Lighting and Reflectance

The squares marked A and B are the same shade of gray

Lighting

Let's go to paintbrush

Lighting

Lighting can have a big effect on how an object looks.

Modeling the effect of lighting can be used for:
Recognition – particularly face recognition
Shape reconstruction
Motion estimation
Re-rendering / Re-lighting

...
Lighting is Complex

Lighting can come from any direction and at any strength. Infinite degree of freedom.

Capture lighting variation

Illuminate subject from many incident directions.

Example images:

\[ \text{image} = .2 \times \text{image} + .3 \times \text{image} + \ldots \]
**Image brightness**

What determines the brightness of an image pixel?

- Lighting
- Surface BRDF (local reflectance)
- Shadowing
- Inter-reflections (global reflectance)

**What is light?**

Electromagnetic radiation (EMR) moving along rays in space

- \( R(\lambda) \) is EMR, measured in units of power (watts)
  - \( \lambda \) is wavelength

Light field

- We can describe all of the light in the scene by specifying the radiation (or "radiance" along all light rays) arriving at every point in space and from every direction

\[ R(X, Y, Z, \theta, \phi, \lambda, t) \]

**The light field**

\[ R(X, Y, Z, \theta, \phi, \lambda, t) \]

- Known as the plenoptic function
- If you know \( R \), you can predict how the scene would appear from any viewpoint.
- Common to think of lighting at infinity (a function on the sphere, a 2D space)
- Usually drop \( \lambda \) and time parameters

**Stanford light field gantry**

A degree-of-freedom gantry
What is light?

Electromagnetic radiation (EMR) moving along rays in space
- \( R(\lambda) \) is EMR, measured in units of power (watts)
  - \( \lambda \) is wavelength

Perceiving light
- How do we convert radiation into "color"?
- What part of the spectrum do we see?

Visible light

We "see" electromagnetic radiation in a range of wavelengths

Light spectrum

The appearance of light depends on its power spectrum
- How much power (or energy) at each wavelength

Our visual system converts a light spectrum into "color"
- This is a rather complex transformation
Light transport

Light sources

Basic types
- point source
- Distant point source
- area source
  - a union of point sources

More generally
- a light field can describe *any* distribution of light sources

What happens when light hits an object?

Reflectance spectrum (albedo)

To a first approximation, surfaces absorb some wavelengths of light and reflect others

These spectra are multiplied by the spectra of the incoming light

Typical Reflections

ideal specular
rough specular
Lambertian

from Steve Marschner
What happens when a light ray hits an object?

Some of the light gets absorbed
- converted to other forms of energy (e.g., heat)

Some gets transmitted through the object
- possibly bent, through “refraction”
- a transmitted ray could possible bounce back

Some gets reflected
- as we saw before, it could be reflected in multiple directions (possibly all directions) at once

Let’s consider the case of reflection in detail

The BRDF

The Bidirectional Reflection Distribution Function
- Given an incoming ray \((\theta_i, \phi_i)\) and outgoing ray \((\theta_e, \phi_e)\)
  what proportion of the incoming light is reflected along outgoing ray?

Answer given by the BRDF: \(\rho(\theta_i, \phi_i, \theta_e, \phi_e)\)

Constraints on the BRDF

Energy conservation
- Quantity of outgoing light ≤ quantity of incident light
  - integral of BRDF ≤ 1

Helmholtz reciprocity
- reversing the path of light produces the same reflectance

Diffuse (Lambertian) reflection

Diffuse reflection
- Dull, matte surfaces like chalk or latex paint
- Microfacets scatter incoming light randomly
- Effect is that light is reflected equally in all directions
Diffuse reflection

- Viewed brightness does not depend on viewing direction
- Brightness does depend on direction of illumination
- This is the model most often used in computer vision

Lambert’s Law: \( I_e = k_d N \cdot L I_i \)

\( k_d \) is called albedo

BRDF for Lambertian surface

\( \rho(\theta_i, \phi_i, \theta_e, \phi_e) = k_d \cos \theta_i \)

Specular reflection

For a perfect mirror, light is reflected about \( N \)

\( I_e = \begin{cases} I_i & \text{if } V = R \\ 0 & \text{otherwise} \end{cases} \)

