF(\theta_L, \varphi_L, \theta_V, \varphi_V) * \cos(\theta_L)$$

where

$I_V \equiv$ intensity of light reflected in direction V from the surface point

$I_L \equiv$ intensity of light arriving at the surface point from direction L

The BRDF Lighting Equation

- **Multiple Point Light Sources**

$$\mathbf{I_V} = \sum_{i=1}^{n} \left(\mathbf{I_{L_i}} * \mathbf{BRDF}(\theta_{L_i}, \varphi_{L_i}, \theta_V, \varphi_V) * \mathbf{cos}(\theta_{L_i}) \right)$$

where

$I_V \equiv$ intensity of light reflected in direction V from the surface point

$L_i \equiv$ direction to i^{th} light source

$I_{L_i} \equiv$ intensity of light arriving at the surface point from direction L_i

The BRDF Lighting Equation

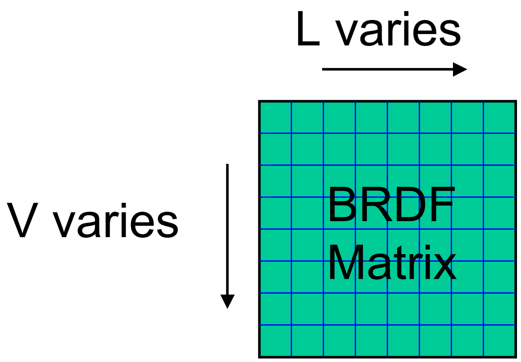
- **Fundamental Problems:**
  - **Given that a BRDF is a 4D function, how can we evaluate the lighting equation using current graphics HW?**
  - **How can we evaluate the lighting equation on a per-pixel basis?**

Multi-Texture BRDF Approximations

- **Basic Idea:**
  - **Approximate the 4D function with lower dimensional functions**
  - **“Separate” the BRDF into products of simpler functions**
  - **$BRDF(L,V) \cong G_1(L)*H_1(V) + G_2(L)*H_2(V) + \ldots$**

Multi-Texture BRDF Approximations

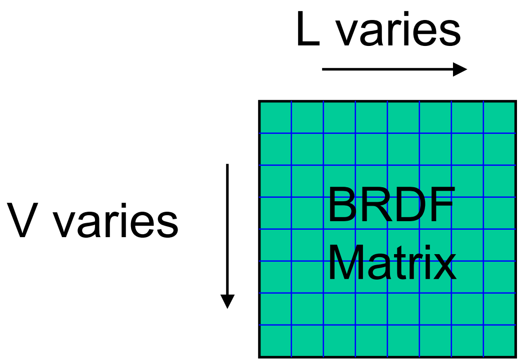
- **How it works:**



  - **Sample the BRDF and "unroll" the 4D function into a 2D matrix**
  - **Each column (or row) corresponds to a single incoming light direction (or outgoing view direction).**

