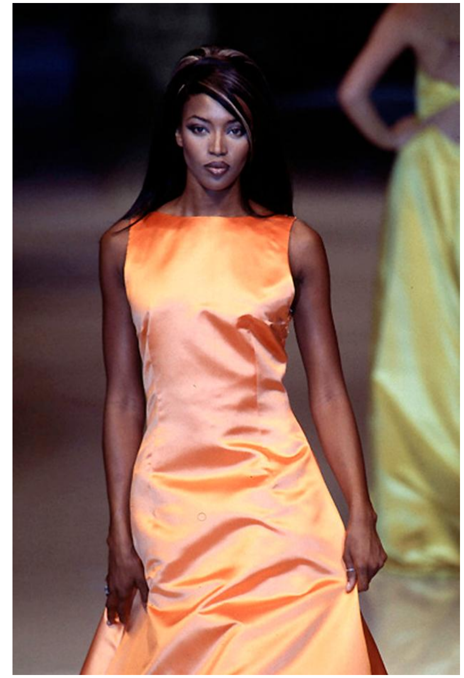
CS4670: Computer Vision

Noah Snavely

Lecture 32: Photometric stereo, Part 2

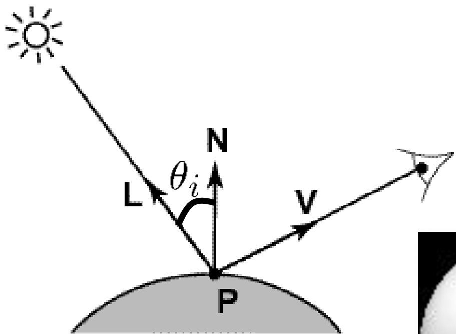
BRDF’s can be incredibly complicated…








Shape from shading



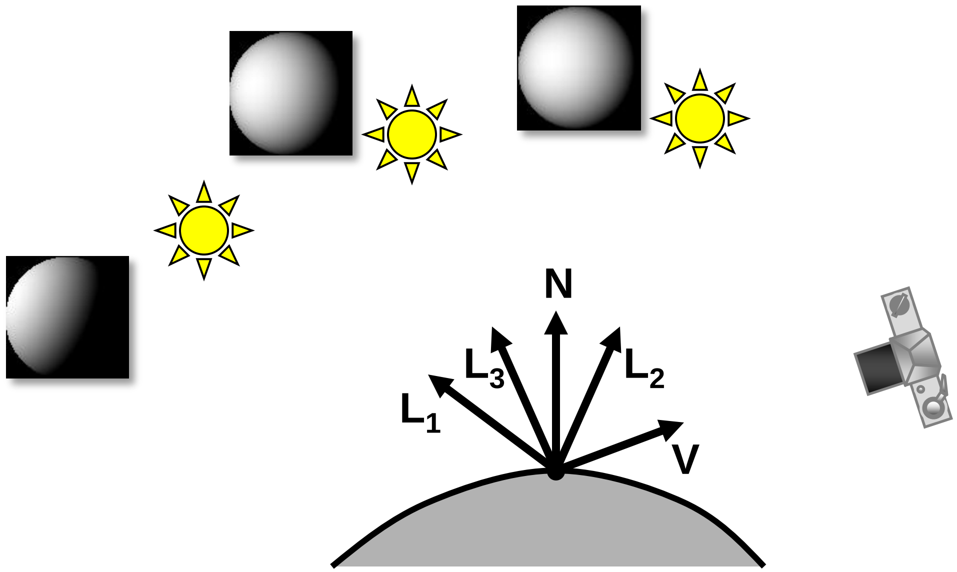
Suppose $k_d = 1$

$$\begin{aligned} I &= k_d \mathbf{N} \cdot \mathbf{L} \\ &= \mathbf{N} \cdot \mathbf{L} \\ &= cos\ \theta_i \end{aligned}$$

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



$$I_1 = k_d \mathbf{N} \cdot \mathbf{L_1}$$
$$I_2 = k_d \mathbf{N} \cdot \mathbf{L_2}$$
$$I_3 = k_d \mathbf{N} \cdot \mathbf{L_3}$$

Can write this as a matrix equation:

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

Solving the equations

$$\underbrace{\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix}}_{\substack{\mathbf{I} \\ 3 \times 1}} = \underbrace{\begin{bmatrix} \mathbf{L_1}^T \\ \mathbf{L_2}^T \\ \mathbf{L_3}^T \end{bmatrix}}_{\substack{\mathbf{L} \\ 3 \times 3}} \underbrace{k_d \mathbf{N}}_{\substack{\mathbf{G} \\ 3 \times 1}}$$

$$\mathbf{G} = \mathbf{L}^{-1}\mathbf{I}$$

$$k_d = \|\mathbf{G}\|$$

$$\mathbf{N} = \frac{1}{k_d}\mathbf{G}$$

More than three lights

Get better results by using more lights

$$\begin{bmatrix} I_1 \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} \mathbf{L_1} \\ \vdots \\ \mathbf{L_n} \end{bmatrix} k_d \mathbf{N}$$

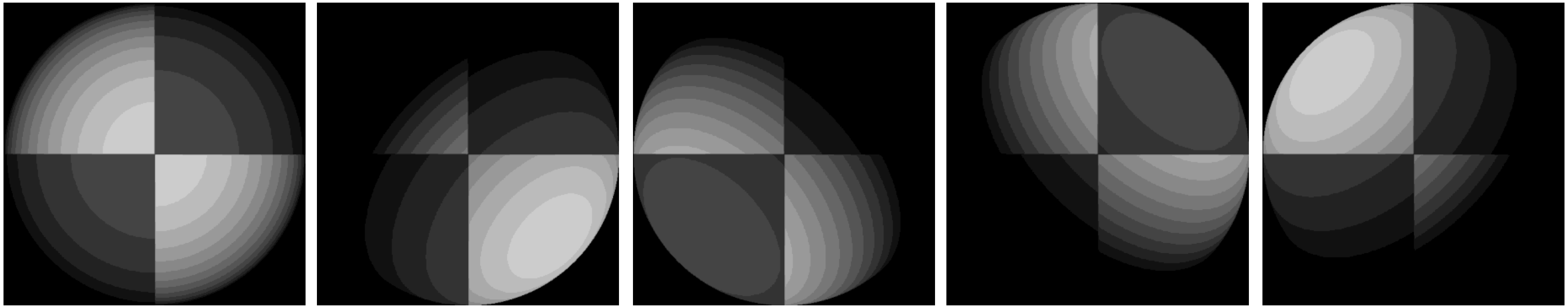
Least squares solution:

$$\begin{aligned} \mathbf{I} &= \mathbf{L}\mathbf{G} \\ \mathbf{L}^{\mathbf{T}}\mathbf{I} &= \mathbf{L}^{\mathbf{T}}\mathbf{L}\mathbf{G} \\ \mathbf{G} &= (\mathbf{L}^{\mathbf{T}}\mathbf{L})^{-1}(\mathbf{L}^{\mathbf{T}}\mathbf{I}) \end{aligned}$$

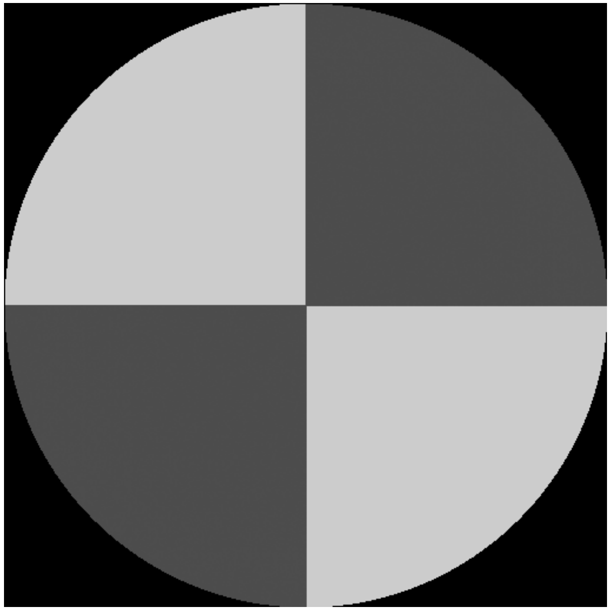
Solve for N, k_d as before

What's the size of $\mathbf{L^T L}$?