Near-perfect mirrors have a highlight around \( R \)

- common model:

\[ I_e = k_s (V \cdot R)^n I_i \]

Specular reflection

Moving the light source

\( \text{Changing } n_s \)

Phong illumination model

Phong approximation of surface reflectance

- Assume reflectance is modeled by three components
  - Diffuse term
  - Specular term
  - Ambient term (to compensate for inter-reflected light)

\[
I_e = k_d I_a + I_i [k_d (N \cdot L)^\gamma + k_s (V \cdot R)^\gamma s] + I_a k_a \max(0, L_i)
\]

\( L, N, V \) unit vectors

\( I_e \) = outgoing radiance

\( I_i \) = incoming radiance

\( I_a \) = ambient light

\( k_a \) = ambient light reflectance factor

\( \gamma \) = max(1, 0)
BRDF models

Phenomenological
- Phong [75]
- Ward [92]
- Lafortune et al. [97]
- Ashikhmin et al. [00]

Physical
- Cook-Torrance [81]
- Dichromatic [Shafer 85]
- He et al. [91]

Here we’re listing only some well-known examples

Lambertian reflection

\[ I_e = k_d N \cdot L I_i \]

is light source intensity

\[ I_i = \frac{1}{I_e} \]

Lets assume that \( I_i = 1 \)

can achieve this by dividing each pixel in the image by \( I_i \)

\[ I = k_d N \cdot L \]

image intensity at a single point

\[ \text{Lighting direction (same for all points)} \]

\[ \text{Albedo at a point} \]

\[ \text{Surface normal at a point} \]

Shape from shading

Input:
- Single Image

Output:
- 3D shape of the object in the image

Problem is ill-posed: many shapes can give rise to same image.

Common assumptions:
- Lighting is known
- Lambertian reflectance + uniform albedo
- Boundary conditions are known

Suppose \( k_d = 1 \)

\[ I = k_d N \cdot L \]

\[ = N \cdot L \]

\[ = \cos \theta_i \]

You can directly measure angle between normal and light source

- Not quite enough information to compute surface shape \( f(x, y) = \cos \theta = 0.5 \)
  \( \theta = 60^\circ, \phi = ? \)

- But can be if you add some additional info, for example
  - Assume normals along the silhouette are known
  - Constraints on neighboring normals—“integrability”
  - Smoothness
Surface Normal

\[ N = (n_x, n_y, n_z)^T \]

A surface \( z(x,y) \)
A point on the surface: \( (x, y, z(x, y))^T \)
Tangent directions

\[ t_x = (1, 0, z_x)^T \quad t_y = (0, 1, z_y)^T \]

\[ N = \frac{t_x \times t_y}{\|t_x \times t_y\|} = \frac{1}{\sqrt{n_x^2 + n_y^2 + 1}} (-z_x, -z_y, 1)^T \]

Shape from shading

\[ I(x, y) = N \cdot L = \frac{-l_x z_x - l_y z_y + I_0}{\sqrt{z_x^2 + z_y^2 + 1}} \]

Assume that \( L = (0, 0, 1)^T \)
And get that

\[ I(x, y) = N \cdot L = \frac{1}{\sqrt{z_x^2 + z_y^2 + 1}} \]

Two unknowns \( z_x, z_y \)

Shape from shading

\[ I(x, y) = N \cdot L = \frac{1}{\sqrt{z_x^2 + z_y^2 + 1}} \]

But both unknowns come from an integrable surface: \( Z(x, y) \) thus we can use the integrability constraint:

\[ z_{xy} = z_{yx} \]

Shape from shading

\[ I(x, y) = N \cdot L = \frac{1}{\sqrt{z_x^2 + z_y^2 + 1}} \]

\[ \sqrt{z_x^2 + z_y^2 + 1} = \frac{1}{I(x, y)} \quad \sqrt{z_x^2 + z_y^2} = \sqrt{\frac{1}{I(x, y)^2} - 1} \]

\[ \|\nabla Z\| = \frac{1}{\sqrt{I(x, y)^2} - 1} \]

is called Eikonal equation can be solved using variation of Dijkstra’s algorithm

Need to know the extrema points for this
Results

Shape from shading

It is hard to get shape from shading work well in practice.
The assumptions are quite restrictive
But this is recovery of 3D from single 2D image