Multi-Texture BRDF Approximations

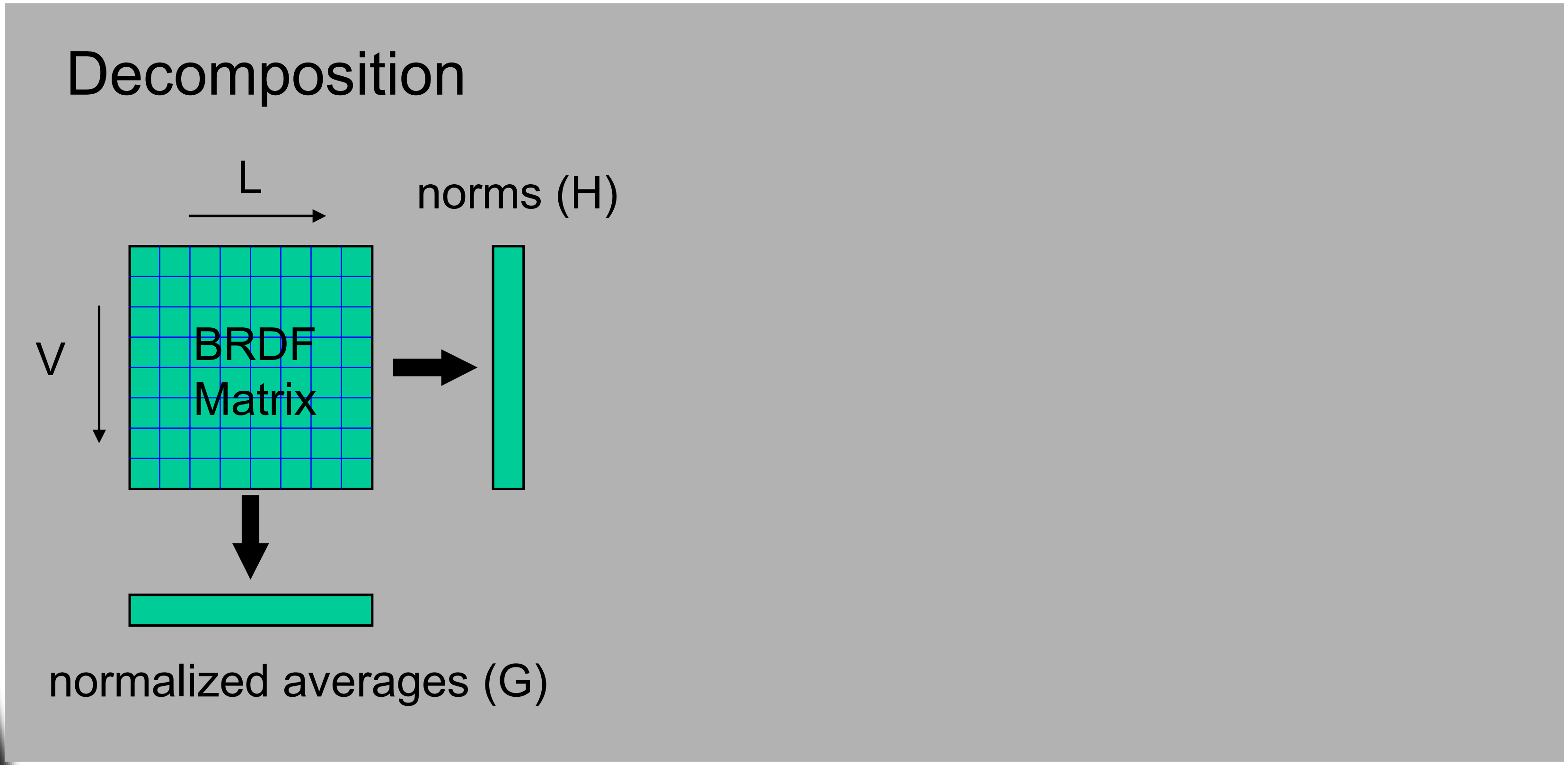
- **How it works (cont):**



  - **Once the matrix has been created, perform a matrix decomposition to produce factors.**
    - **Singular Value Decomposition (SVD)**
    - **Normalized Decomposition (ND)**

