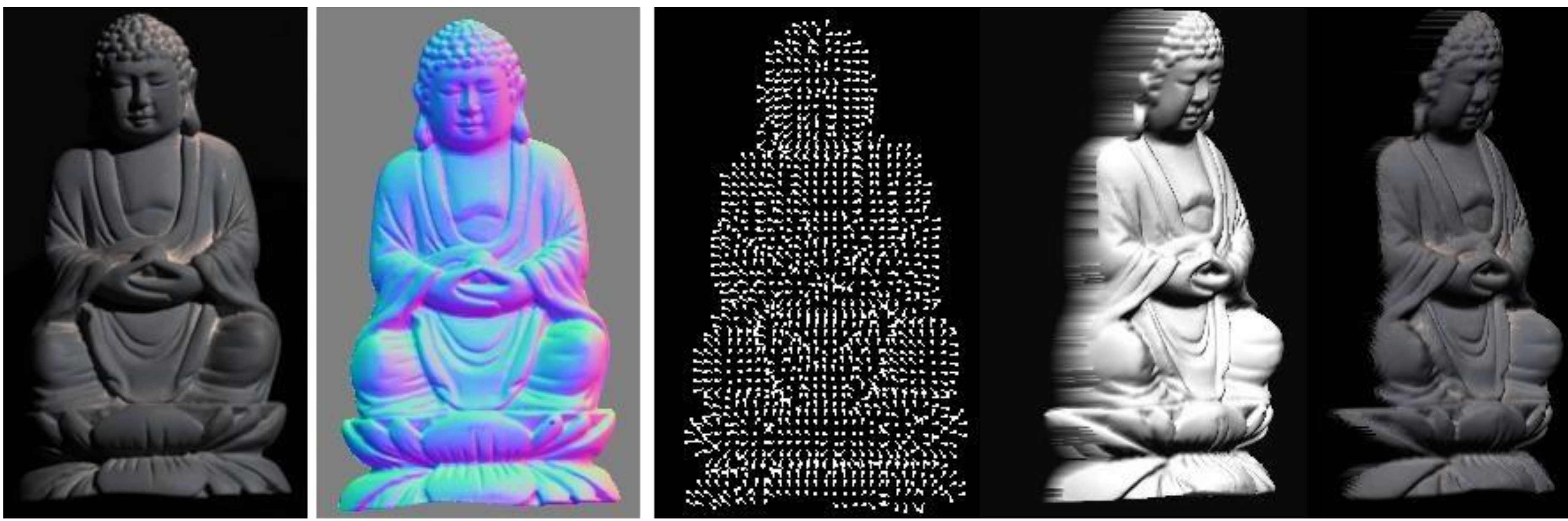
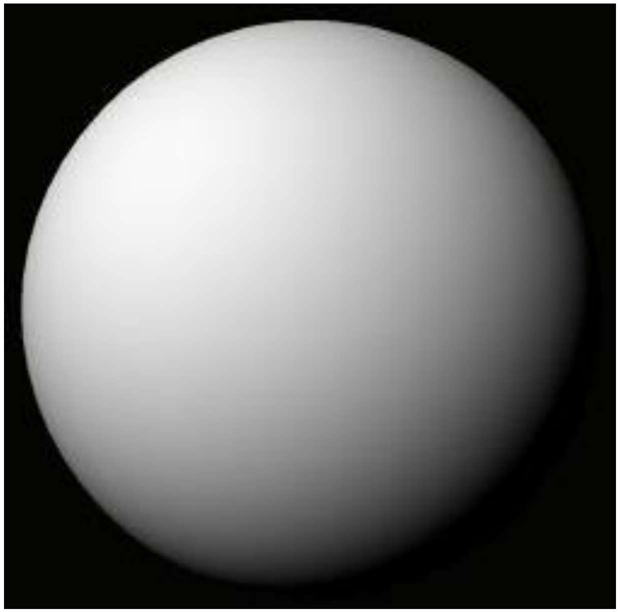
CS5670 : Computer Vision

