Photometric Stereo

October 8, 2013
Dr. Grant Schindler

schindler@gatech.edu

Multiple Images: Different Lighting

Multiple Images: Different Lighting

Multiple Images: Different Lighting

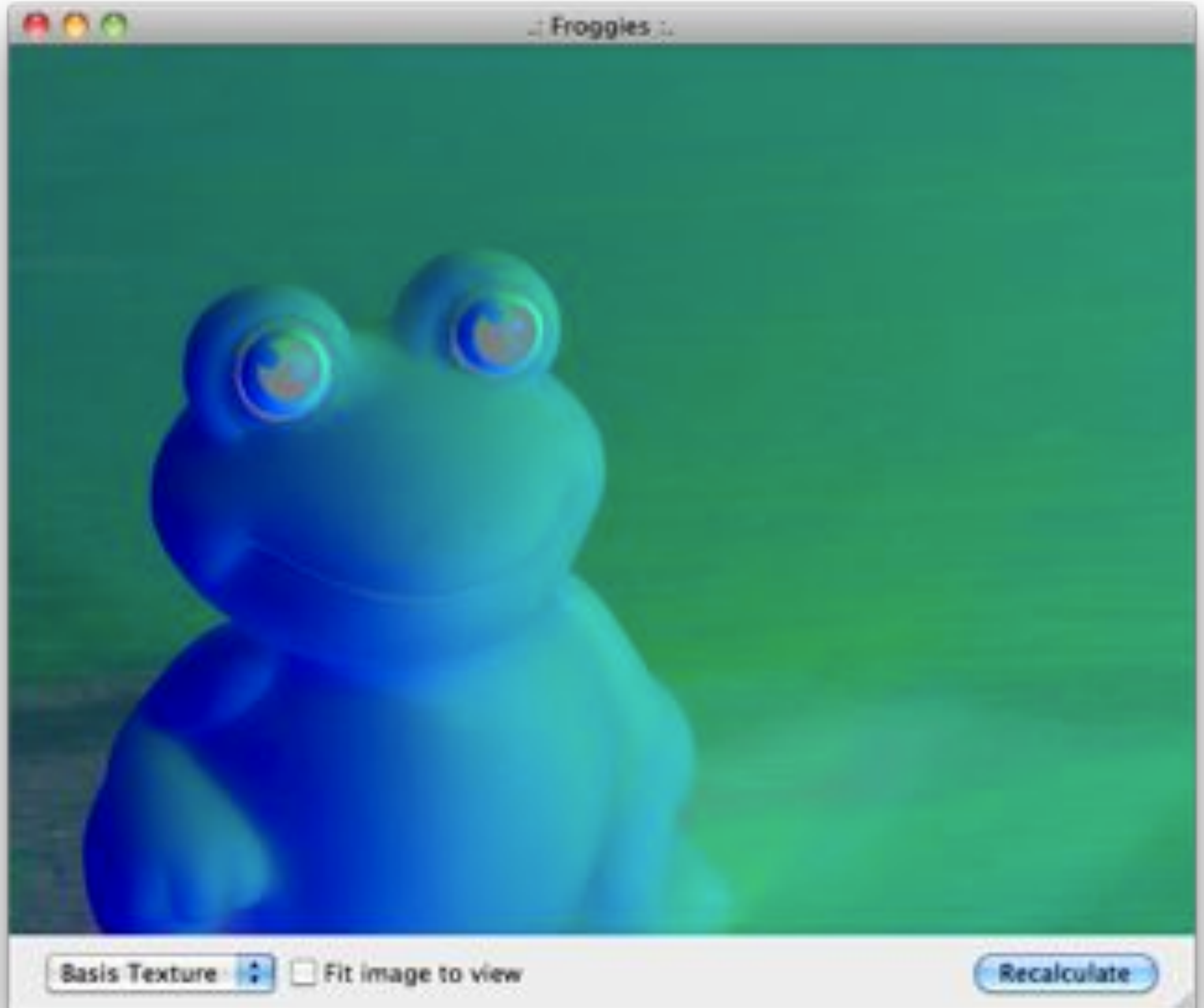
Relighting the Scene

Credit: Alex Powell

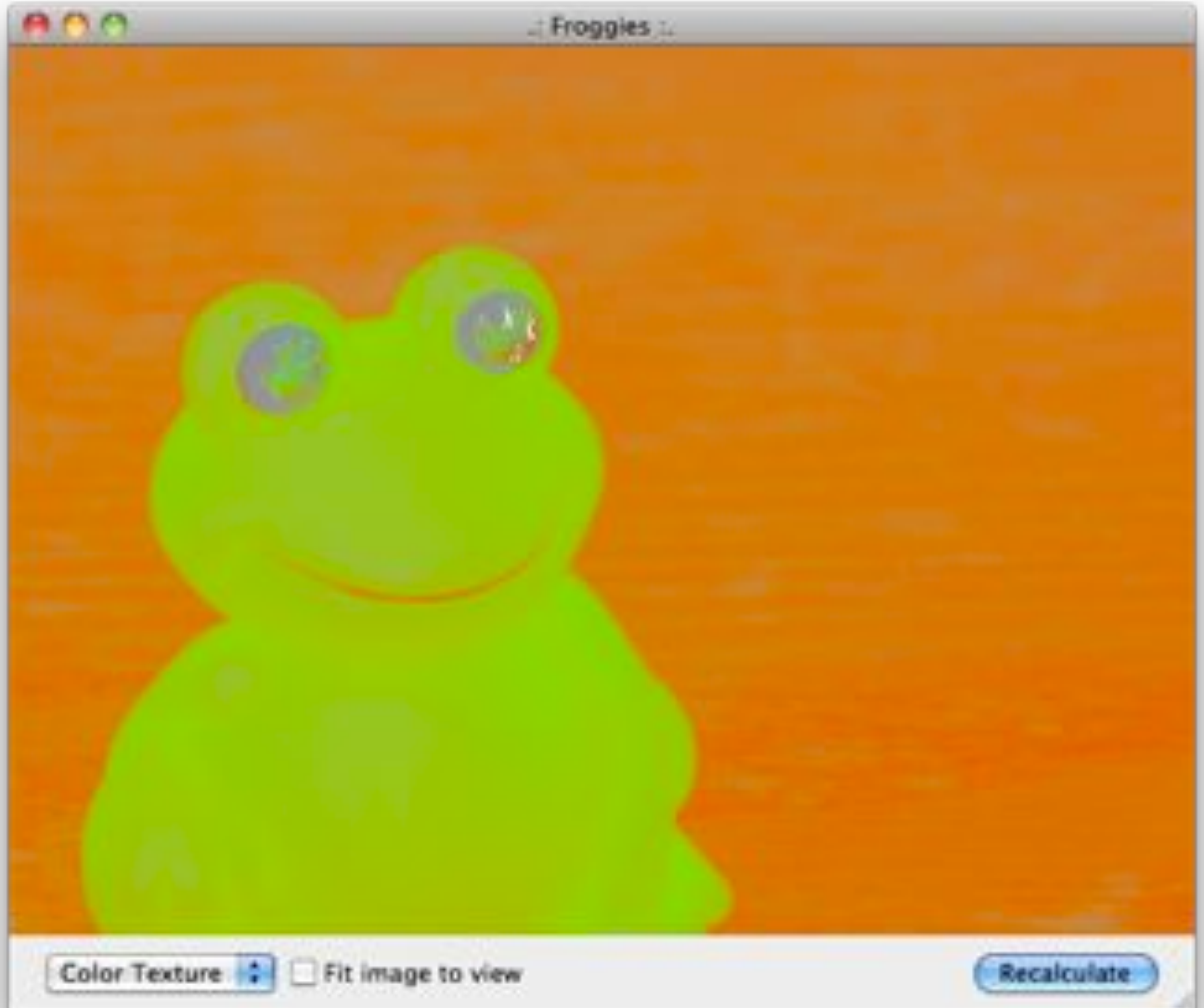
Credit: Alex Powell

Credit: Alex Powell

Credit: Alex Powell



Credit: Alex Powell

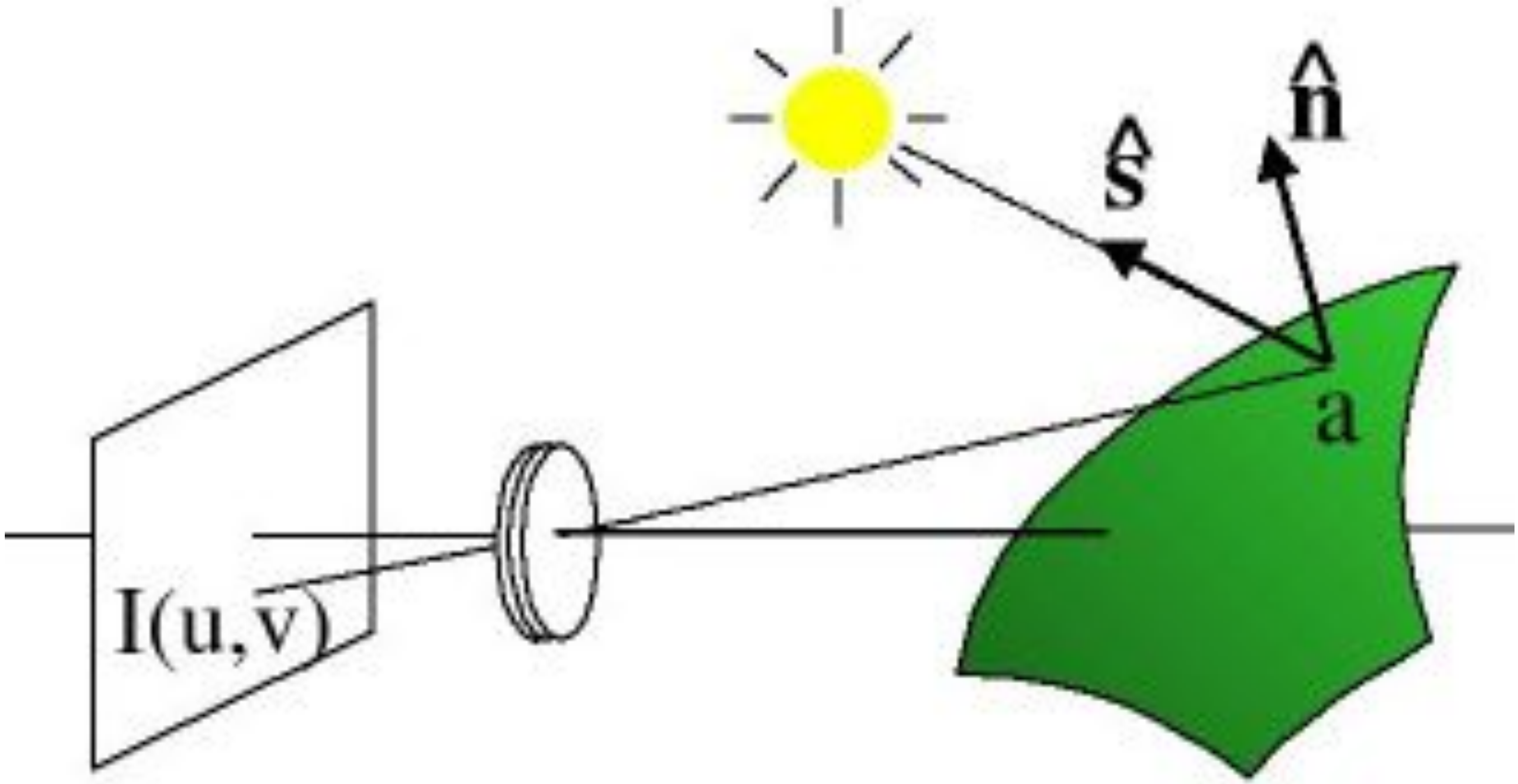


Credit: Alex Powell

Demo

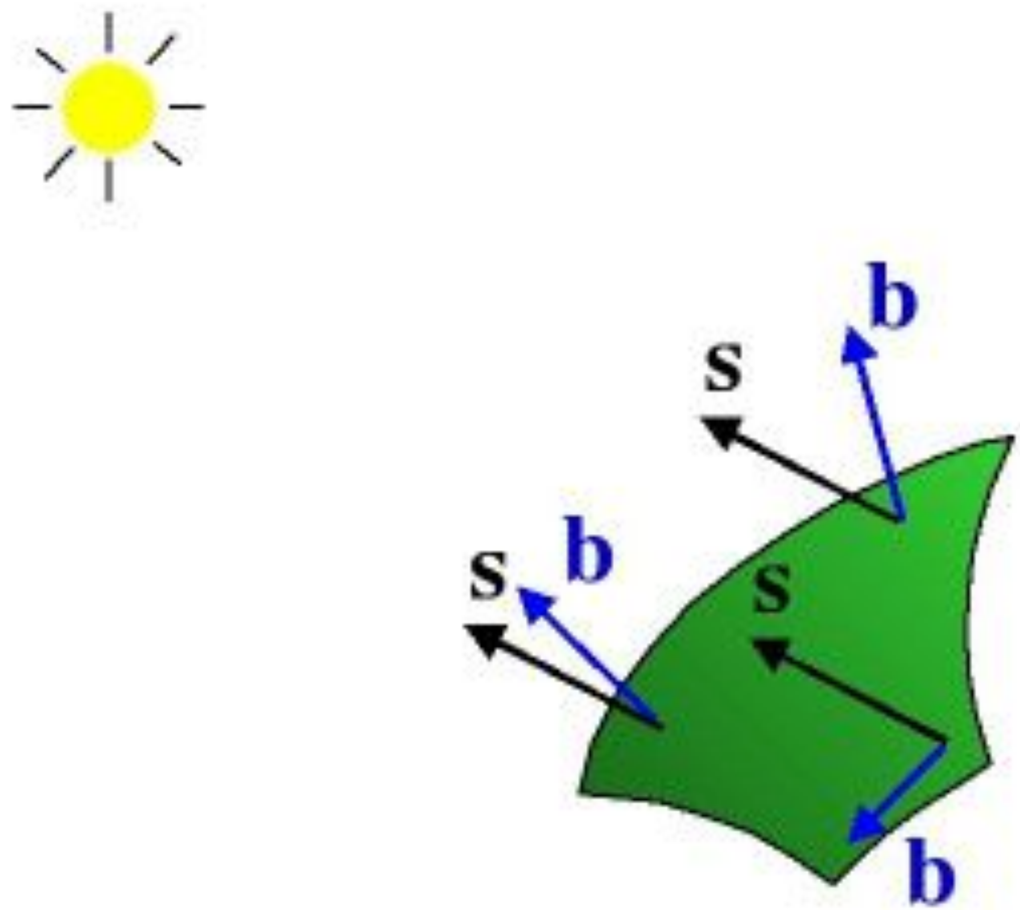
- Froggies

Geometry of Lighting

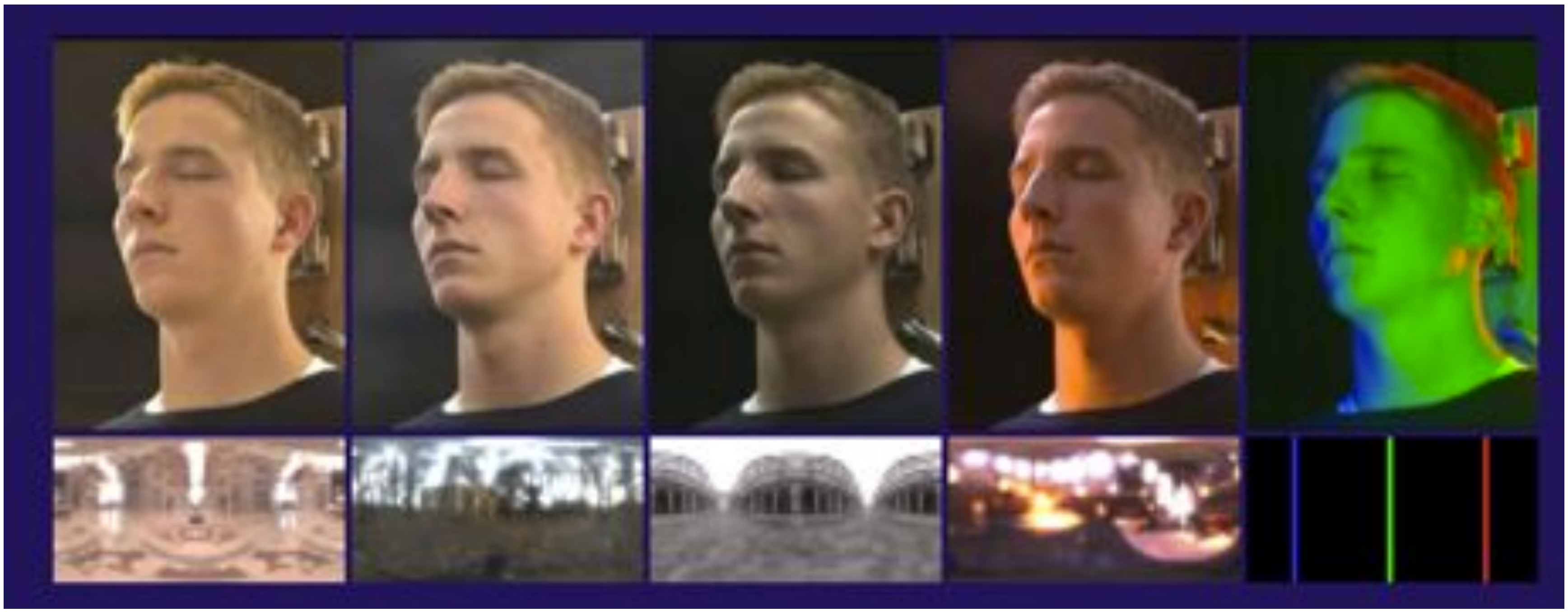


Lambertian Surface

Infinitely Distant Light Source



Acquiring the Reflectance Field of a Human Face: Paul Debevec et al



http://gl.ict.usc.edu/LightStages/

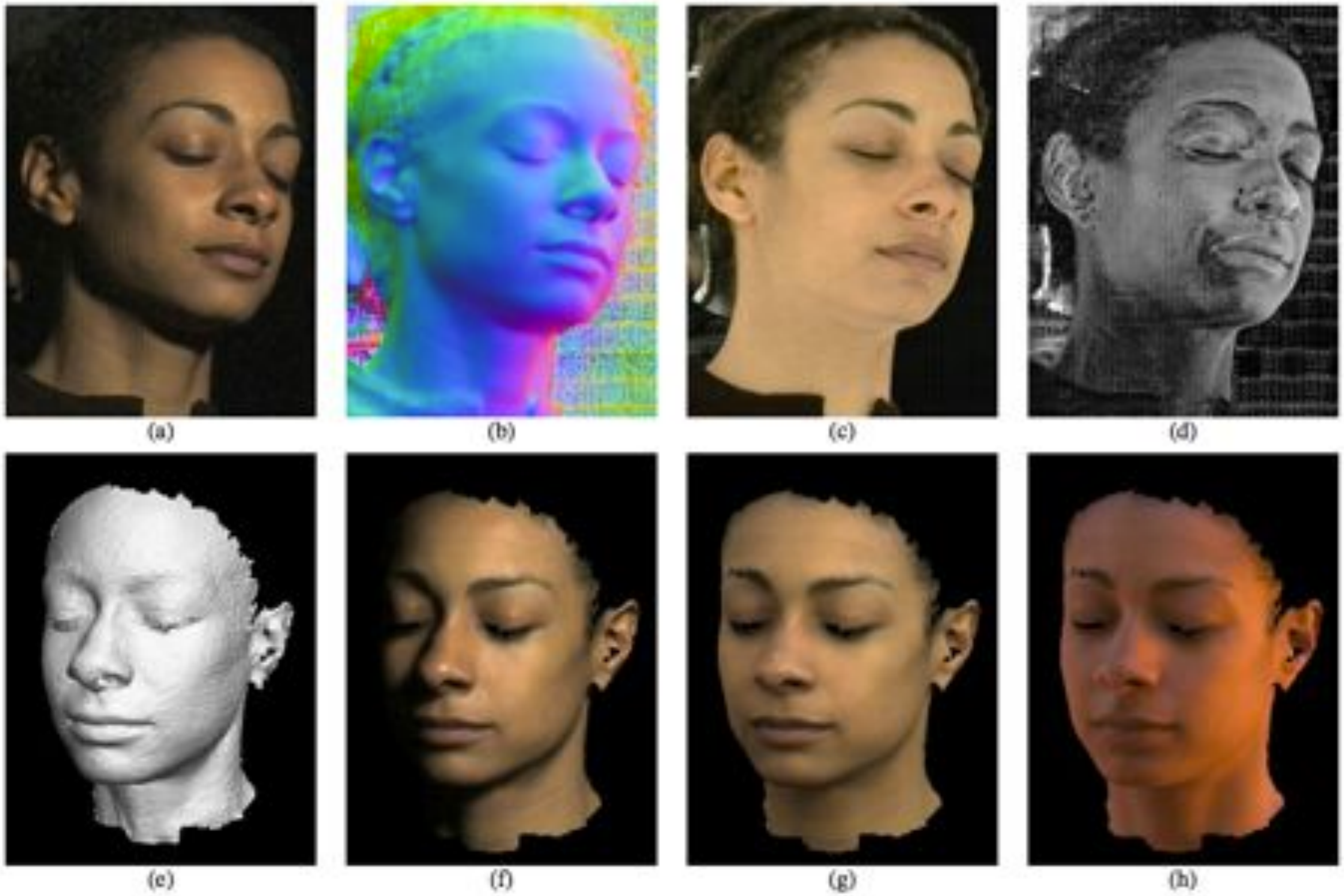


Figure 14: **Analyzing Reflectance and Changing the Viewpoint** (**a**) An original light stage image taken by the left camera. (**b**) Recovered surface normals n_d derived from the fitted diffuse reflectance lobe for each pixel; the RGB value for each pixel encodes the X, Y, and Z direction of each normal. (**c**) Estimated diffuse albedo ρ_d. Although not used by our rendering algorithm, such data could be used in a traditional rendering system. (**d**) Estimated specular energy ρ_s, also of potential use in a traditional rendering system. (**e**) Face geometry recovered using structured lighting. (**f**) Face rendered from a novel viewpoint under synthetic directional illumination. (**g**,**h**) Face rendered from a novel viewpoint under the two sampled lighting environments used in the second two renderings of Fig. 6.

