CS5670 : Computer Vision
Reflectance & Photometric stereo
Reading

• Szeliski 2nd Edition: Chapter 2.2 & 13.1
Announcements

• Project 3 (Panorama) artifact due tonight at 8pm
• Project 4 (Stereo) released today due Friday, March 31, at 8pm
  – To be done in groups of 2
Roadmap for the rest of the course

• The next three lectures will finish up geometry and image formation
  – Next up (after Spring Break): deep learning, image recognition, neural radiance fields, image synthesis

• Coming up
  – Reflectance and Photometric Stereo (today)
  – Two-view geometry
  – Multi-view geometry
Project 4 Demo
Last time: Light & Perception

- Now: Reflectance
Light sources

• Basic types
  – point source
  – directional source
    • a point source that is infinitely far away
  – 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?
Modeling Image Formation

We need to reason about:

• How light interacts with the scene
• How a pixel value is related to light energy in the world

Track a “ray” of light all the way from light source to the sensor
Directional Lighting

• Key property: all rays are parallel
• Equivalent to an infinitely distant point source
Lambertian Reflectance

\[ I = N \cdot L \]

- Image intensity
- Surface normal
- Light direction

\[ \propto \cos(\text{angle between N and L}) \]
Materials - Three Forms

- Ideal diffuse (Lambertian)
- Ideal specular
- Directional diffuse
Reflectance — Three Forms

- Ideal diffuse (Lambertian)
- Ideal specular
- Directional diffuse

© Kavita Bala, Computer Science, Cornell University
Ideal Diffuse Reflection

• Characteristic of multiple scattering materials
• An idealization but reasonable for matte surfaces
Lambertian Reflectance

1. Reflected energy is proportional to cosine of angle between L and N \((\text{incoming})\)

2. Measured intensity is viewpoint-independent \((\text{outgoing})\)
Lambertian Reflectance: Incoming

• Reflected energy is proportional to cosine of angle between $L$ and $N$
Lambertian Reflectance: Incoming

- Reflected energy is proportional to cosine of angle between $L$ and $N$
Lambertian Reflectance: Incoming

- Reflected energy is proportional to cosine of angle between L and N

Light hitting surface is proportional to the cosine
Lambertian appearance is view-independent

- Number of photons reflected to a given angle $\theta$ is proportional to $\cos(\theta)$

\[ B = B_0 \cos(\theta) \]
Lambertian appearance is view-independent

- Number of photons reflected to a given angle $\theta$ is proportional to $\cos(\theta)$
- But appearance is the same from every angle due to larger pixel footprint at larger angles

Lambert's cosine law: $B = B_0 \cos(\theta)$
Lambertian appearance is view-independent

- Number of photons reflected to a given angle $\theta$ is proportional to $\cos(\theta)$

\[
B = B_0 \cos(\theta)
\]

- But appearance is the same from every angle due to larger pixel footprint at larger angles

\[
A_\theta \propto A_0 \frac{1}{\cos \theta}
\]

Lambert's cosine law: $B = B_0 \cos(\theta)$
Lambertian appearance is view-independent

- Number of photons reflected to a given angle $\theta$ is proportional to $\cos(\theta)$
  \[ B = B_0 \cos(\theta) \]

- But appearance is the same from every angle due to larger pixel footprint at larger angles
  \[ A_\theta \propto A_0 \frac{1}{\cos \theta} \]

Lambert's cosine law: $B = B_0 \cos(\theta)$

Radiance (what eye sees) $\propto B_0 A_0 \cos(\theta) \frac{1}{\cos(\theta)}$
1. Diffuse albedo: what fraction of incoming light is reflected?
   • Introduce scale factor $k_d$
2. Light intensity: how much light is arriving?
   • Compensate with camera exposure (global scale factor)
3. Camera response function
   • Assume pixel value is linearly proportional to incoming energy
     (perform radiometric calibration if not)
Albedo

Object can have varying albedo and albedo varies with wavelength

Source: https://en.wikipedia.org/wiki/Albedo
Can we determine shape from lighting?

