Inverse Rendering

Inverse rendering problems could be divided into three types.

- Given geometry, reflectance, and the recorded photograph, solves for the lighting in the scene
  - Inverse lighting
  - Re-lighting
- Given geometry, lighting and photograph, solves for
  - texture
  - BRDF
- Given lighting, photograph, solves for
  - normals
  - depth map
  - reflectance

Photometric Stereo

We want to measure the surface normal of an object in real life using the direct illumination rendering equation we have learned.

$$L_o(\mathbf{x}, \vec{\omega}_o) = \int_{H^2} f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_o) L_i(\mathbf{x}, \vec{\omega}_i)(\vec{\omega}_i \cdot \vec{n})\, d\vec{\omega}_i$$

Assume that we are using active illumination, which means:

- Incident radiance is known and controllable
- Indirect illumination is considered negligible
- Directional light is used
- Incident direction and radiance are controlled to probe properties of the material
  thus we have

  $$L_i(\mathbf{x}, \vec{\omega}_i) = L_i(\vec{\omega}_i)$$

  Assume also
- The directional light is delta

  $$L_i(\vec{\omega}_i) = \delta(\vec{\omega}_i)$$

- The BRDF is Lambertian

  $$f_r = \frac{\rho_d}{\pi}$$

- We are using orthographic camera
- We have a single view of the object

Then the rendering equation reduces to

$$I = \frac{\rho_d}{\pi}(\vec{n} \cdot \vec{\omega}_i)$$

which is a linear system equivalent to

$$A_{n\times 3}\mathbf{x}_{3\times 1} = \mathbf{b}_{n\times 1}$$

where

$$A_{n\times 3} = \begin{bmatrix} \vec{\omega}_{i,1}^\top \\ \vdots \\ \vec{\omega}_{i,n}^\top \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} I_1 \\ \vdots \\ I_n \end{bmatrix}, \quad \mathbf{x} = \frac{\rho_d}{\pi}\vec{n}$$

We need at least 3 lighting conditions to solve the system because we have 3 unknowns: $\vec{n} = (\theta, \phi)$ and ρ_d. The solution is thus

$$\vec{n} = \frac{\mathbf{x}}{\|\mathbf{x}\|}, \quad \rho_d = \pi\|\mathbf{x}\|$$

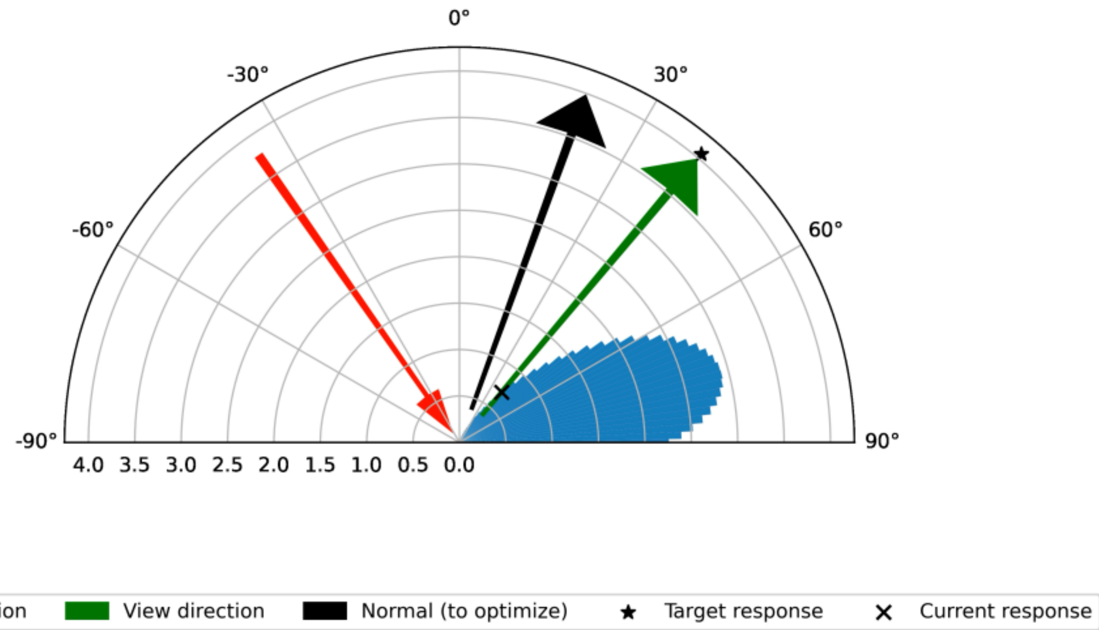
Suppose now we don't assume Lambertian reflectance anymore, but use Blinn-Phong model instead, i.e.