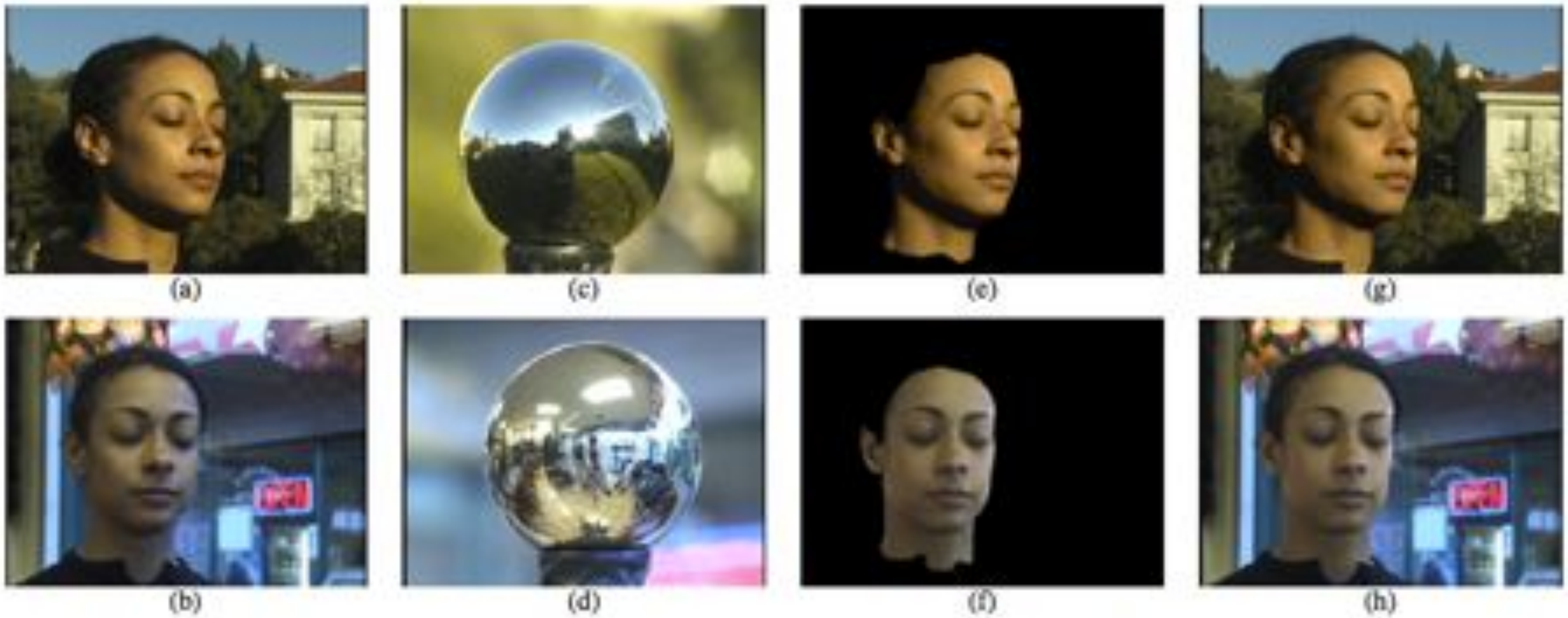


Figure 15: **Matching to Real-World Illumination** **(a,b)** Actual photographs of the subject in two different environments. **(c,d)** Images of a light probe placed in the position of the subject's head in the same environments. **(e,f)** Synthetic renderings of the face matched to the photographed viewpoints and illuminated by the captured lighting. **(g,h)** Renderings of the synthetic faces (e,f) composited over the original faces (a,b); the hair and shoulders come from the orginal photographs and are not produced using our techniques. The first environment is outdoors in sunlight; the second is indoors with mixed lighting coming from windows, incandescent lamps, and fluorescent ceiling fixtures.

More Info

- http://www.pauldebevec.com/
- http://www.pauldebevec.com/Research/LS/
- http://gl.ict.usc.edu/Research/DigitalEmily/
- http://gl.ict.usc.edu/Research/RHL/

Real-Time Photometric Stereo

Surface Normals
(Eigenvectors via Power Iteration)
6 ms

Height Field
(Integrate Surface Normals via Gauss-Seidel)
32 ms

Real-Time Photometric Stereo

Demo

- Processing Demo

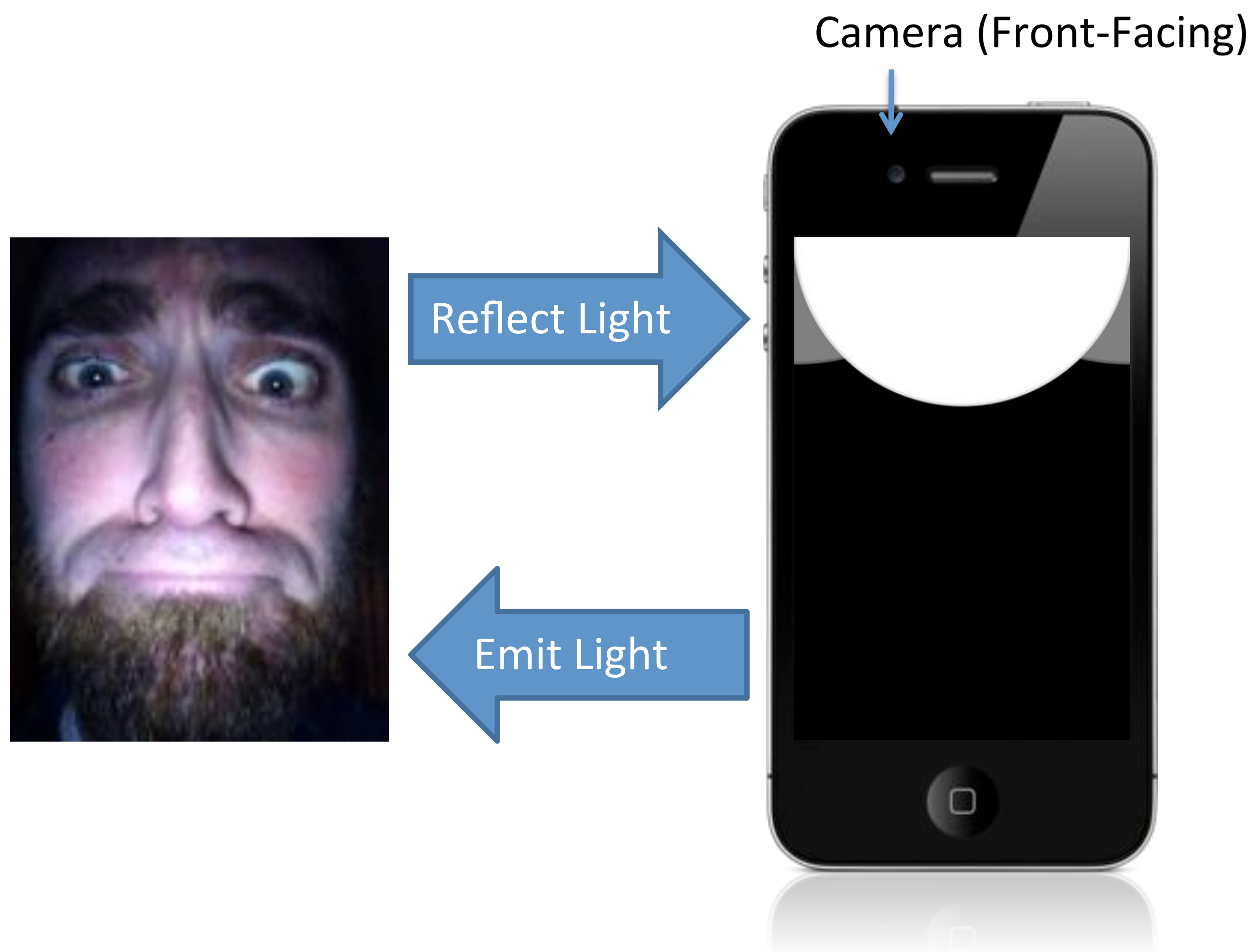
Reading

- Szeliski Book: Section 12.1.1 (page 580-583)