Noah Snavely

Photometric stereo



A Single Image: Shape from Shading



$$I = k_d \mathbf{N} \cdot \mathbf{L}$$

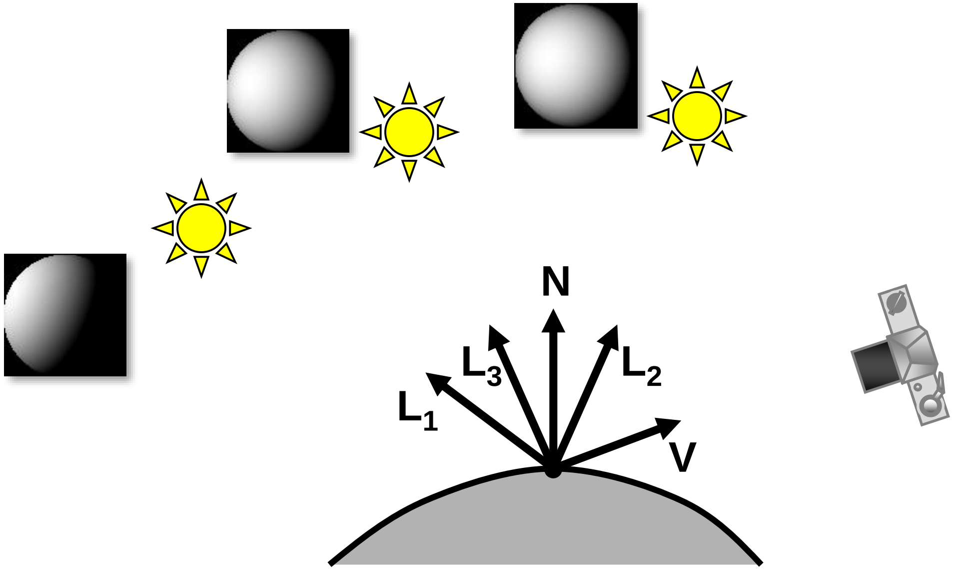
Assume k_d is 1 for now.

What can we measure from one image?

- $\cos^{-1}(I)$ is the angle between N and L
- Add assumptions:
  - Constant albedo
  - A few known normals (e.g. silhouettes)
  - Smoothness of normals

In practice, SFS doesn’t work very well:
assumptions are too restrictive,
too much ambiguity in nontrivial scenes.

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:

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

Solving the equations

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

$$\underbrace{\qquad}_{\substack{\mathbf{I} \\ 3 \times 1}} \qquad \underbrace{\qquad}_{\substack{\mathbf{L} \\ 3 \times 3}} \quad \underbrace{\qquad}_{\substack{\mathbf{G} \\ 3 \times 1}}$$

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

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

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

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:

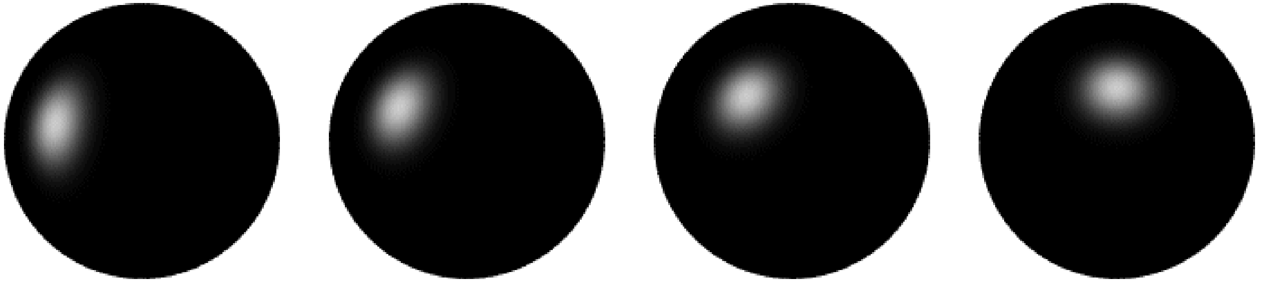
$$\begin{aligned} \mathbf{I} &= \mathbf{L}\mathbf{G} \\ \mathbf{L^T}\mathbf{I} &= \mathbf{L^T}\mathbf{L}\mathbf{G} \\ \mathbf{G} &= (\mathbf{L^T}\mathbf{L})^{-1}(\mathbf{L^T}\mathbf{I}) \end{aligned}$$