- Are these spheres?
  - Or just flat discs painted with varying albedo?
A Single Image: Shape from shading

Suppose (for now) \( k_d = 1 \)

\[
I = k_d N \cdot L = N \cdot L = \cos \theta_i
\]

You can directly measure angle between normal and light source
- Not quite enough information to compute surface shape
- But can be if you add some additional info, for example
  - assume a few of the normals are known (e.g., along silhouette)
  - constraints on neighboring normals—"integrability"
  - smoothness
- Hard to get it to work well in practice
  - plus, how many real objects have constant albedo?
  - But, deep learning can help
Let’s take more than one photo!
Photometric stereo

Can write this as a matrix equation:

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

$$I_1 = k_d N \cdot L_1$$

$$I_2 = k_d N \cdot L_2$$

$$I_3 = k_d N \cdot L_3$$
Solving the equations

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

\[
G = L^{-1}I \\
k_d = \|G\| \\
N = \frac{1}{k_d}G
\]

Solve one such linear system **per pixel** to solve for that pixel’s surface normal
More than three lights

Can get better results by using more than 3 lights

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

Least squares solution:

\[
I = LG
\]

\[
LTI = LTLG
\]

\[
G = (LTL)^{-1}(LTI)
\]

Solve for N, \(k_d\) as before

What’s the size of \(LTL\)?
Computing light source directions

Trick: place a chrome sphere in the scene

– the location of the highlight tells you where the light source is
Example

Input views

Recovered albedo

Recovered normal field

Forsyth & Ponce, Sec. 5.4

Example

Recovered albedo

Recovered normal field

Forsyth & Ponce, Sec. 5.4
Depth from normals

- Solving the linear system per-pixel gives us an estimated surface normal for each pixel

- How can we compute depth from normals?
  - Normals are like the “derivative” of the true depth
Normal Integration

• Integrating a set of derivatives is easy in 1D
  • (similar to Euler’s method from diff. eq. class)

• Could integrate normals in each column / row separately
  • Wouldn’t give a good surface

• Instead, we formulate as a linear system and solve for depths that best agree with the surface normals
Depth from normals

Get a similar equation for $\mathbf{V}_2$

- Each normal gives us two linear constraints on $z$
- compute $z$ values by solving a matrix equation

\[
\begin{align*}
V_1 &= (x + 1, y, z_{x+1,y}) - (x, y, z_{xy}) \\
&= (1, 0, z_{x+1,y} - z_{xy})
\end{align*}
\]

\[
\begin{align*}
0 &= \mathbf{N} \cdot V_1 \\
&= (n_x, n_y, n_z) \cdot (1, 0, z_{x+1,y} - z_{xy}) \\
&= n_x + n_z(z_{x+1,y} - z_{xy})
\end{align*}
\]
Results

from Athos Georghiades
Results
Extension

- Photometric Stereo from Colored Lighting

---

**Video Normals from Colored Lights**
Gabriel J. Brostow, Carlos Hernández, George Vogiatzis, Björn Stenger, Roberto Cipolla

---

*Fig. 2.* Applying the original algorithm to a face with white makeup. Top: example input frames from video of an actor smiling and grimacing. Bottom: the resulting integrated surfaces.
Questions?
For now, ignore specular reflection
And Refraction...
And Interreflections...
And Subsurface Scattering...
Limitations

Bigger problems
– doesn’t work for shiny things, semi-translucent things
– shadows, inter-reflections

Smaller problems
– 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:
Johnson and Adelson, 2009
Johnson and Adelson, 2009
Lights, camera, action

Sensor

Lights

Camera
Figure 7: Comparison with the high-resolution result from the original retrographic sensor. (a) Rendering of the high-resolution $20 bill example from the original retrographic sensor with a close-up view. (b) Rendering of the captured geometry using our method.
Figure 9: Example geometry measured with the bench and portable configurations. Outer image: rendering under direct lighting. Inset: macro photograph of original sample. Scale shown in upper left. Color images are shown for context and are to similar, but not exact scale.
Sensing Surfaces with GelSight

https://www.youtube.com/watch?v=S7gXih4XS7A
InverseRenderNet: Learning single image inverse rendering

Ye Yu and William A. P. Smith
Department of Computer Science, University of York, UK
{yy1571, william.smith}@york.ac.uk

Figure 1: From a single image (col. 1), we estimate albedo and normal maps and illumination (col. 2-4); comparison multi-view stereo result from several hundred images (col. 5); re-rendering of our shape with frontal/estimated lighting (col. 6-7).