Fewer assumptions are needed if we have several
images of the same object under different lightings

Photometric stereo

Can write this as a matrix equation:

\[
\begin{bmatrix}
I_1 \\
I_2 \\
I_3
\end{bmatrix}
= k_d
\begin{bmatrix}
L_1^T \\
L_2^T \\
L_3^T
\end{bmatrix}
N
\]

Solving the equations

\[
\begin{bmatrix}
I_1 \\
I_2 \\
I_3
\end{bmatrix}
= \begin{bmatrix}
L_1^T \\
L_2^T \\
L_3^T
\end{bmatrix}
k_d N
\]

\[
G = L^{-1} I
\]

\[
k_d = \|G\|
\]

\[
N = \frac{1}{k_d} G
\]
More than three lights

Get better results by using more lights

\[
\begin{bmatrix}
I_1 \\
\vdots \\
I_n
\end{bmatrix} = \begin{bmatrix}
L_1 \\
\vdots \\
L_n
\end{bmatrix} k_d N
\]

Least squares solution:

\[
\begin{align*}
I &= LG \\
L^TI &= L^T L G \\
G &= (L^T L)^{-1} (L^T I)
\end{align*}
\]

Solve for \( N, k_d \) as before

What's the size of \( L^T L \)?

Example

Recovered albedo

Recovered normal field

Forsyth & Ponce, Sec. 5.4

Color images

The case of RGB images

- get three sets of equations, one per color channel:

\[
\begin{align*}
I_R &= k_{dR} L N \\
I_G &= k_{dG} L N \\
I_B &= k_{dB} L N
\end{align*}
\]

- Simple solution: first solve for \( N \) using one channel or grayscale
- Then substitute known \( N \) into above equations to get \( k_d \)s

\[
k_u = \frac{\sum_i I_i L_i N_i^T}{\sum_i (L_i N_i^T)^2}
\]

Where do we get the lighting directions?
Capture lighting variation
Illuminate subject from many incident directions

Example images:

Computing light source directions
Trick: place a chrome sphere in the scene

• the location of the highlight tells you where the light source is

Recall the rule for specular reflection
For a perfect mirror, light is reflected about N

\[
R_v = \begin{cases} 
R_v & \text{if } V = R \\
0 & \text{otherwise} 
\end{cases}
\]

We see a highlight when \( V = R \)
• then \( L \) is given as follows:
\[
L = 2(N \cdot R)N - R
\]
Computing the light source direction

Chrome sphere that has a highlight at position $h$ in the image

Can compute $\theta$ (and hence $N$) from this figure

Now just reflect $V$ about $N$ to obtain $L$

Depth from normals

What we have

What we want

Depth from normals

Get a similar equation for $V_2$

- Each normal gives us two linear constraints on $z$
- Compute $z$ values by solving a matrix equation
- On the boundary we have only one constraint.

Trick for handling shadows

Weight each equation by the pixel brightness:

$$I_i(I_i) = I_i[k_dN \cdot L_i]$$

Gives weighted least-squares matrix equation:

$$\begin{bmatrix} I_1^2 & \cdots & I_n^2 \end{bmatrix} = k_dN^T \begin{bmatrix} I_1L_1 & \cdots & I_nL_n \end{bmatrix}$$

Solve for $N$, $k_d$ as before
Limitations

- doesn’t work for shiny things, semi-translucent things
- shadows, inter-reflections are difficult
- Single light source illumination
- camera and lights have to be distant
- calibration requirements
  - measure light source directions, intensities
  - camera response function

Newer work addresses some of these issues

Some pointers for further reading:

- Zickler, Belhumeur, and Kriegman, “
- Hertzmann & Seitz, Example-Based Photometric Stereo: Shape Reconstruction with General, Varying BRDFs,” IEEE Trans. PAMI 2005

Hertzmann & Seitz,
*Example-Based Photometric Stereo: Shape Reconstruction with General, Varying BRDFs.*” IEEE Trans. PAMI 2005
Shiny things

“Orientation consistency”