Photometric Stereo

Readings
• Optional: Woodham’s original photometric stereo paper

Merle Norman Cosmetics, Los Angeles

Diffuse reflection

\[ I = k_d \mathbf{N} \cdot \mathbf{L} \]

image intensity of \( P \)
\[ I = k_d \mathbf{N} \cdot \mathbf{L} \]

Simplifying assumptions
• \( I = R_e \): camera response function \( f \) is the identity function:
  – can always achieve this in practice by solving for \( f \) and applying \( f^{-1} \) to each pixel in the image
• \( R_i = 1 \): light source intensity is 1
  – can achieve this by dividing each pixel in the image by \( R_i \)

Shape from shading

Suppose \( k_d = 1 \)
\[ I = k_d \mathbf{N} \cdot \mathbf{L} \]
\[ = \mathbf{N} \cdot \mathbf{L} \]
\[ = \cos \theta_i \]

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?

Photometric stereo

Can write this as a matrix equation:
\[ \begin{bmatrix} I_1 & I_2 & I_3 \end{bmatrix} = k_d \mathbf{N}^T \begin{bmatrix} \mathbf{L}_1 & \mathbf{L}_2 & \mathbf{L}_3 \end{bmatrix} \]
Solving the equations

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

\[
I_{1 \times 3} \quad \quad \quad \quad G_{1 \times 3} \quad \quad \quad \quad L_{3 \times 3}
\]

\[
G = IL^{-1}
\]

\[
k_d = ||G||
\]

\[
N = \frac{1}{k_d} G
\]

More than three lights

Get better results by using more lights

\[
\begin{bmatrix}
I_1 & \ldots & I_n
\end{bmatrix}
= k_d N^T
\begin{bmatrix}
L_1 & \ldots & L_n
\end{bmatrix}
\]

Least squares solution:

\[
I = GL
\]

\[
IL^T = GLL^T
\]

\[
G = (IL^T)(LL^T)^{-1}
\]

Solve for N, k_d as before

What's the size of LL^T?

Color images

The case of RGB images

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

\[
I_R = k_{dR}N^TL
\]

\[
I_G = k_{dG}N^T\ell
\]

\[
I_B = k_{dB}N^T\ell
\]

• Simple solution: first solve for N using one channel

• Then substitute known N into above equations to get k_d s:

\[
I_R = k_{dR}J
\]

\[
J \cdot I_R = k_{dR}J \cdot J
\]

\[
k_{dR} = \frac{J \cdot I_R}{J \cdot J}
\]

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 \( \mathbf{N} \)

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

We see a highlight when \( \mathbf{V} = \mathbf{R} \)

- then \( \mathbf{L} \) is given as follows:

\[
\mathbf{L} = 2(\mathbf{N} \cdot \mathbf{R})\mathbf{N} - \mathbf{R}
\]

Computing the light source direction

Chrome sphere that has a highlight at position \( \mathbf{h} \) in the image

Can compute \( \theta \) (and hence \( \mathbf{N} \)) from this figure

Now just reflect \( \mathbf{V} \) about \( \mathbf{N} \) to obtain \( \mathbf{L} \)

Computing the light source direction

Chrome sphere that has a highlight at position \( \mathbf{h} \) in the image

Can compute \( \mathbf{N} \) by studying this figure

- Hints:
  - use this equation: \( \| \mathbf{H} - \mathbf{C} \| = r \)
  - can measure \( \mathbf{c}, \mathbf{h}, \) and \( r \) in the image
  - can choose \( c_z = 0 \)

Depth from normals

Orthographic projection

\[
V_1 = (x + 1, y, z_{x+1,y}) - (x, y, z_{xy}) = (1, 0, z_{x+1,y} - z_{xy})
\]

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

Get a similar equation for \( V_2 \)

- Each normal gives us two linear constraints on \( z \)
- compute \( z \) values by solving a matrix equation
Results…

Limitations

Big 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

Trick for handling shadows

Weight each equation by the pixel brightness:

\[ I_i(I_i) = I_i[k_d N \cdot L_i] \]

Gives weighted least-squares matrix equation:

\[
\begin{bmatrix}
I_1^2 & \cdots & I_n^2
\end{bmatrix}
= k_d N^T
\begin{bmatrix}
I_1 L_1 & \cdots & I_n L_n
\end{bmatrix}
\]

Solve for \( N, k_d \) as before

from Athos Georghiades
http://cvc.yale.edu/people/Athos.html