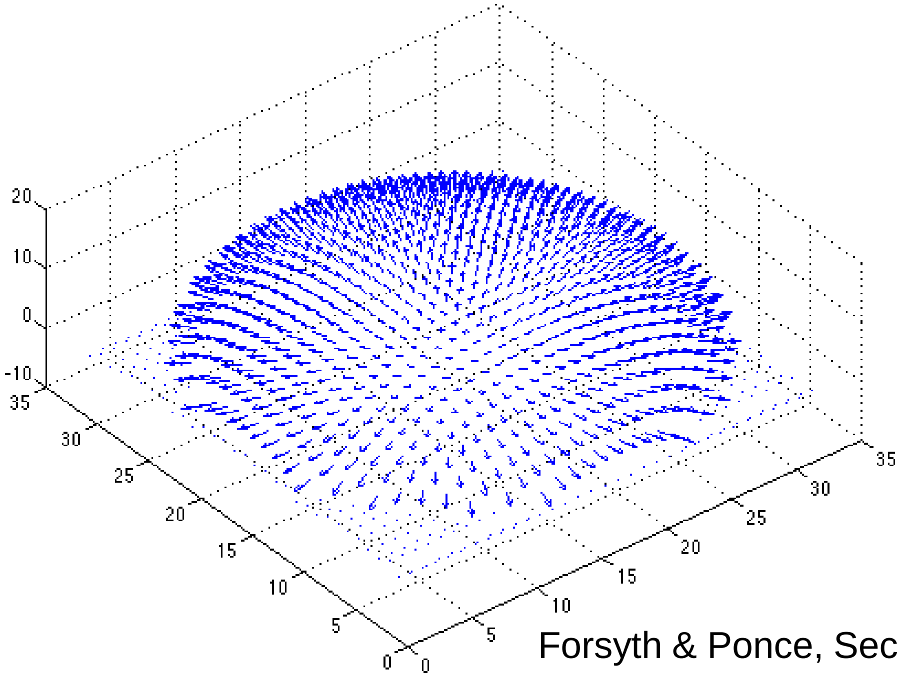
Example



Recovered albedo



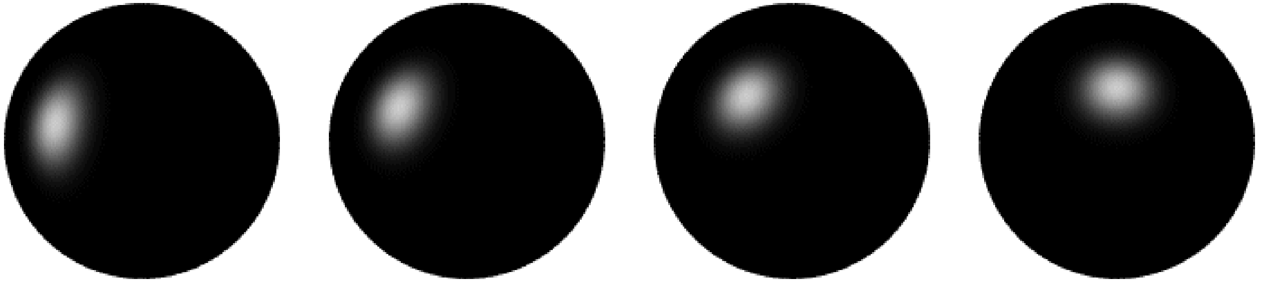
Recovered normal field



Forsyth & Ponce, Sec. 5.4

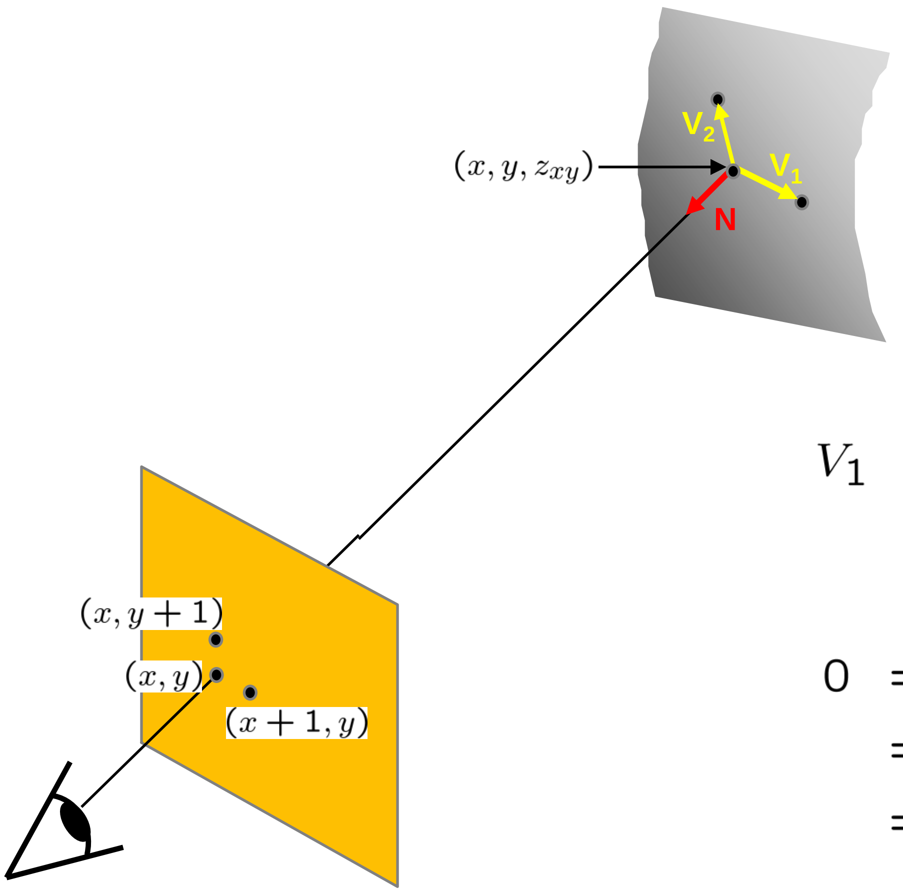
Computing light source directions

Trick: place a chrome sphere in the scene



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

Depth from normals



$$V_1 = (x+1, y, z_{x+1,y}) - (x, y, z_{xy}) = (1, 0, z_{x+1,y} - z_{xy})$$

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

Get a similar equation for $\mathbf{V_2}$

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

Example

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