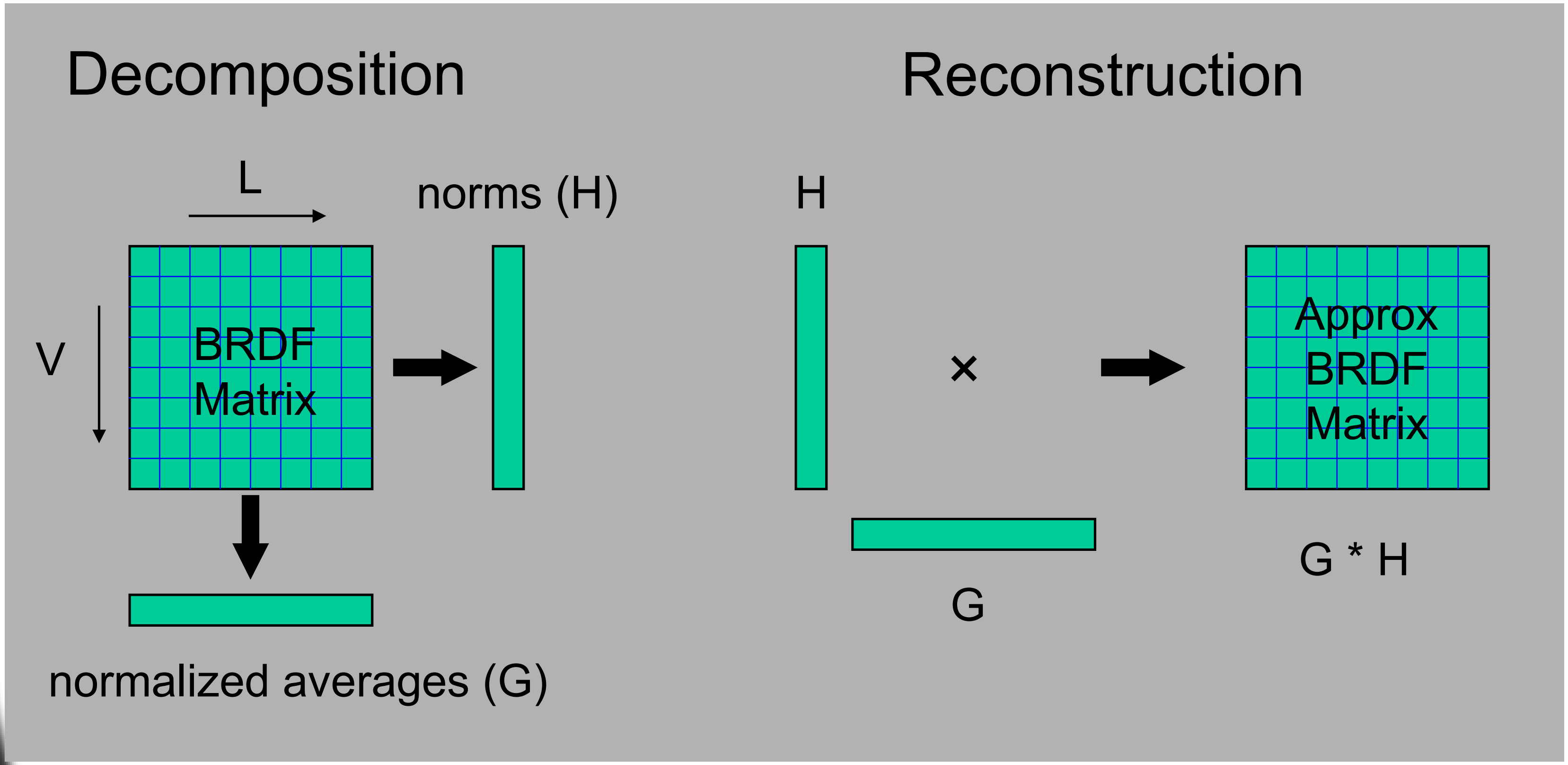
Multi-Texture BRDF Approximations

- **Normalized Decomposition**
  - **Produces a single term approximation**



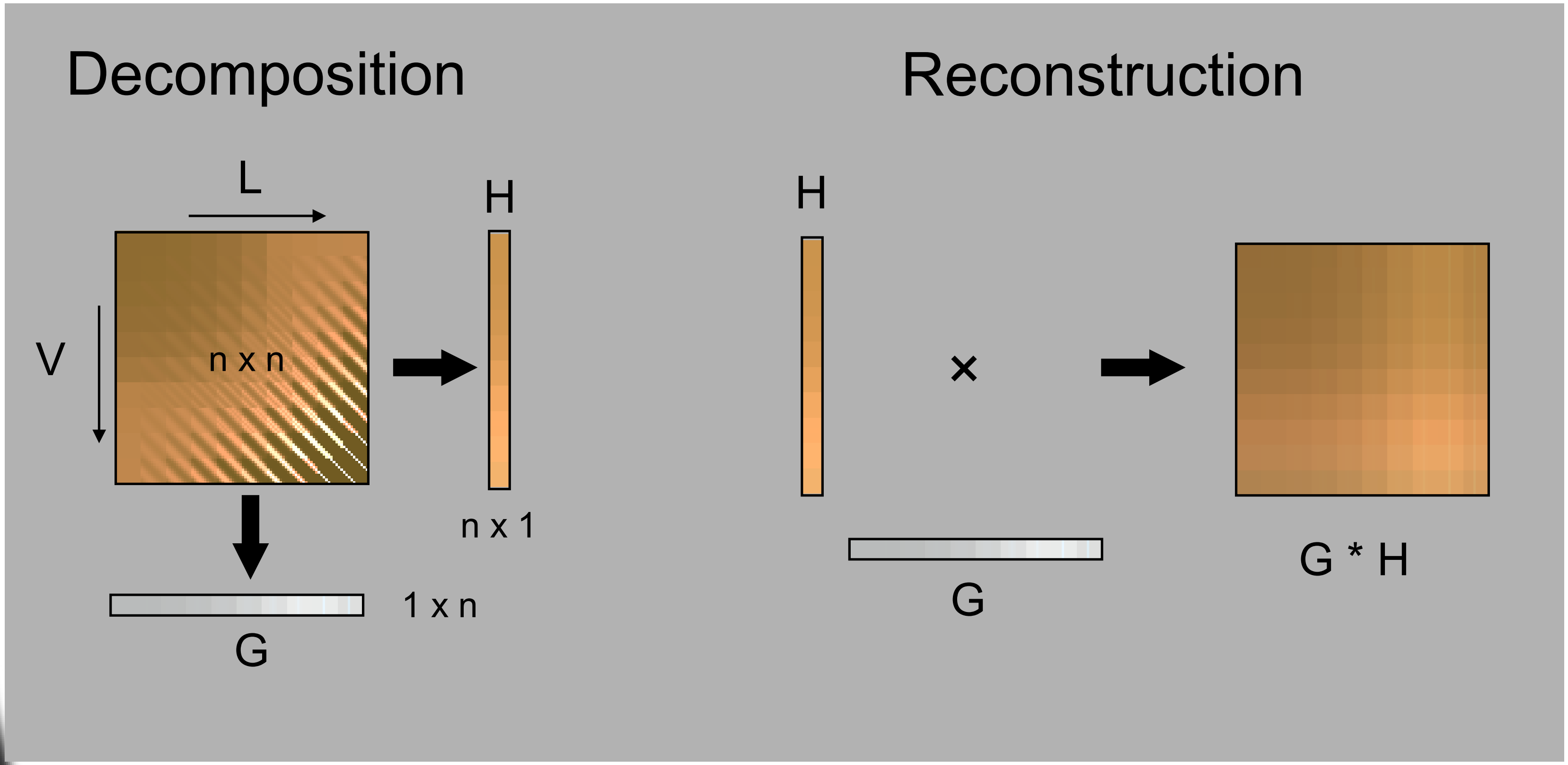
Multi-Texture BRDF Approximations

- **Normalized Decomposition**
  - **Produces a single term approximation**



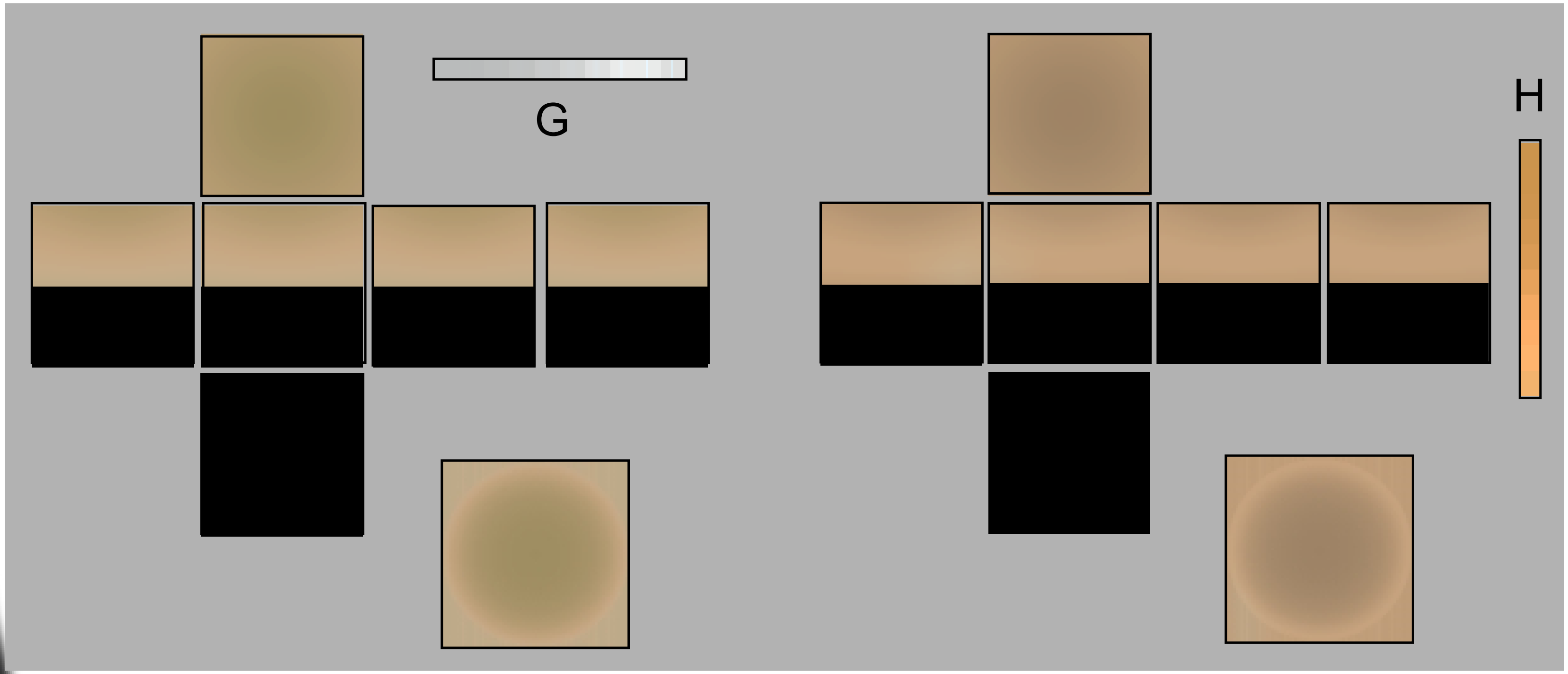
Multi-Texture BRDF Approximations

- **Normalized Decomposition (Example)**



Multi-Texture BRDF Approximations

- **Normalized Decomposition (Example)**
  - **Scale & Resample functions to create textures**
    - **Cube Maps, Dual-Parabaloid, or 2D Textures**



Multi-Texture BRDF Approximations

- **Normalized Decomposition (Example)**
  - **At Run-Time, compute L and V vectors per-vertex.**
  - **Derive texture coordinates.**
  - **Apply multi-texturing to compute G *H per-pixel**

Multi-Texture BRDF Approximations

- **Multi-Term Approximations**
  - **Use SVD to compute:**

    $$\mathbf{BRDF(L,V) \cong G_1(L)*H_1(V) + G_2(L)*H_2(V) + \ldots}$$

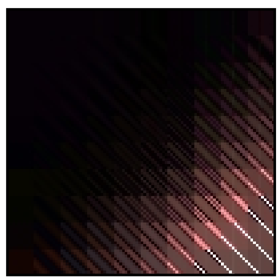
  - **Use error from ND to compute $G_2(L)$ and $H_2(V)$.**