$$f_r(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) = \frac{e+2}{2\pi}(\boldsymbol{\omega}_h \cdot \mathbf{n})^e$$

with

$$\boldsymbol{\omega}_h = \frac{\boldsymbol{\omega}_i + \boldsymbol{\omega}_o}{\|\boldsymbol{\omega}_i + \boldsymbol{\omega}_o\|}$$

The system is then no longer linear w.r.t. $\vec{n}$ and we have no closed-form solution. However, we could still use gradient descent to find a minimal.



Taking the derivative of L_o w.r.t. $\vec{n}$, we could see the system is extremely complicated

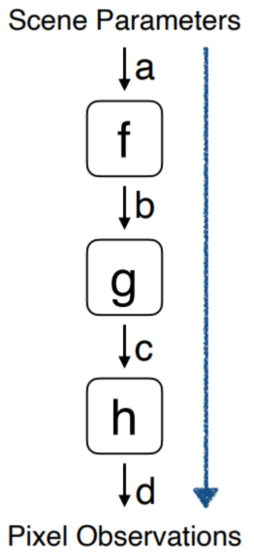
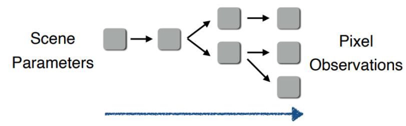
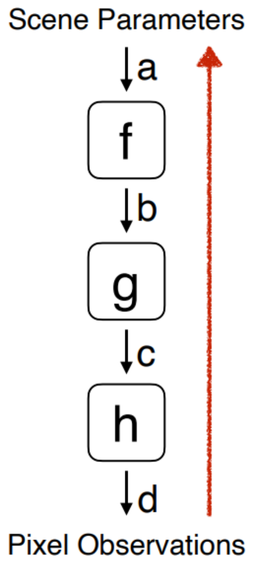
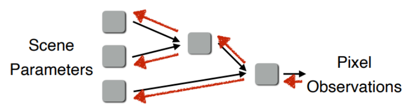
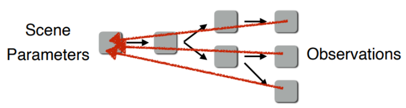
$$\frac{\partial L_o}{\partial \vec{n}} = \int_{H^2} \frac{e+2}{2\pi} L_i\left(\vec{\omega}_i\right)\left(e\left(\vec{\omega}_h \cdot \vec{n}\right)^{e-1}\vec{\omega}_h\left(\vec{\omega}_i \cdot \vec{n}\right) + \left(\vec{\omega}_h \cdot \vec{n}\right)^e \vec{\omega}_i\right) d\vec{\omega}_i$$