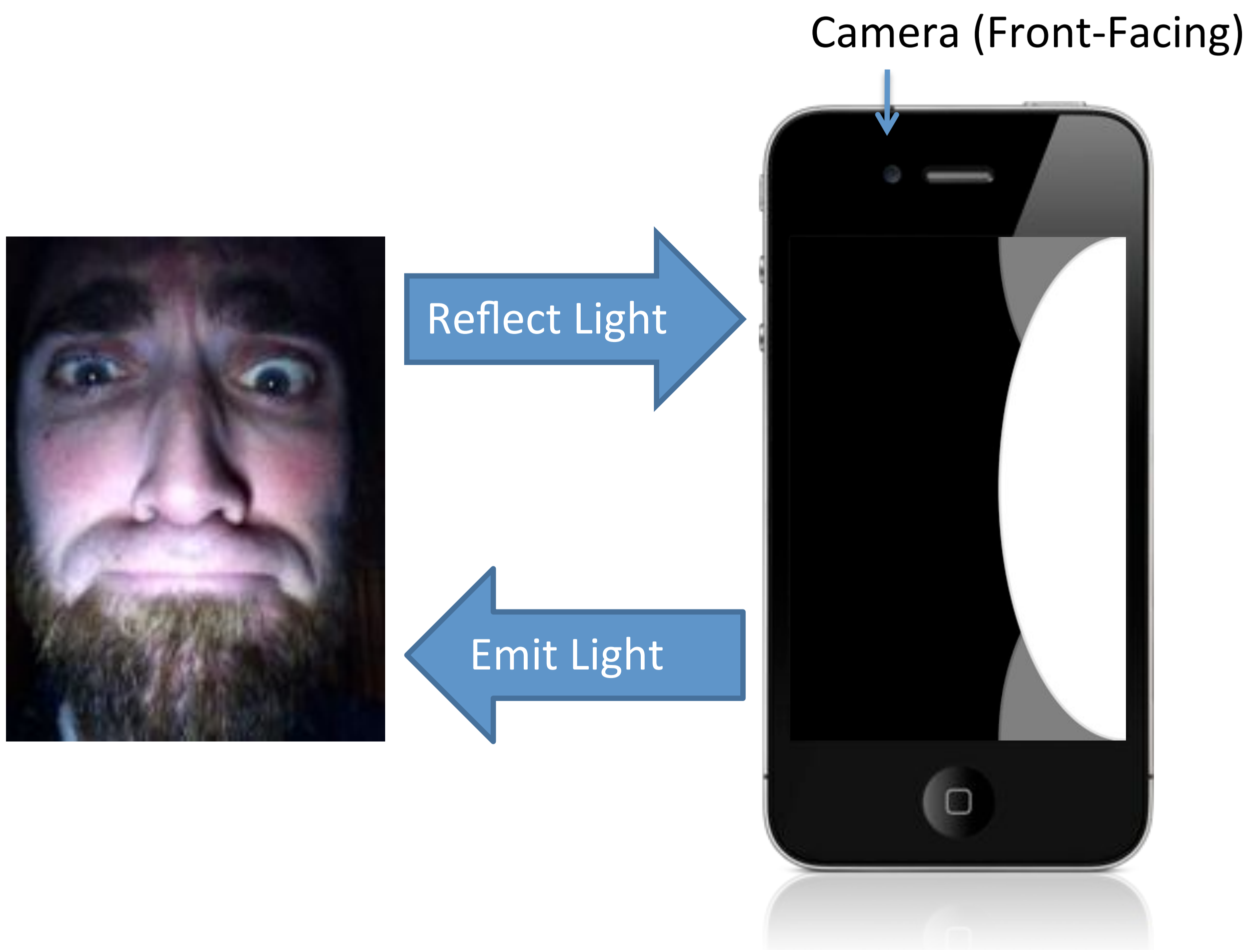
3D Scanner in Your Pocket

trimensional

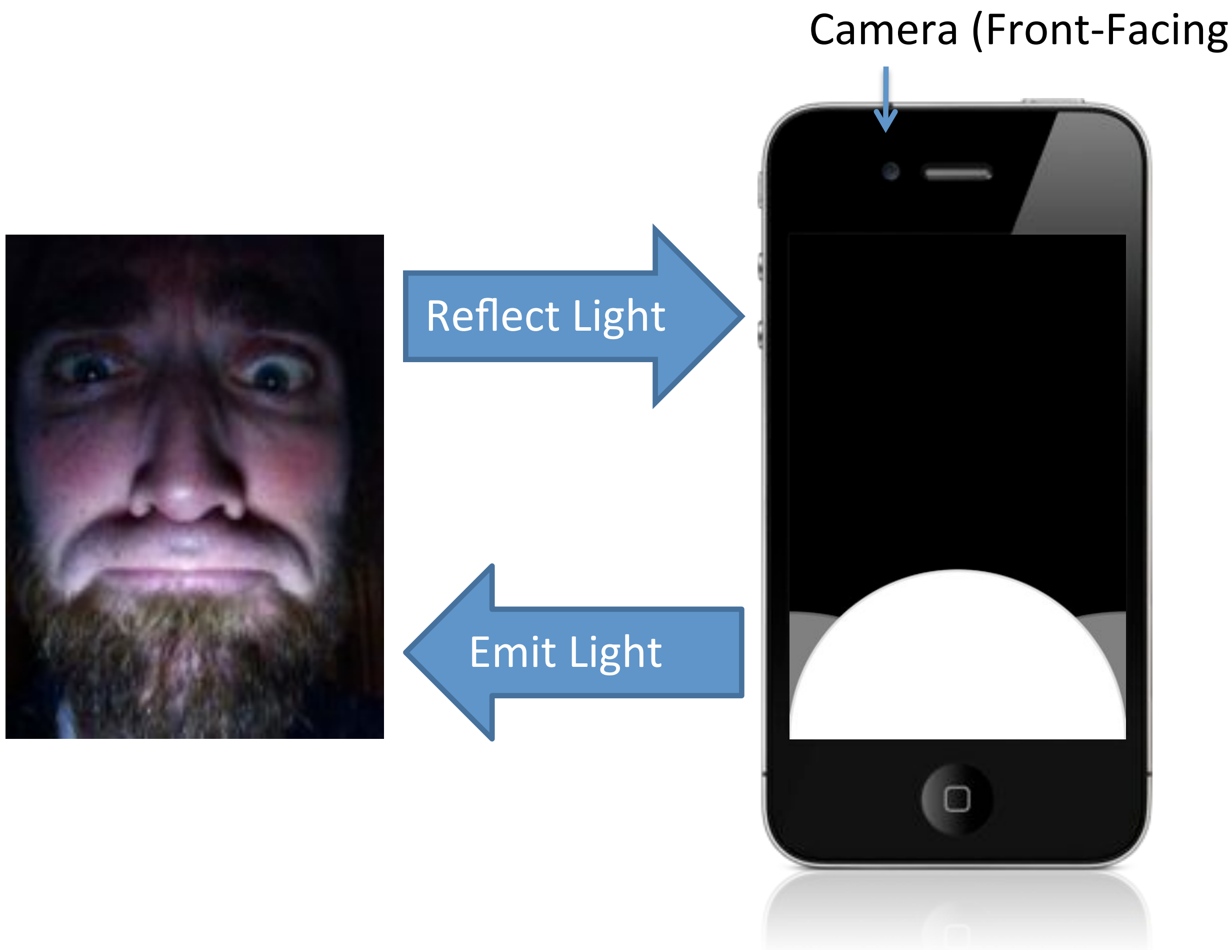
How does it work?



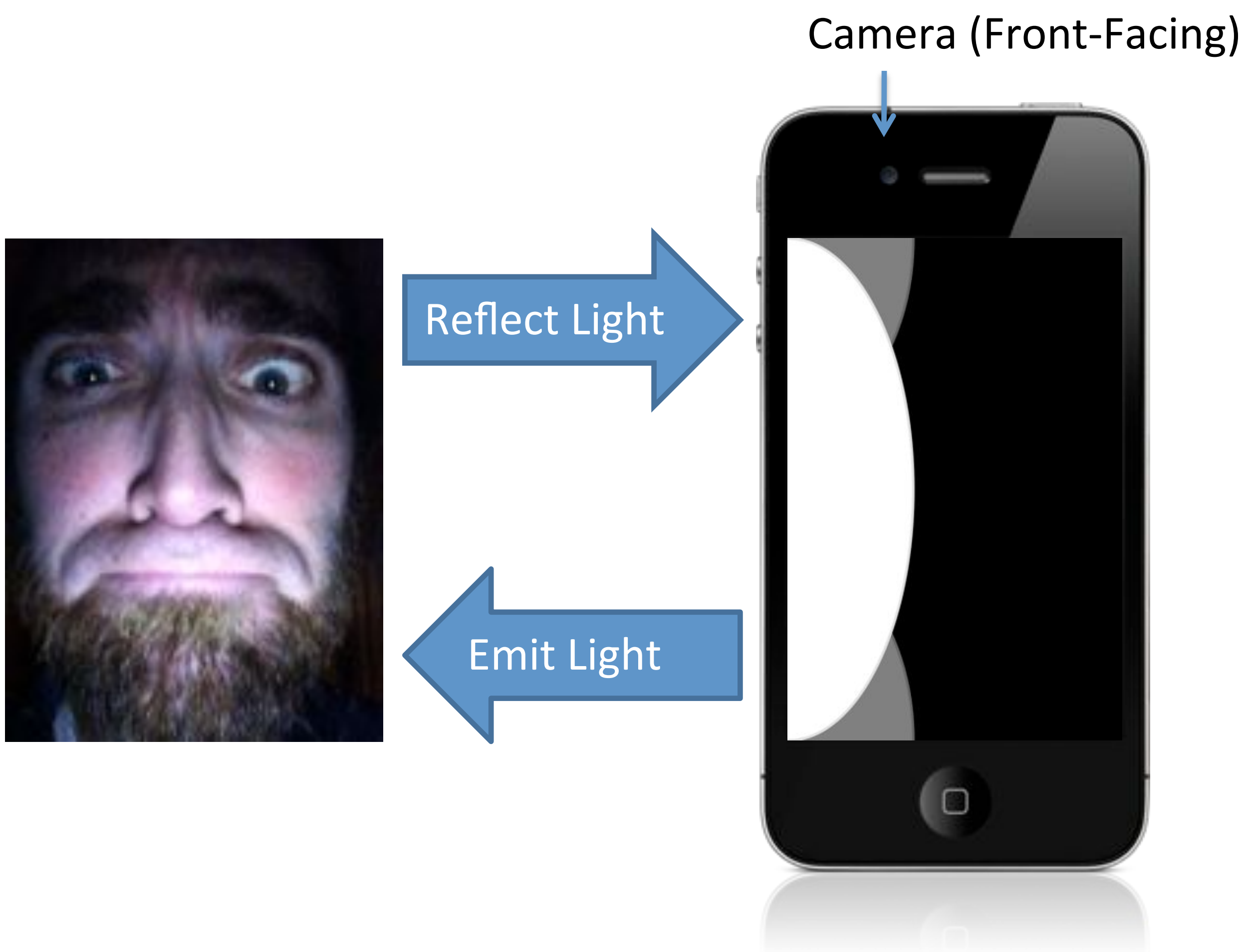
How does it work?



How does it work?



How does it work?



How does it work?