Solve for N, k_d as before

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

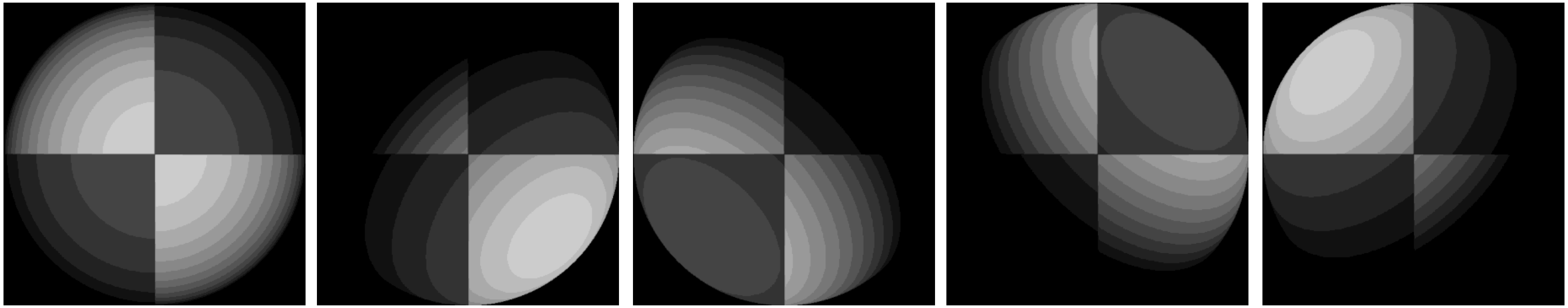
Computing light source directions

Trick: place a chrome sphere in the scene

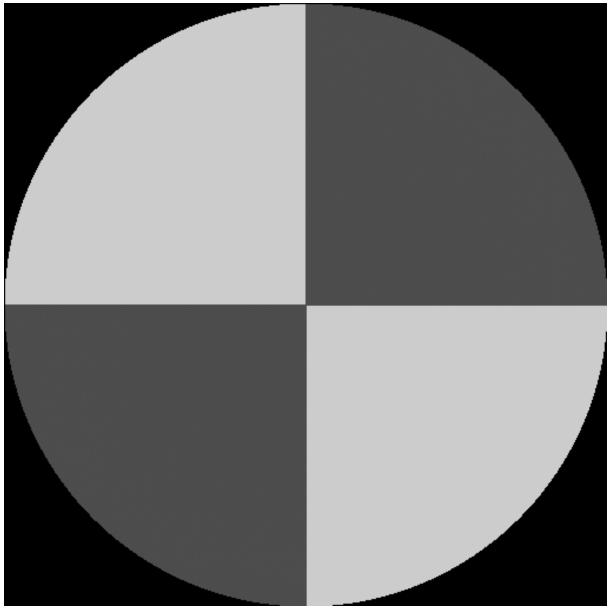