We could still make it by

- symbolic differentiation
- finite differences
- automatic differentiation

Automatic Differentiation

Mode	Representation	Formula	Pros/ Cons

Mode	Representation	Formula	Pros/ Cons
Forward	 $\frac{\mathrm{dd}}{\mathrm{da}} = \frac{\partial h(c)}{\partial c}\frac{\partial g(b)}{\partial b}\frac{\partial f(a)}{\partial a}$	$\tilde{a} = \frac{\mathrm{da}}{\mathrm{da}} = 1$ $\tilde{b} = \frac{\mathrm{db}}{\mathrm{da}} = \frac{\partial f(a)}{\partial a}\tilde{a}$ $\tilde{c} = \frac{\mathrm{dc}}{\mathrm{da}} = \frac{\partial g(b)}{\partial b}\tilde{b}$ $\tilde{d} = \frac{\mathrm{dd}}{\mathrm{da}} = \frac{\partial h(c)}{\partial c}\tilde{c}$	- Easy to implement - Supports multiple outputs  - Does not supports multiple inputs: expensive
Backward	 $\frac{\mathrm{dd}}{\mathrm{da}} = \frac{\partial h(c)}{\partial c}\frac{\partial g(b)}{\partial b}\frac{\partial f(a)}{\partial a}$	$\hat{d} = \frac{\mathrm{dd}}{\mathrm{dd}} = 1$ $\hat{c} = \frac{\mathrm{dd}}{\mathrm{dc}} = \hat{d}\frac{\partial h(c)}{\partial c}$ $\hat{b} = \frac{\mathrm{dd}}{\mathrm{db}} = \hat{c}\frac{\partial g(b)}{\partial b}$ $\hat{a} = \frac{\mathrm{dd}}{\mathrm{da}} = \hat{b}\frac{\partial f(a)}{\partial a}$	- Supports multiple inputs  - Supports only a single output  - Store the whole graph and meta-information per node

Volume Optimization

Recall that the transmittance of an volume is given by

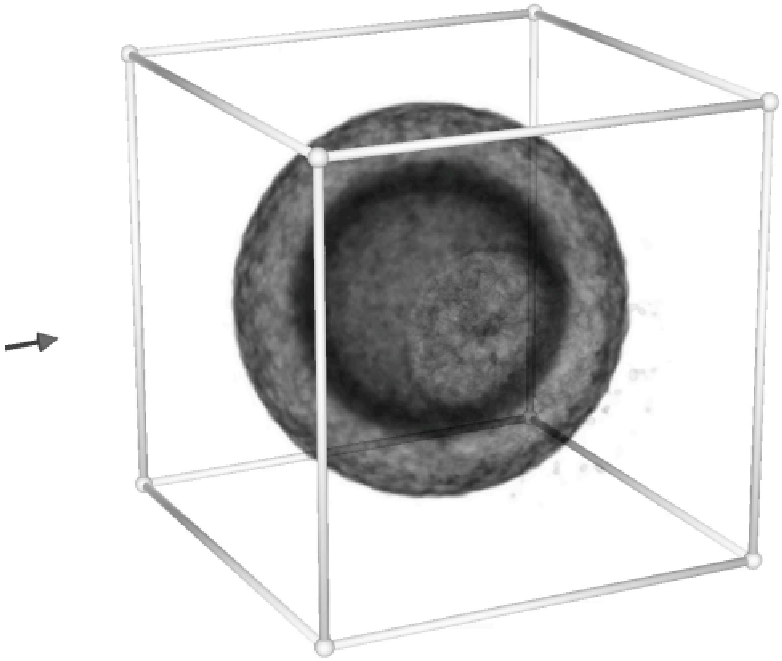
$$T_r(\mathbf{x}, \mathbf{x}_z) = e^{-\int_{\mathbf{x}}^{\mathbf{x}_z} \sigma(\mathbf{x})\, d\mathbf{x}} = \frac{L_{\mathrm{gt}}(\mathbf{x}, \omega)}{L_o(\mathbf{x}_z, \omega)}$$

which could be rearranged to

$$\int_{\mathbf{x}}^{\mathbf{x}_z} \sigma(x)dx = \underbrace{-\log\left(L_{\mathrm{gt}}(\mathbf{x}, \omega)/L_o\left(\mathbf{x}_z, \omega\right)\right)}_{=:b}$$