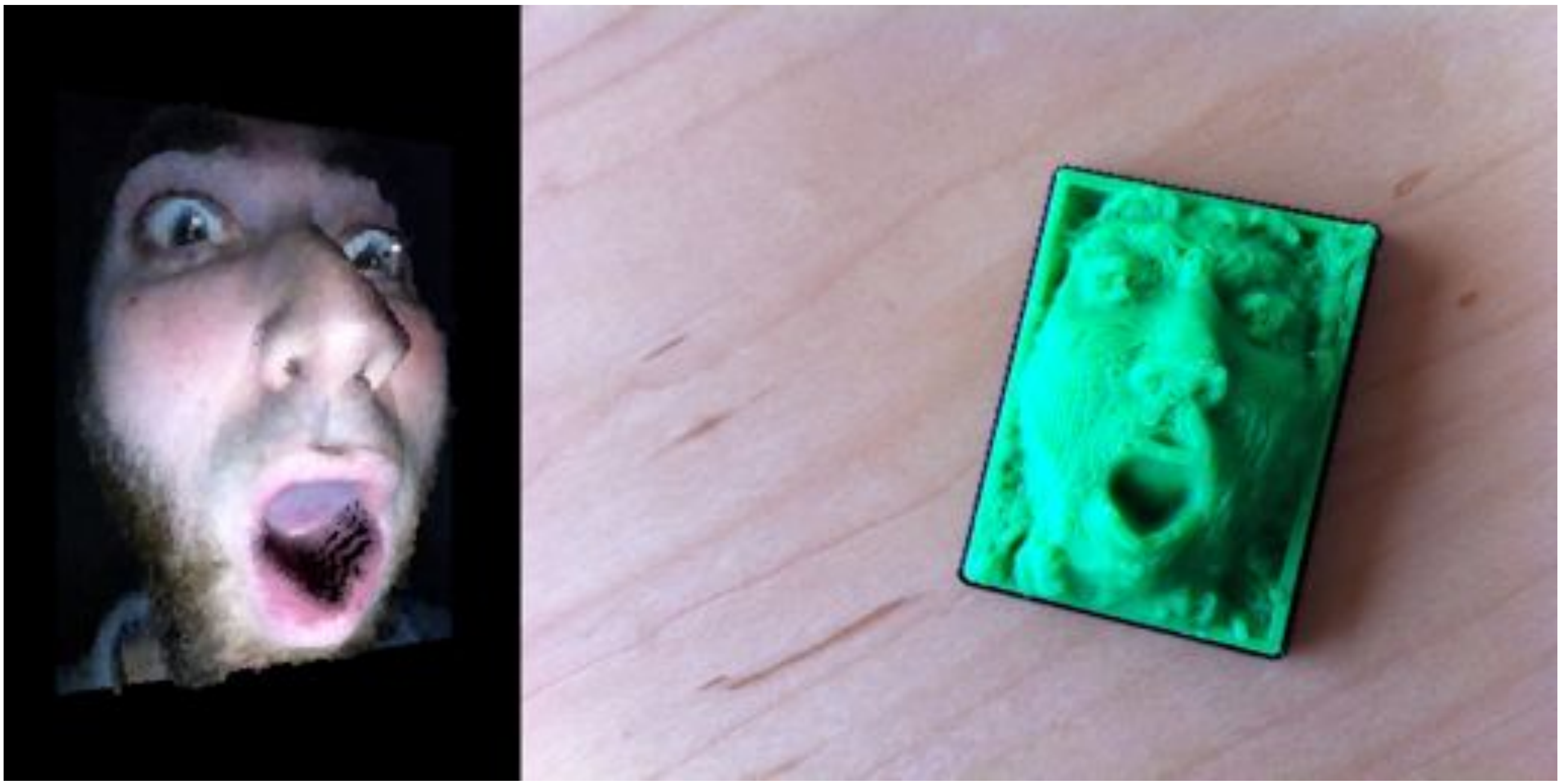
2D

3D

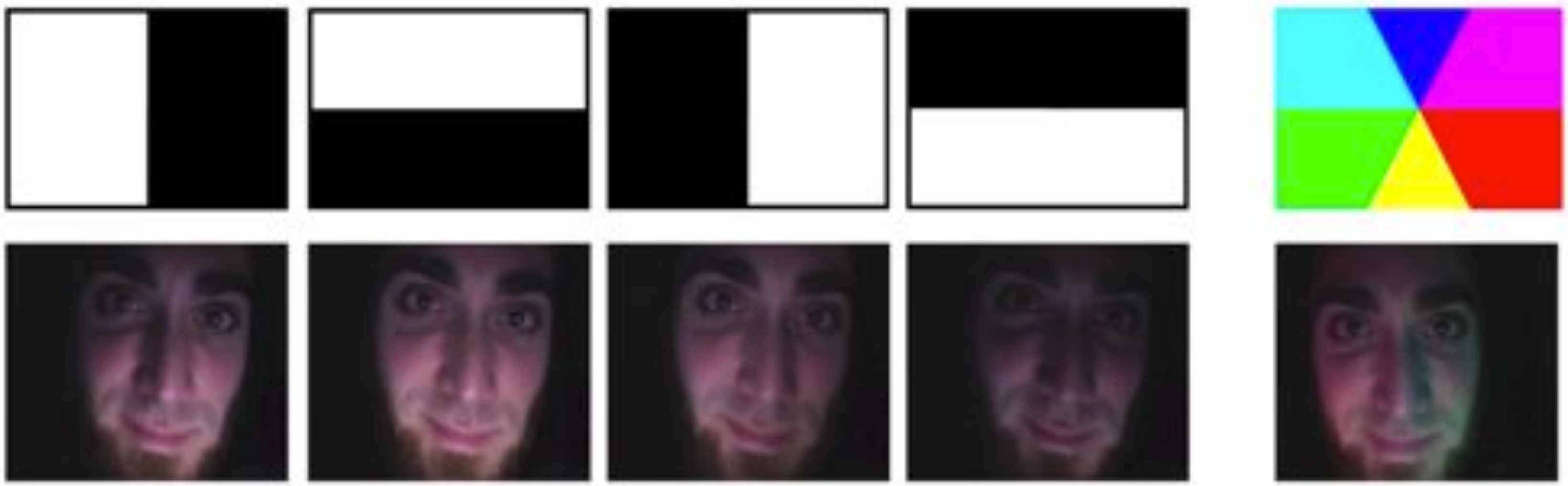

trimensional
3D Scanner for iPhone


Georgia Tech
Trimensional iPhone App


3D Scans
766 of 768
Done



Computer Screen Lighting

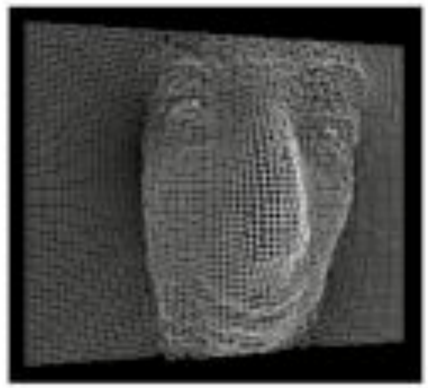


Recovering Surface Normals

Singular Value Decomposition. Images live in a 3-dimensional subspace defined by dominant eigenvectors corresponding to the z, x, and y components of the surface normal at every pixel in the image.

Surface Reconstruction

Depth Map



Wireframe Mesh

Textured Model

Gauss-Seidel Relaxation. Update depth estimate at each pixel based on measured surface normals and previously estimated values for pixel's neighbors in the depth map. Iterate.

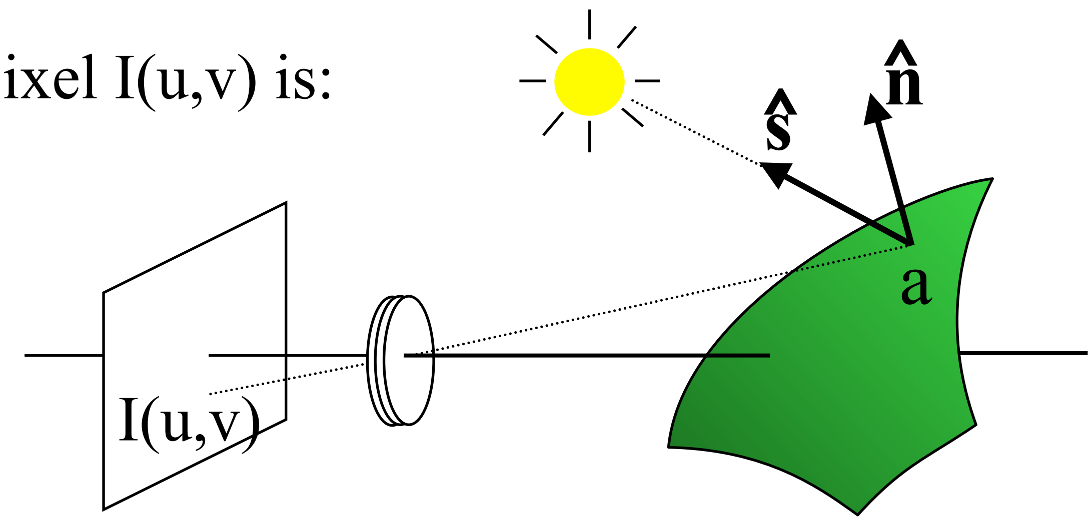
Real-time Reconstruction

Continually light the scene, capture an image, recompute surface normals, update depth map, and display resulting textured model.

Currently runs at 10 frames per second on 320x240 images. Timings for each step of the algorithm during a single cycle are: surface normal computation (9 ms), depth map computation (32 ms), combined lighting, 3D display, and video capture (26 ms).

Lambertian Surface

The intensity of a pixel I(u,v) is:



$$I(u,v) = [a(u,v)\,\hat{\mathbf{n}}(u,v)] \bullet [s_0\,\hat{\mathbf{s}}] = \mathbf{b}(u,v) \bullet \mathbf{s}$$

a(u,v) is the albedo of the surface projecting to (u,v).

n(u,v) is the direction of the surface normal.

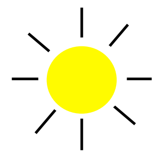
s_0 is the light source intensity.

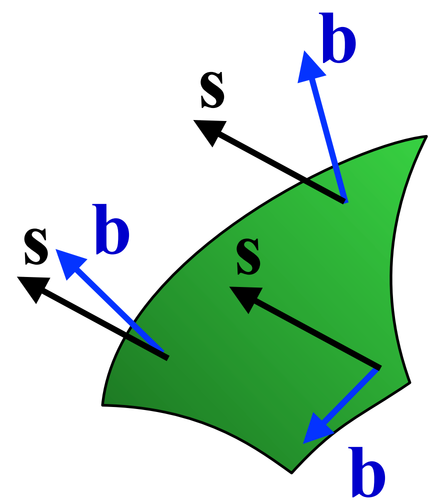
s is the direction to the light source.

Slide courtesy of
David Kriegman

Orthographic Projection

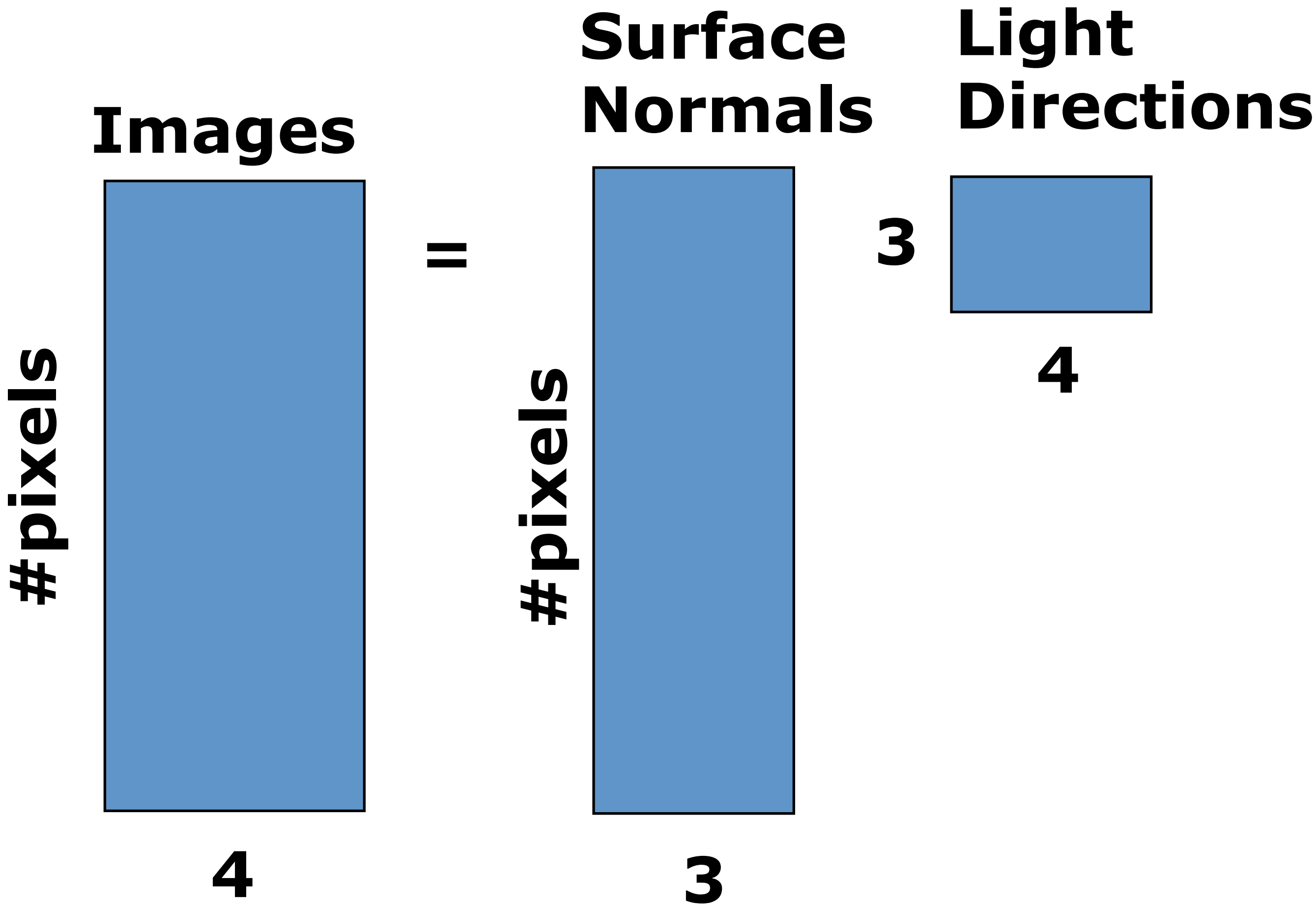
- Simplification for light sources that are sufficiently far away from an object.
- All incoming light rays are parallel.
- Thus, while **b** vectors vary over the surface, **s** vector is *constant*.