    $$\mathbf{Error(L,V) = BRDF(L,V) - G(L)*H(V)}$$

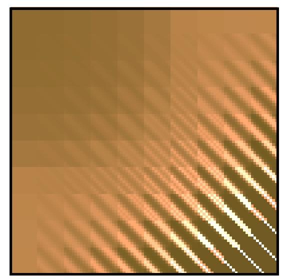
Multi-Texture BRDF Approximations

- **Multi-Term Approximations**

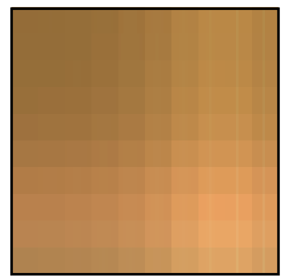
$$\text{Error}(L,V) = \text{BRDF}(L,V) - G_1(L)*H_1(V)$$



=



–



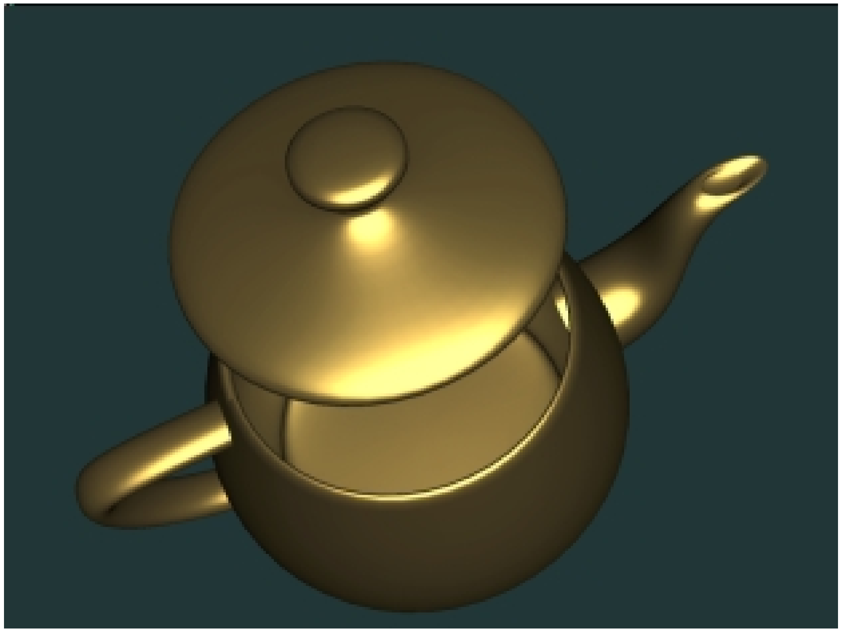
Perform ND on Error to find G_2 and H_2

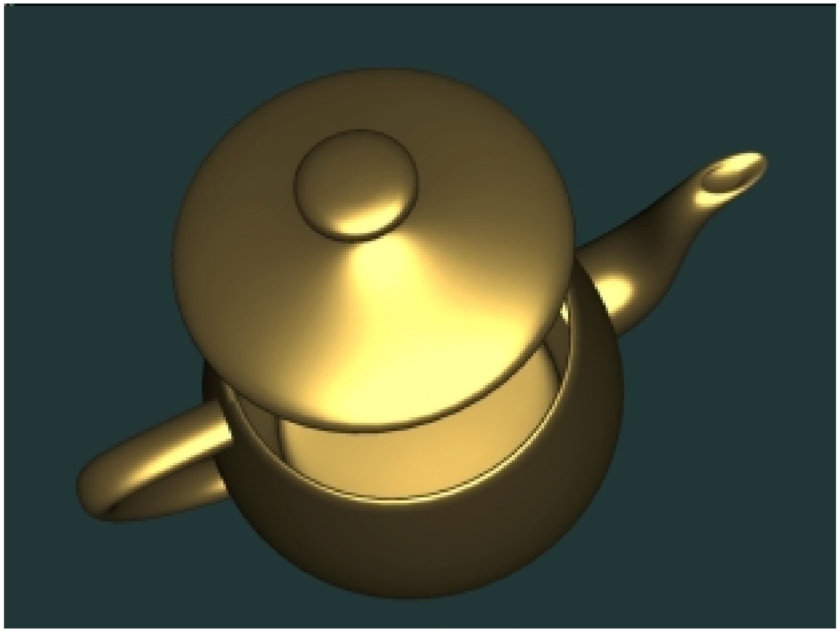
Multi-Texture BRDF Approximations

- Multi-Term Approximations
  - Multiple terms improve results but require multiple textures.
    - One Term $\Rightarrow$ 2 textures
    - Two Term $\Rightarrow$ 4 textures
  - Perhaps there is other cool stuff we can do with additional textures.