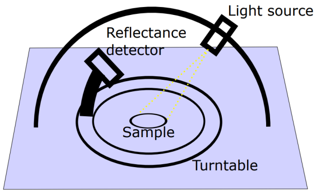
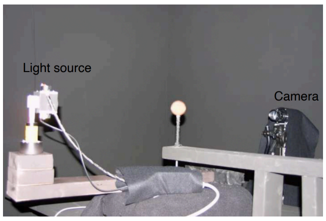
The left side could be replaced by an numerical approximation (ray marching) which gives a linear system

$$W_{\#\mathrm{pixels}\times\#\mathrm{voxels}} \cdot \sigma_{\#\mathrm{voxels}} = b_{\#\mathrm{pixels}}$$



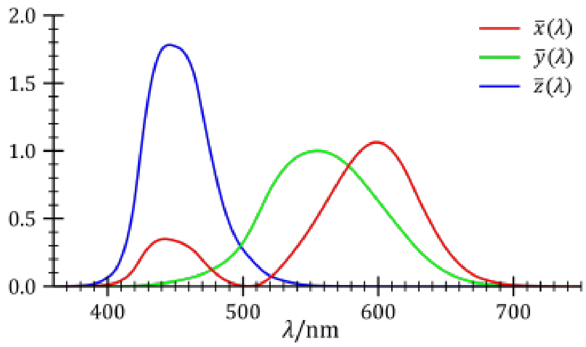
BRDF Measurement

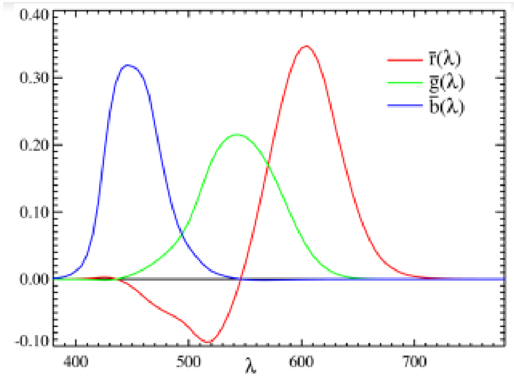
We could use a gonioreflectometer to measure a material's BRDF

Gonioreflectometer	EPFL RGL	MERL Database Setup
		

Colorimetric Calibration

Color Spaces

Color Space	Spectrum	Formula
CIE 1931 XYZ - Y: Luminance - XZ plane: Chromaticity		$X = \int_\lambda L_{e,\lambda}(\lambda)\bar{x}(\lambda)d\lambda$ $Y = \int_\lambda L_{e,\lambda}(\lambda)\bar{y}(\lambda)d\lambda$ $Z = \int_\lambda L_{e,\lambda}(\lambda)\bar{z}(\lambda)d\lambda$

Color Space	Spectrum	Formula
CIE 1931 RGB		$R = \int_\lambda L_{e,\lambda}(\lambda)\bar{r}(\lambda)d\lambda$ $G = \int_\lambda L_{e,\lambda}(\lambda)\bar{g}(\lambda)d\lambda$ $B = \int_\lambda L_{e,\lambda}(\lambda)\bar{b}(\lambda)d\lambda$