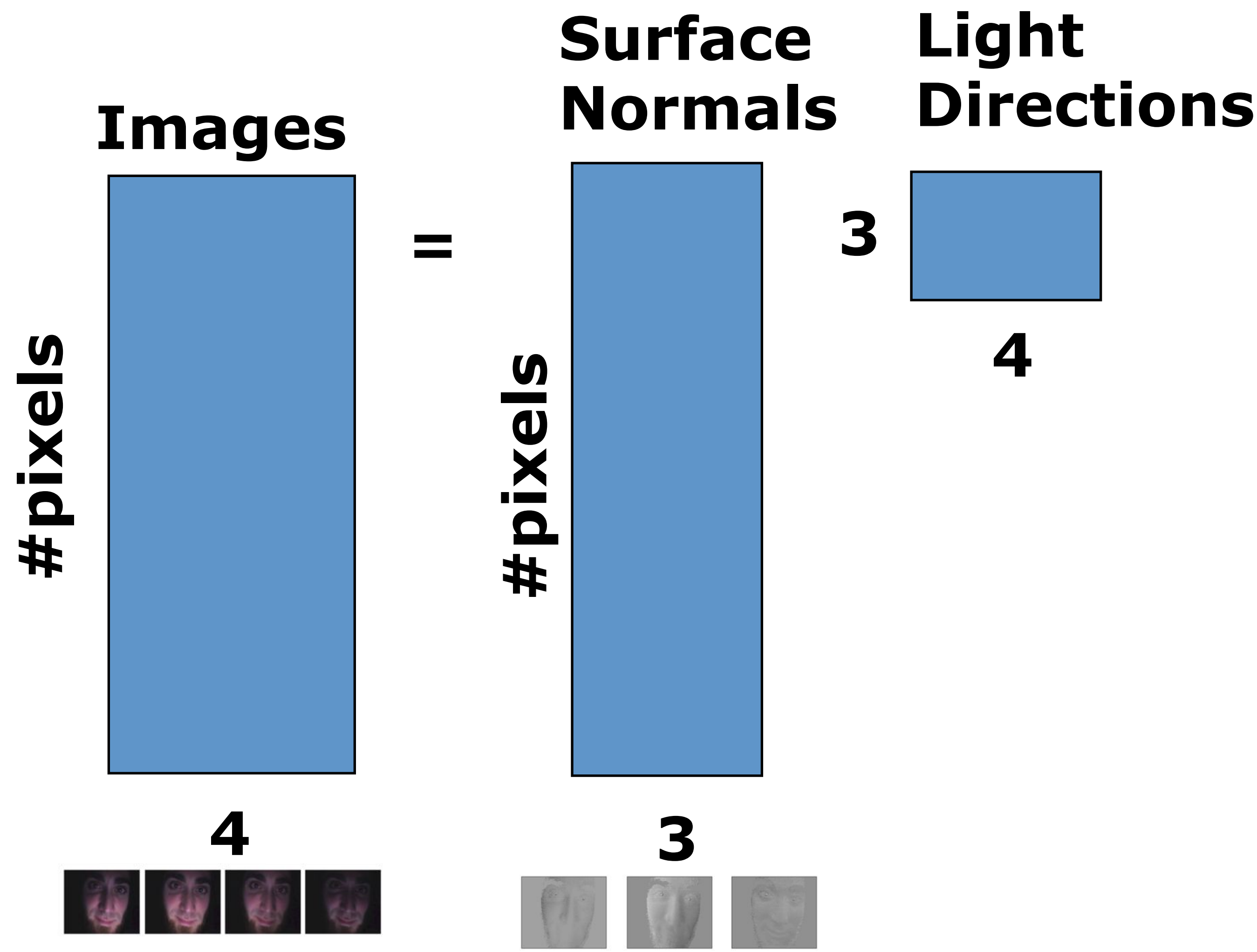


$$\text{Pixels}: \mathbf{b}_1^T\mathbf{s}, \mathbf{b}_2^T\mathbf{s}, \mathbf{b}_3^T\mathbf{s}, \ldots \Rightarrow \mathbf{Bs}$$

Images Live in a 3-D Subspace of All Possible Images



Images Live in a 3-D Subspace of All Possible Images

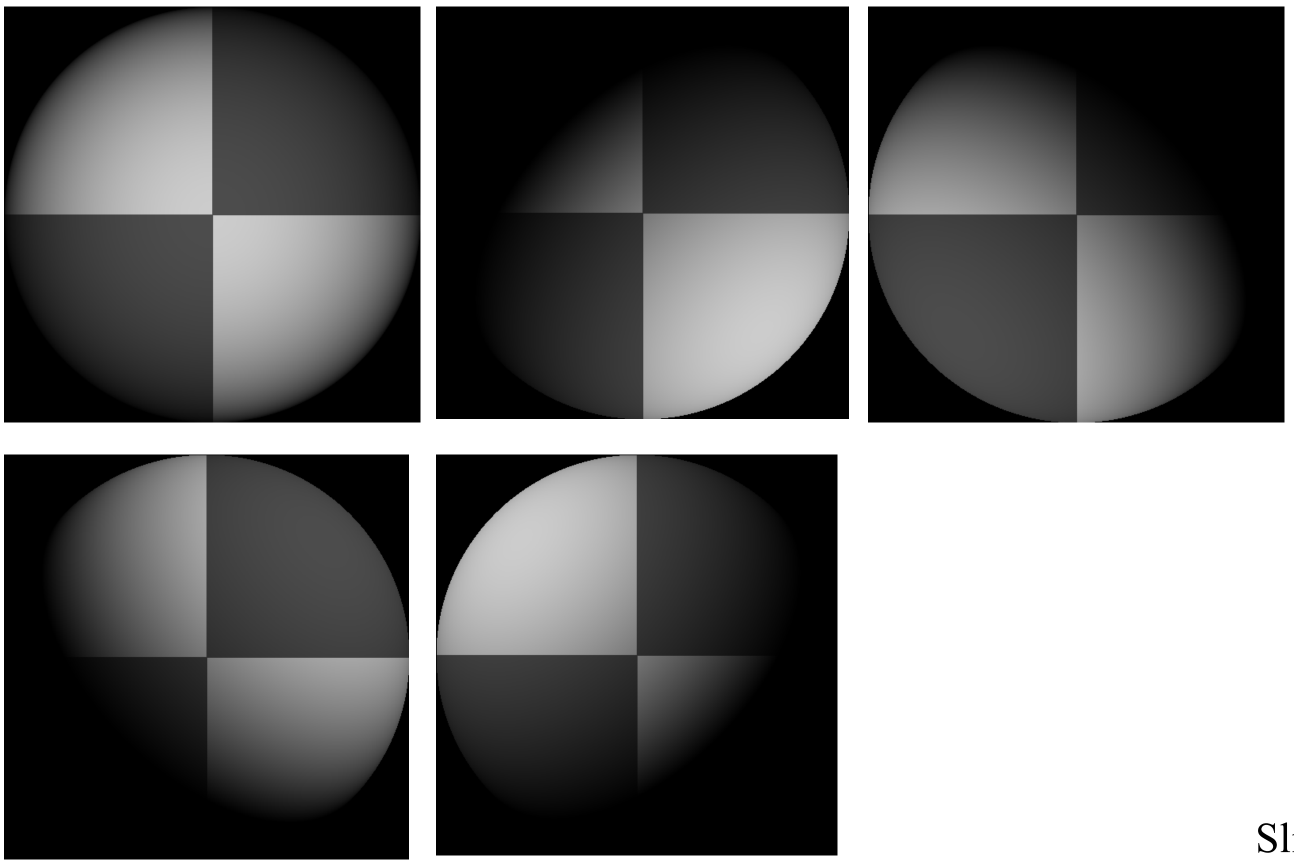


Background: Linear Algebra

- http://www.cc.gatech.edu/~phlosoft/transforms/
- Keywords: Singular Value Decomposition
- Related: Image Compression