- the location of the highlight tells you where the light source is

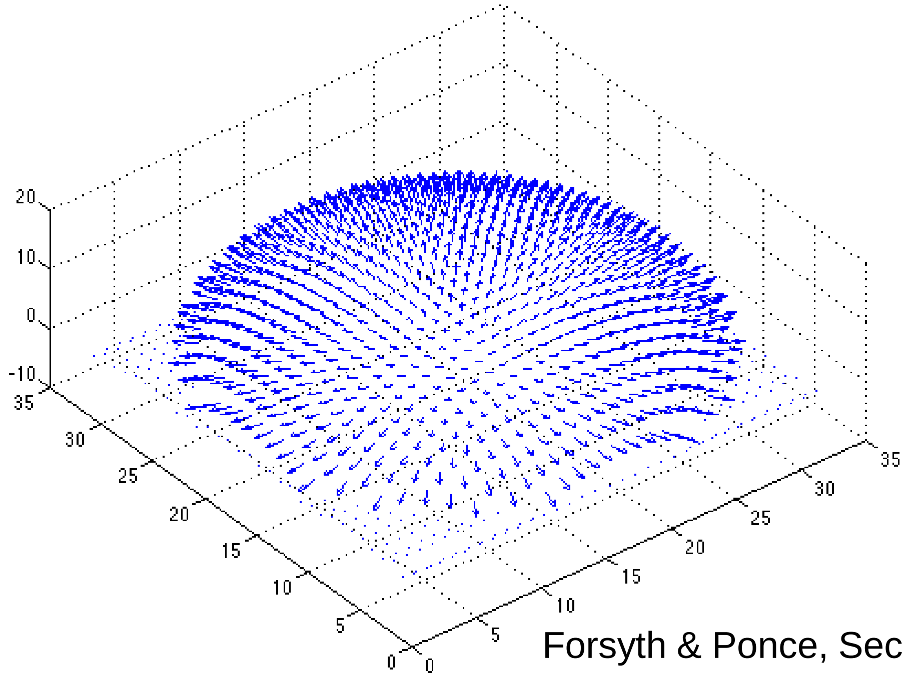
Example



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

Input photo

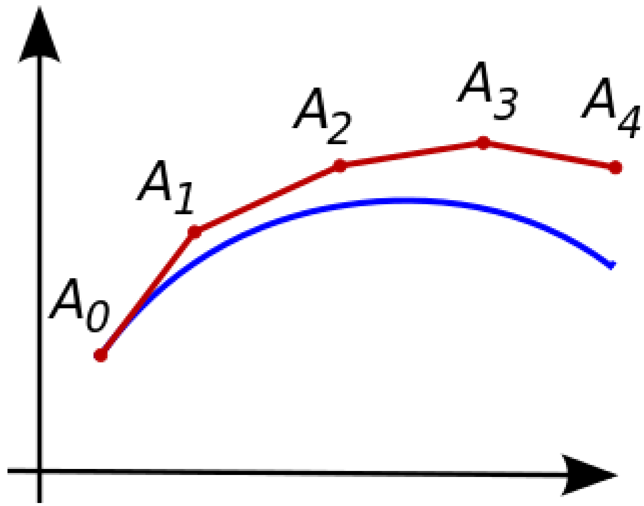
Estimated normals

Estimated normals (needle diagram)