Calibration

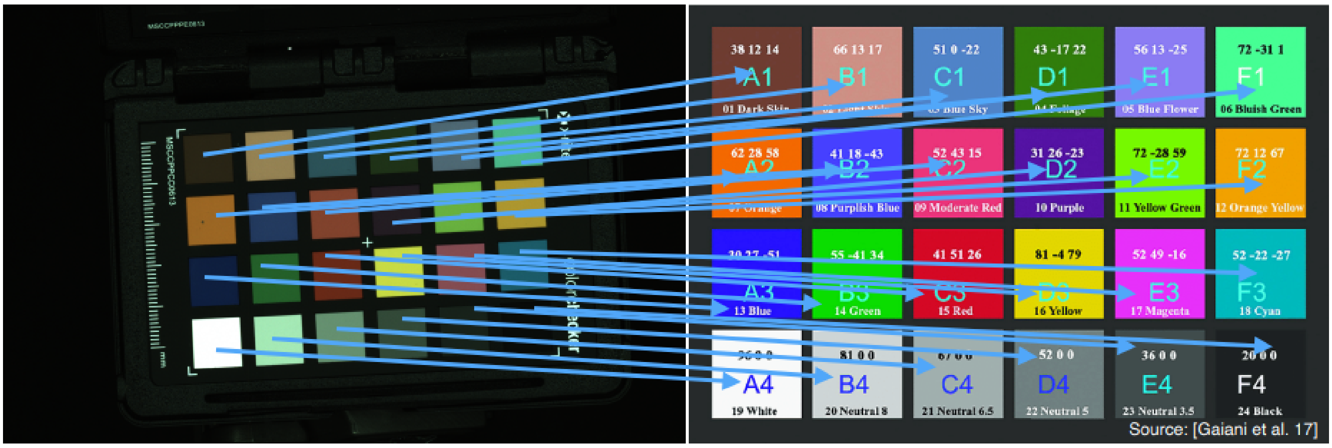


Photo (camera raw colour space) Target values

Like camera calibration, we could build a linear system based on pixel correspondences:

$$A_{24\times3}\mathbf{X}_{3\times3} = \mathbf{b}_{24\times3}$$

The transformation is then

$$C_{RGB'2XYZ} = \mathbf{X}^{-1}$$

In plus, any RGB color space may be obtained by a simple(known) linear transformation:

$$C_{RGB'2sRGB} = C_{RGB'2XYZ} \cdot C_{XYZ_2sRGB}$$

where the matrices could be found at RGB/XYZ Matrices (brucelindbloom.com)

Lighting Estimation

Photo of mirror sphere placed in capture volume

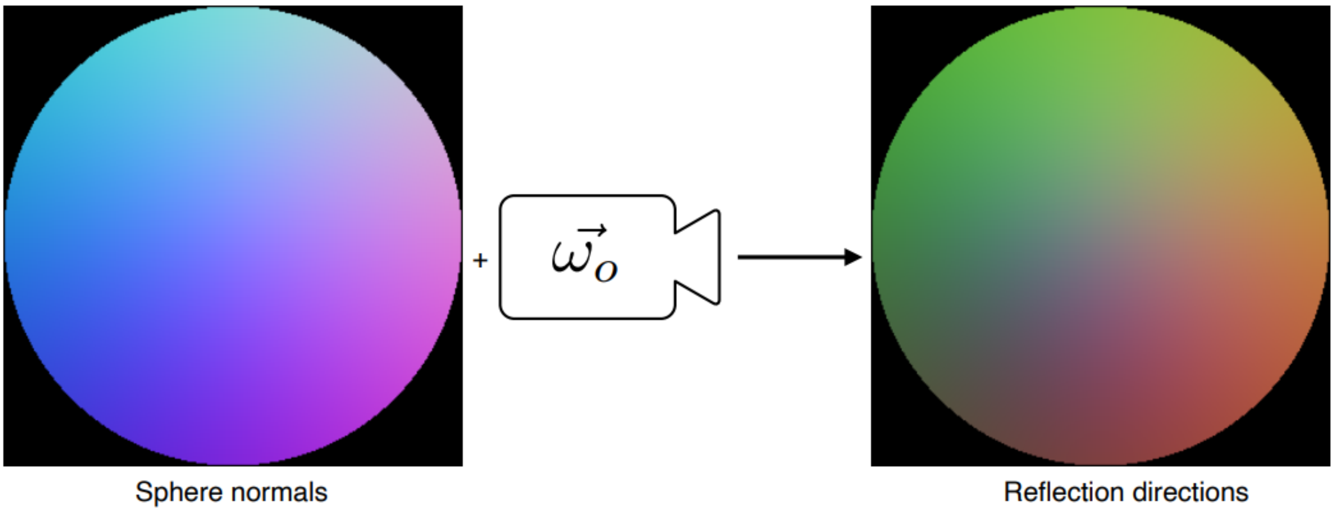
Consider a specular perfect sphere under a spotlight and we want to estimate the incidence ray direction. Assume that we are using orthographic camera and