Newer work addresses some of these issues

Some pointers for further reading:

- Zickler, Belhumeur, and Kriegman, *"Helmholtz Stereopsis: Exploiting Reciprocity for Surface Reconstruction."* IJCV, Vol. 49 No. 2/3, pp 215-227.
- Hertzmann & Seitz, "*Example-Based Photometric Stereo: Shape Reconstruction with General, Varying BRDFs*." IEEE Trans. PAMI 2005

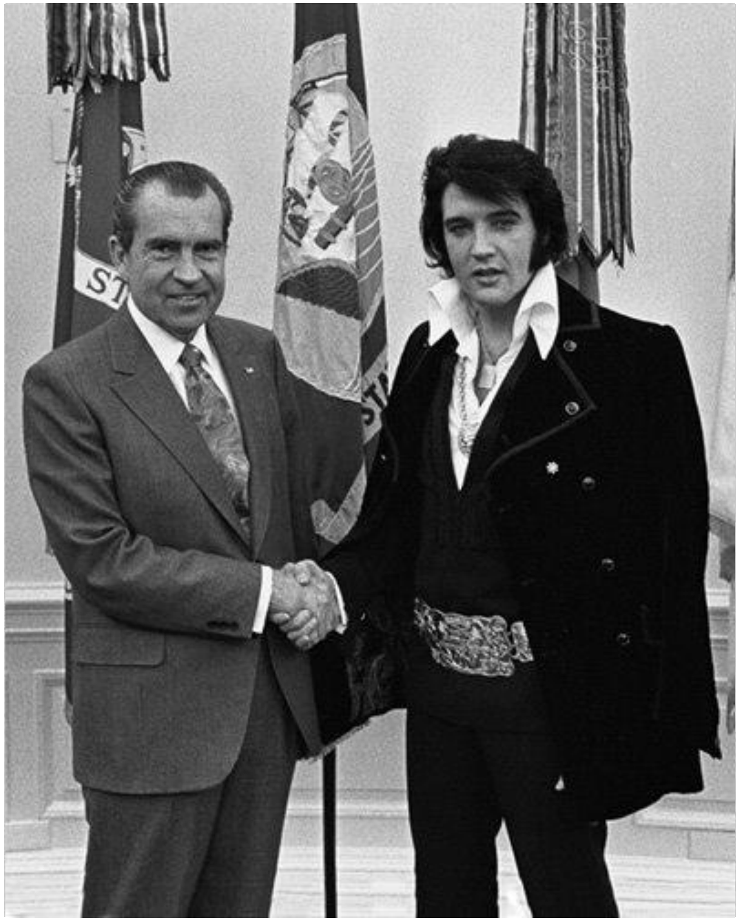
Finding the direction of the light source

P. Nillius and J.-O. Eklundh, “Automatic estimation of the projected light source direction,” CVPR 2001

Application: Detecting composite photos

Which is the real photo?

Fake photo



Real photo

Questions?