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

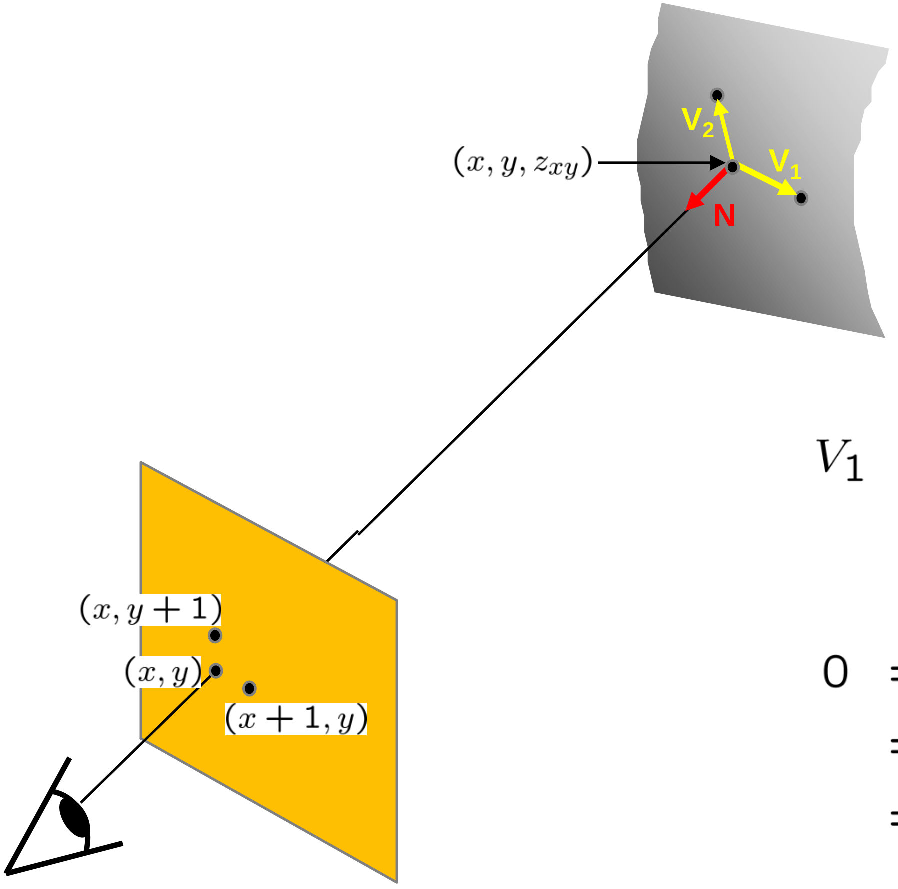
Normal Integration

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



- Could just integrate normals in each column / row separately
- Instead, we formulate as a linear system and solve for depths that *best agree with the surface normals*

Depth from normals



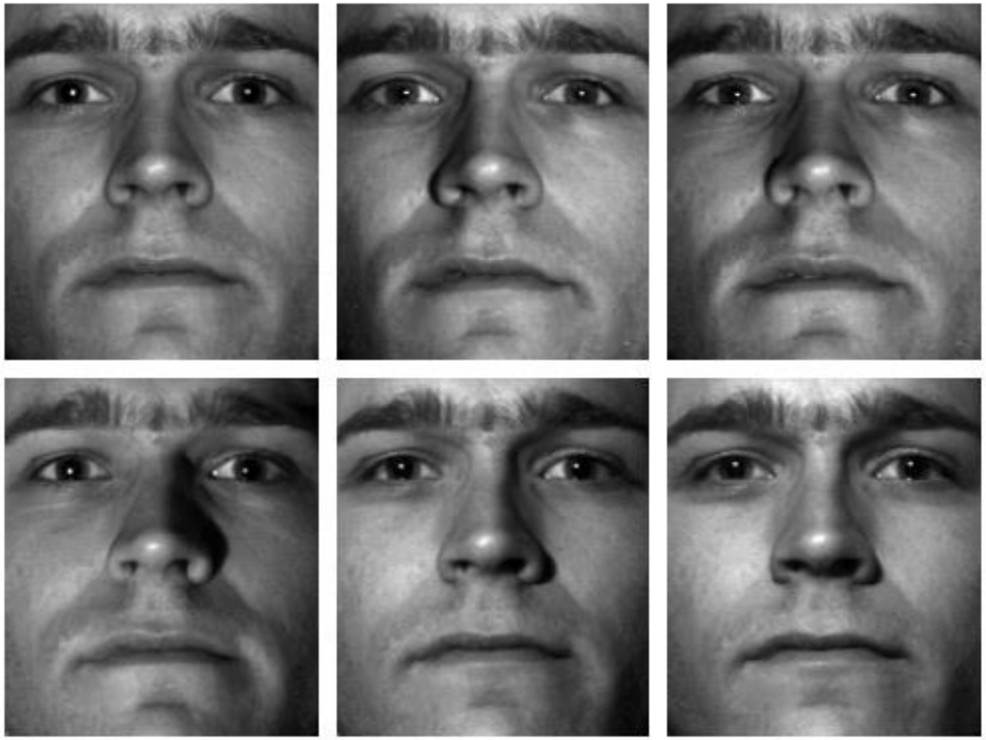
$$
\begin{aligned}
V_1 &= (x+1, y, z_{x+1,y}) - (x, y, z_{xy}) \\
&= (1, 0, z_{x+1,y} - z_{xy})
\end{aligned}
$$