$$\vec{\Omega}_o = (0, 0, 1)^\top$$

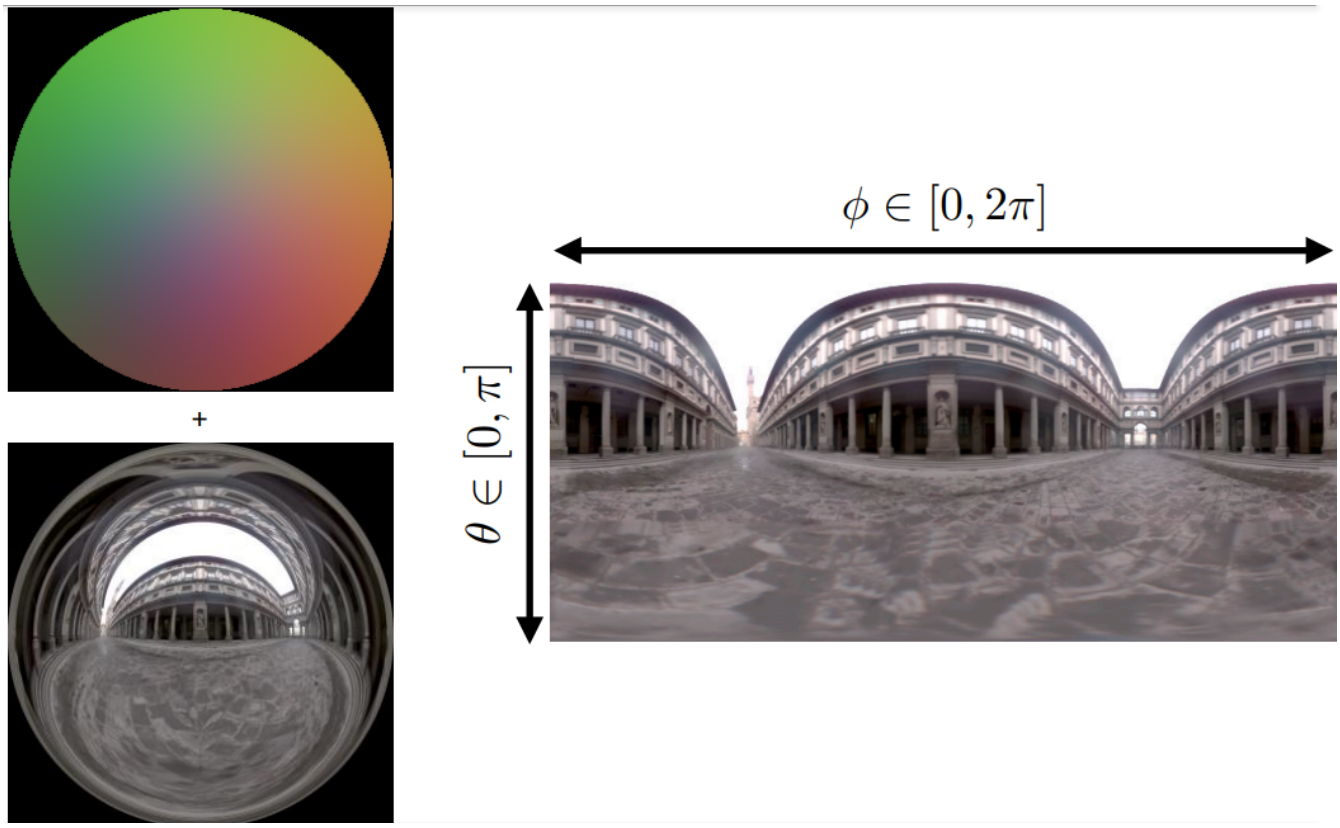
By the rule of reflection,

$$\vec{\omega}_i = 2(\vec{\omega}_o \cdot \vec{n})\vec{n} - \vec{\omega}$$

and the sphere normals by analytical sphere



we could have a reflection map. Besides we could use this map to recover the environment map



Reference

[Physics-Based Differentiable Rendering: A Comprehensive Introduction (shuangz.com)]
[Inverse Rendering for Computer Graphics (cornell.edu)]
[Acquiring Material Models Using Inverse Rendering]