Cool Stuff with BRDFs

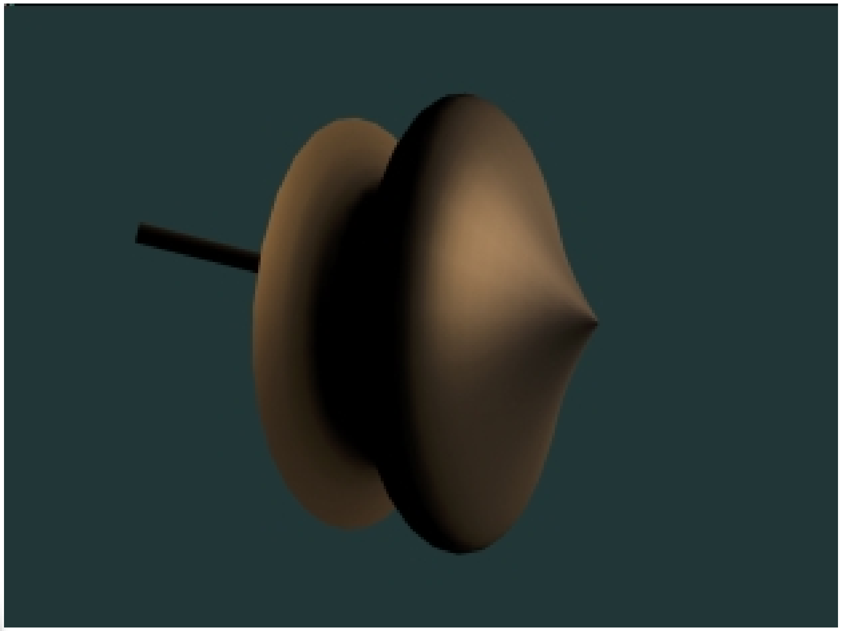
- **Single-Term BRDF-based lighting**

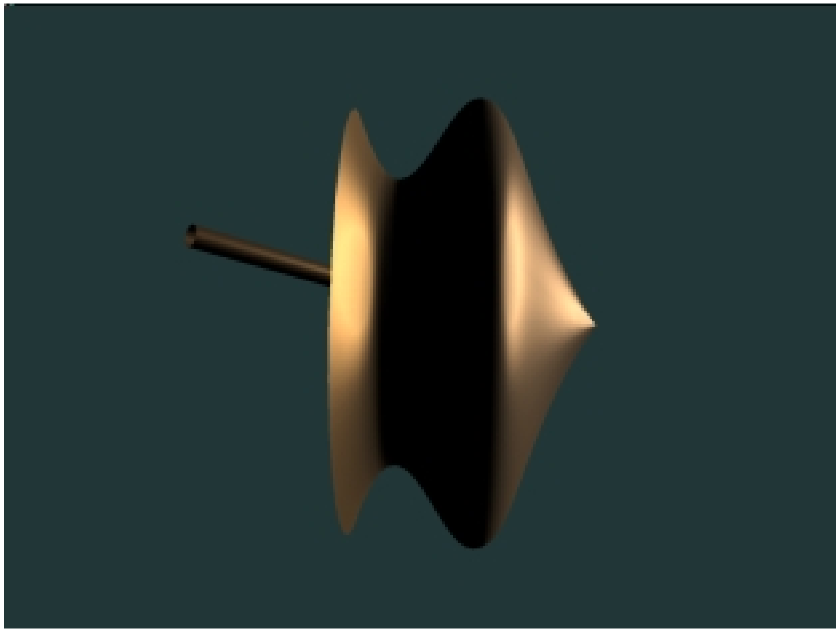




Cool Stuff with BRDFs

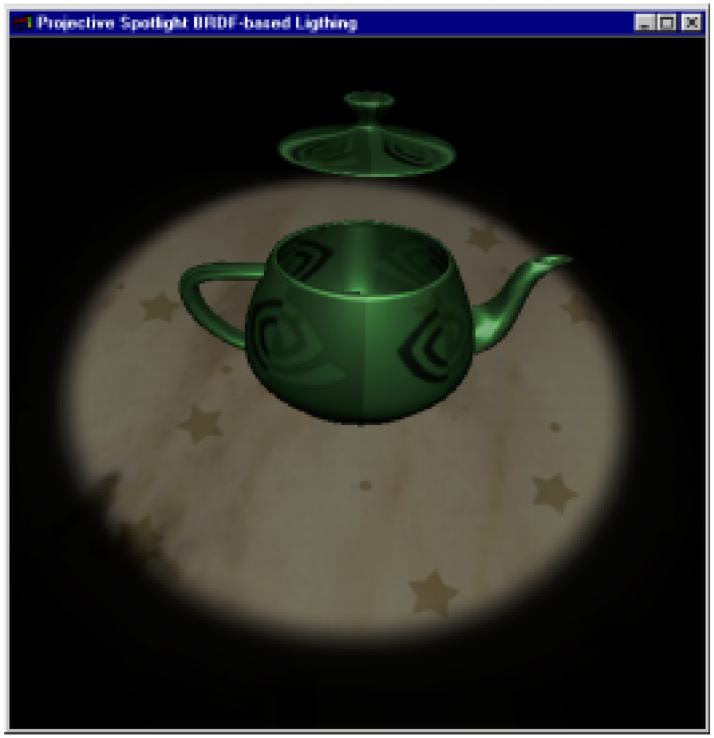
- **Single-Term BRDF-based lighting**