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

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

Results

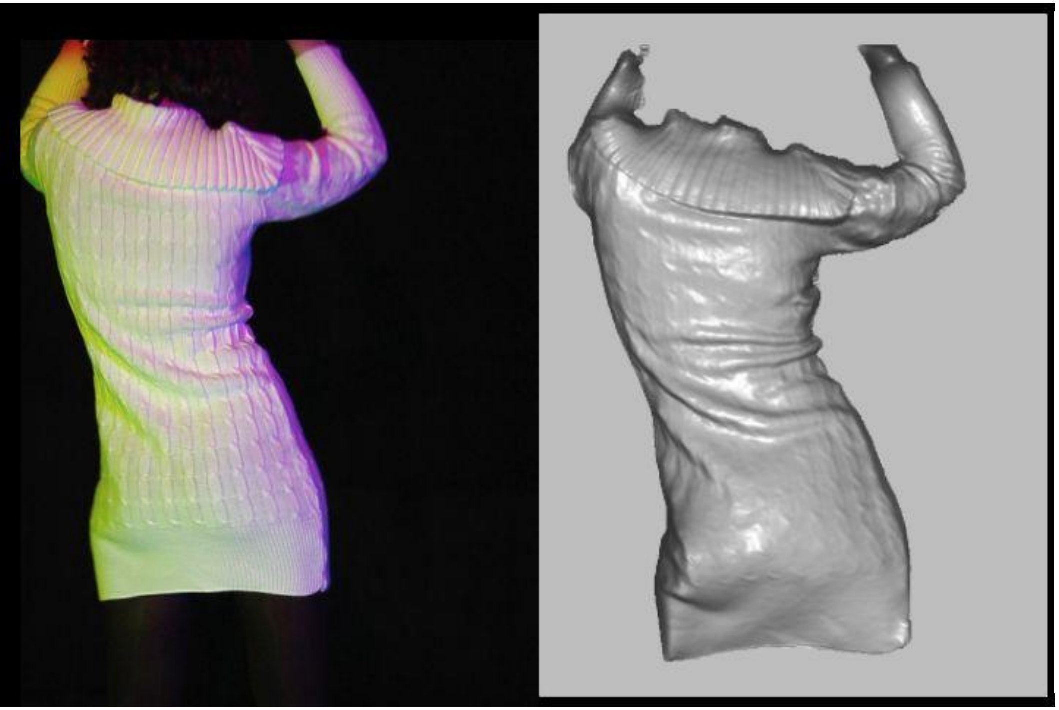


from Athos Georghiades

Example

Extension

Photometric Stereo from Colored Lighting



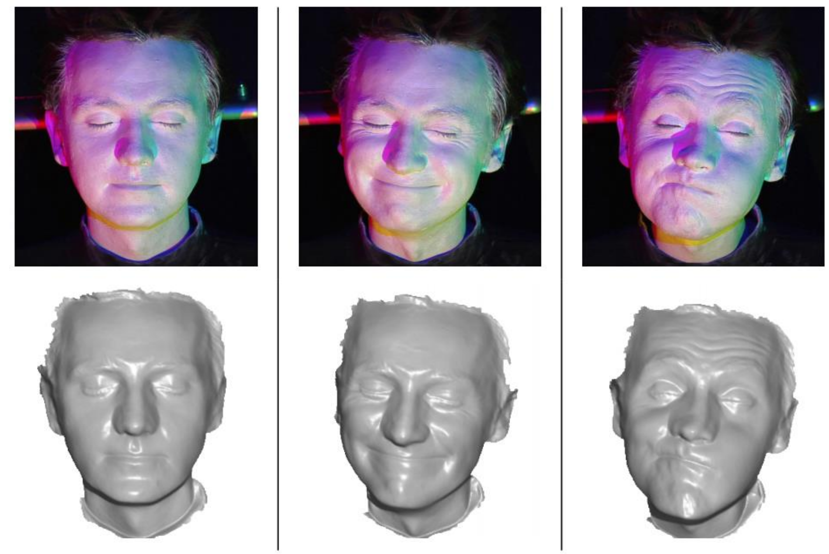


Fig. 2. **Applying the original algorithm to a face with white makeup.** Top: example input frames from video of an actor smiling and grimacing. Bottom: the resulting integrated surfaces.

Video Normals from Colored Lights
Gabriel J. Brostow, Carlos Hernández, George Vogiatzis, Björn Stenger, Roberto Cipolla
IEEE TPAMI, Vol. 33, No. 10, pages 2104-2114, October 2011.