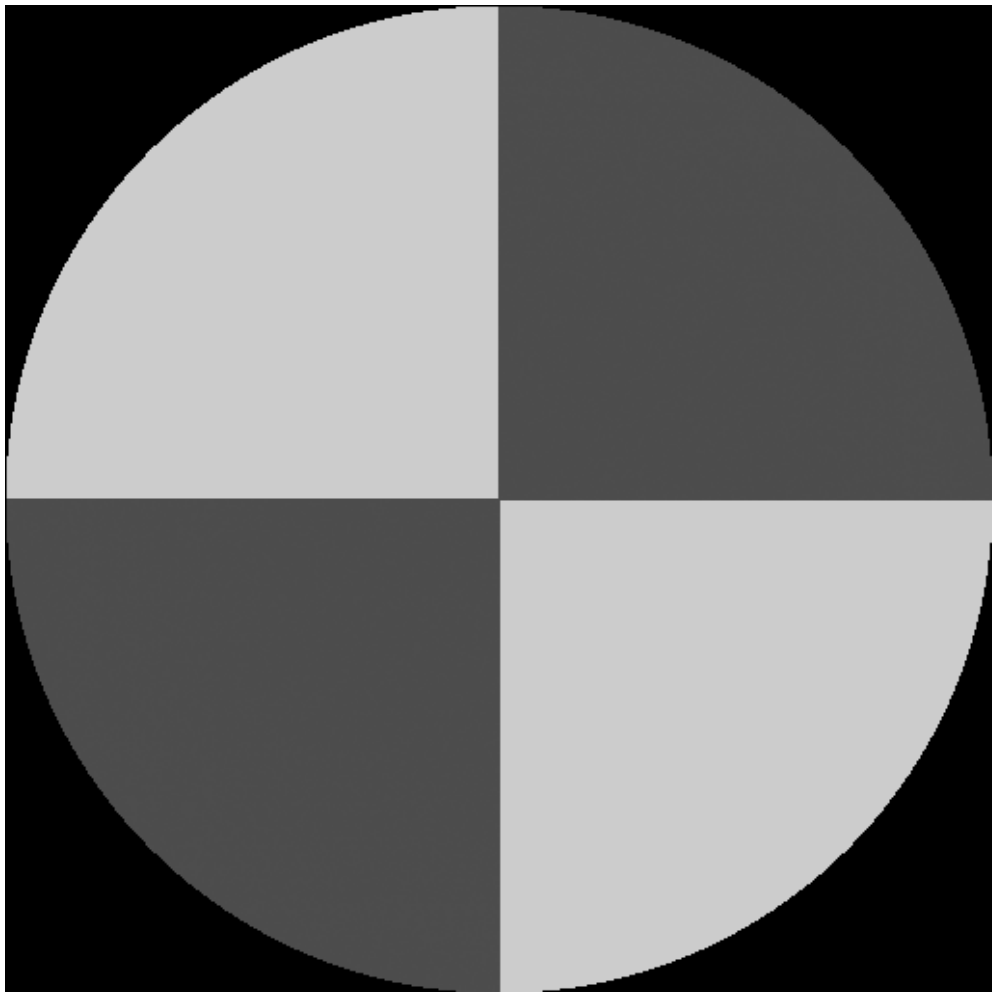
Synthetic Sphere Images

Five different lighting conditions



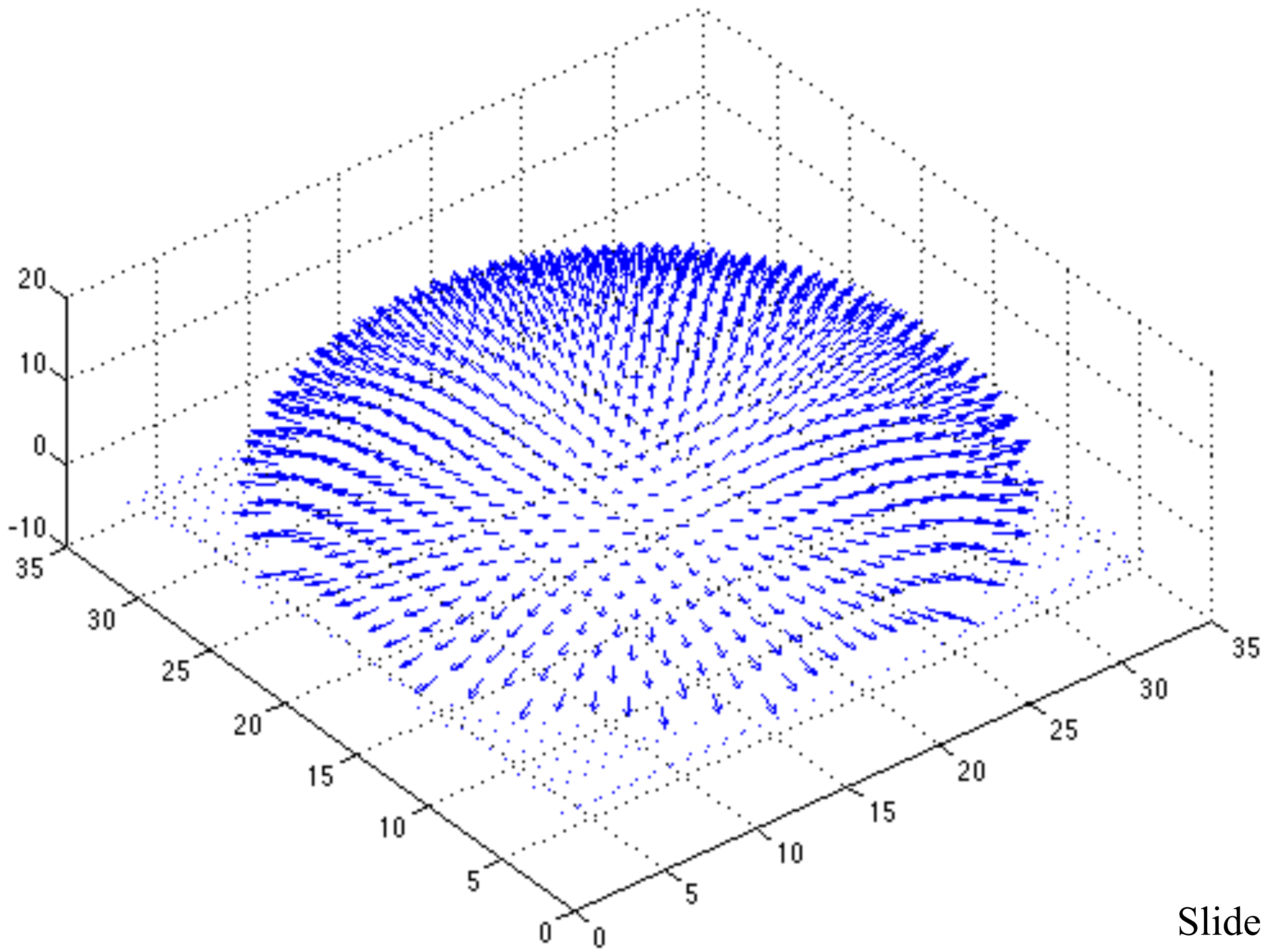
Slide courtesy of
David Forsyth

Recovered Albedo



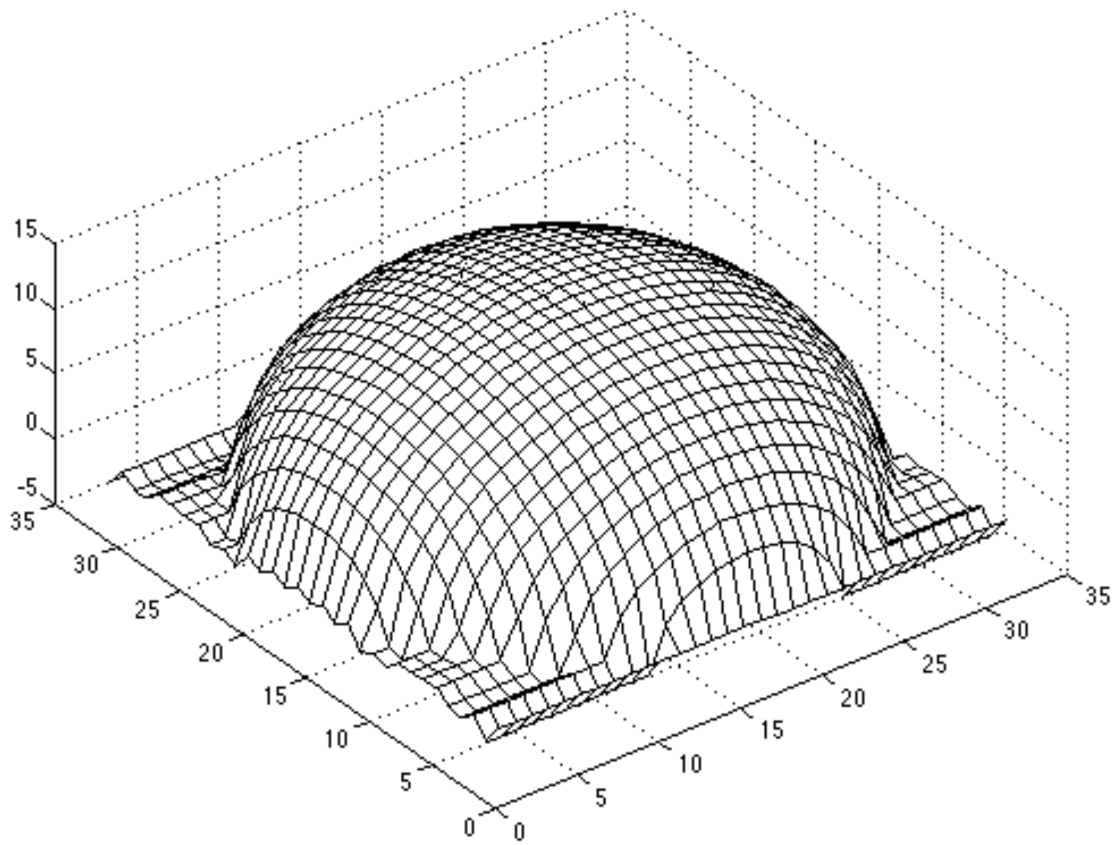
Slide courtesy of
David Forsyth

Recovered Surface Normals



Slide courtesy of
David Forsyth

Recovered Surface Shape



Recovery up to a constant depth error (not absolute depth)

Slide courtesy of David Forsyth

Integration Methods

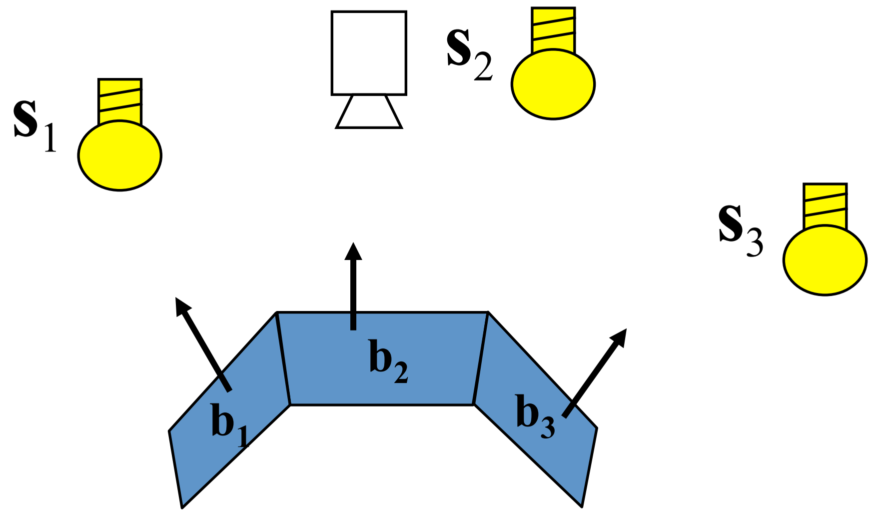
-Frankot-Chellappa (FFT)
-Gauss-Seidel (Iterative Conditional Modes)
-Path Integration

High Quality Results

George Vogiatzis and Carlos Hernandez

Photometric Stereo

Photometric Stereo



- Given multiple images of the same surface under different known lighting conditions, can we recover the surface shape?
  - Yes! (Woodham, 1978)

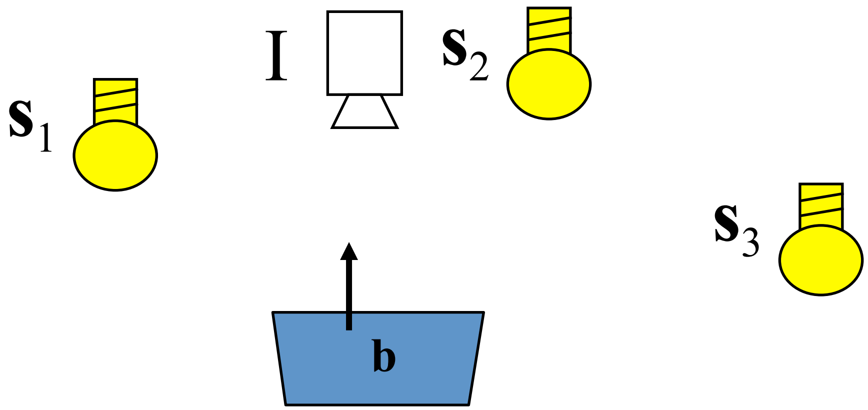
Photometric Stereo

- Assume:
  - A set of point sources that are infinitely distant
  - A set of pictures of an object, obtained in exactly the same camera/object configuration but using different sources

$$\mathbf{I}_1 = \mathbf{B}\mathbf{s}_1; \quad \mathbf{I}_2 = \mathbf{B}\mathbf{s}_2; \quad \mathbf{I}_3 = \mathbf{B}\mathbf{s}_3 \ \ldots$$

  - A Lambertian object (or the specular component has been identified and removed)

Stereo for a Pixel



For a pixel (x,y) we have n measurements:

$$I_1(x,y) = \mathbf{s}_1^T \mathbf{b}(x,y); \quad I_2(x,y) = \mathbf{s}_2^T \mathbf{b}(x,y) \ \ldots \Rightarrow \mathbf{I}(x,y) = \mathbf{S}\mathbf{b}(x,y)$$

*Solve an over-constrained linear system for **b** (with n>3)*

What About Shadows?