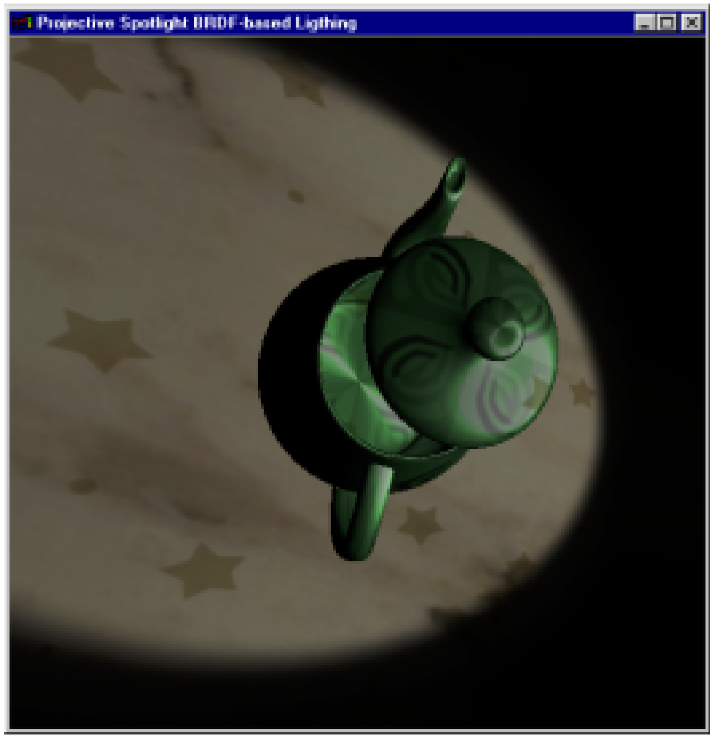




Cool Stuff with BRDFs

- **Single-Term BRDF + Spotlight + Decal**





Demos and Discussion

- **Questions?**

Questions, comments, feedback

- Chris Wynn, cwynn@nvidia.com
- www.nvidia.com/developer