- Shadowed pixels (e.g. attached shadows for a given light source position) are outliers.
- "Max" trick can be adapted for this case too:

$$\begin{bmatrix} I_1(x,y) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & I_n(x,y) \end{bmatrix} \begin{bmatrix} I_1(x,y) \\ \vdots \\ I_n(x,y) \end{bmatrix} = \begin{bmatrix} I_1(x,y) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & I_n(x,y) \end{bmatrix} \begin{bmatrix} \mathbf{s}_1^T \\ \vdots \\ \mathbf{s}_n^T \end{bmatrix} \mathbf{b}(x,y)$$

Pre-multiplying by a thresholded weight matrix zeros the contributions from shadowed pixels

Recovering the Albedo

Recall that $\mathbf{b}(x,y) = a(x,y)\hat{\mathbf{n}}(x,y)$

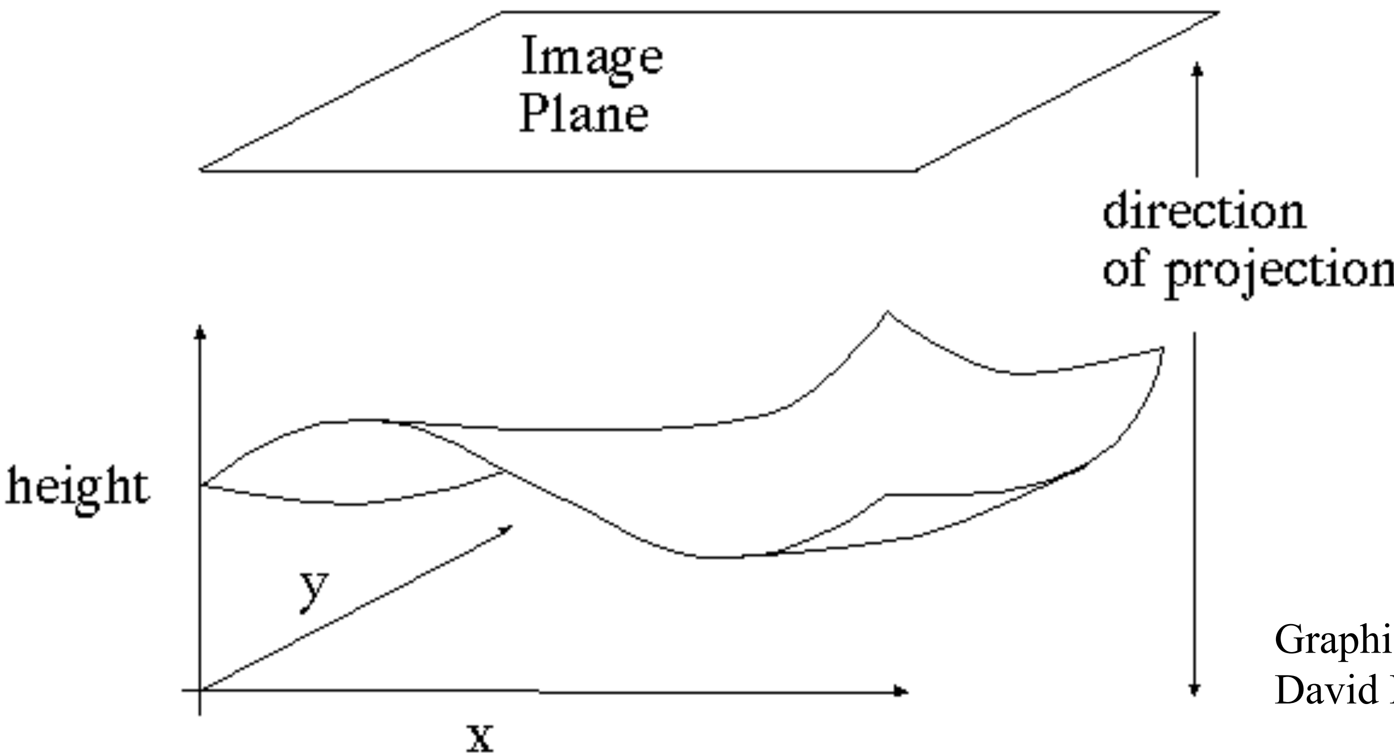
$$\Rightarrow a(x,y) = \left\|\mathbf{b}(x,y)\right\|$$

- This gives a check on the normal recovery at a pixel
  - If the magnitude of $a(x, y)$ is greater than 1, there's a problem

  Then $\hat{\mathbf{n}}(x,y) = \mathbf{b}(x,y)/a(x,y)$

Recovering the Surface Shape

Depth map model (also called Monge patch):



Graphic courtesy of
David Forsyth

$z = f(x, y)$ Surface is set of points $\{x, y, f(x, y)\}$

Recovering a surface from normals - 1

- Recall the surface is written as

$$(x, y, f(x, y))$$

- This means the normal has the form:

$$N(x,y) = \left(\frac{1}{\sqrt{f_x^2 + f_y^2 + 1}} \right) \begin{pmatrix} -f_x \\ -f_y \\ 1 \end{pmatrix}$$

- If we write the known vector **g** as

$$\mathbf{g}(x,y) = \begin{pmatrix} g_1(x,y) \\ g_2(x,y) \\ g_3(x,y) \end{pmatrix}$$

- Then we obtain values for the partial derivatives of the surface:

$$f_x(x,y) = \left(g_1(x,y) / g_3(x,y) \right)$$
$$f_y(x,y) = \left(g_2(x,y) / g_3(x,y) \right)$$

Slide courtesy of David Forsyth

Recovering a surface from normals - 2

- Recall that mixed second partials are equal --- this gives us a **check.** We must have:

$$\frac{\partial\big(g_1(x,y)/g_3(x,y)\big)}{\partial y} =$$

$$\frac{\partial\big(g_2(x,y)/g_3(x,y)\big)}{\partial x}$$

- (or they should be similar, at least)

- We can now recover the surface height at any point by integration along some path, e.g.

$$f(x,y) = \int_0^x f_x(s,y)\,ds +$$

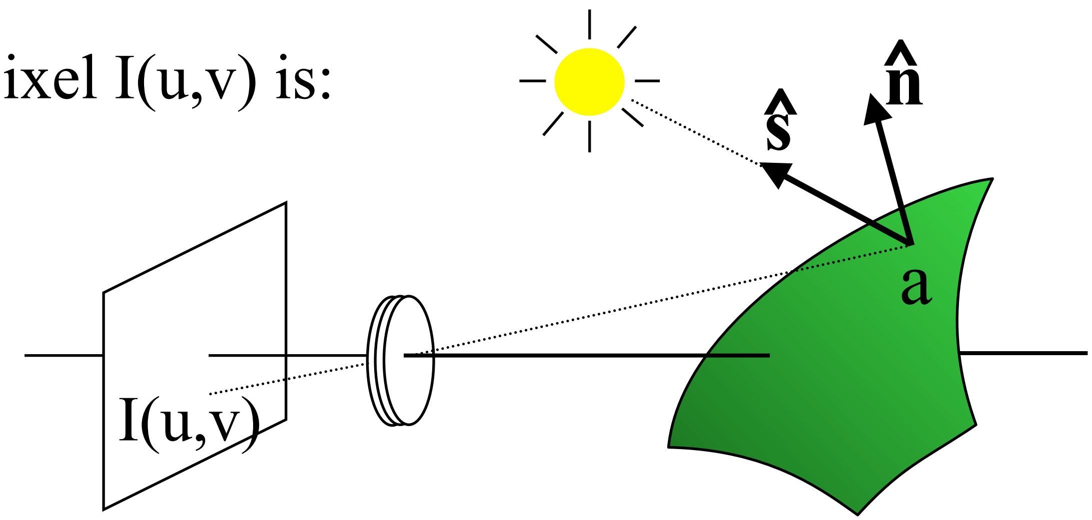
$$\int_0^y f_y(x,t)\,dt + c$$

Slide courtesy of David Forsyth

Light Sources

Lambertian Surface

The intensity of a pixel I(u,v) is:



$$I(u,v) = [a(u,v)\,\hat{\mathbf{n}}(u,v)] \bullet [s_0\,\hat{\mathbf{s}}] = \mathbf{b}(u,v) \bullet \mathbf{s}$$

a(u,v) is the albedo of the surface projecting to (u,v).

n(u,v) is the direction of the surface normal.

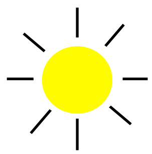
s_0 is the light source intensity.

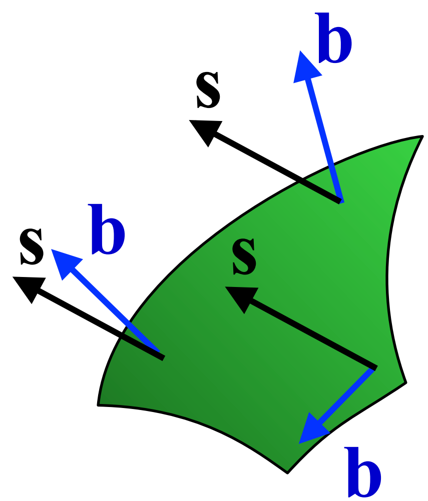
s is the direction to the light source.

Slide courtesy of
David Kriegman

Orthographic Projection

- Simplification for light sources that are sufficiently far away from an object.
- All incoming light rays are parallel.
- Thus, while **b** vectors vary over the surface, **s** vector is *constant*.





$$\text{Pixels}: \mathbf{b}_1^T\mathbf{s}, \mathbf{b}_2^T\mathbf{s}, \mathbf{b}_3^T\mathbf{s}, \ldots \Rightarrow \mathbf{Bs}$$