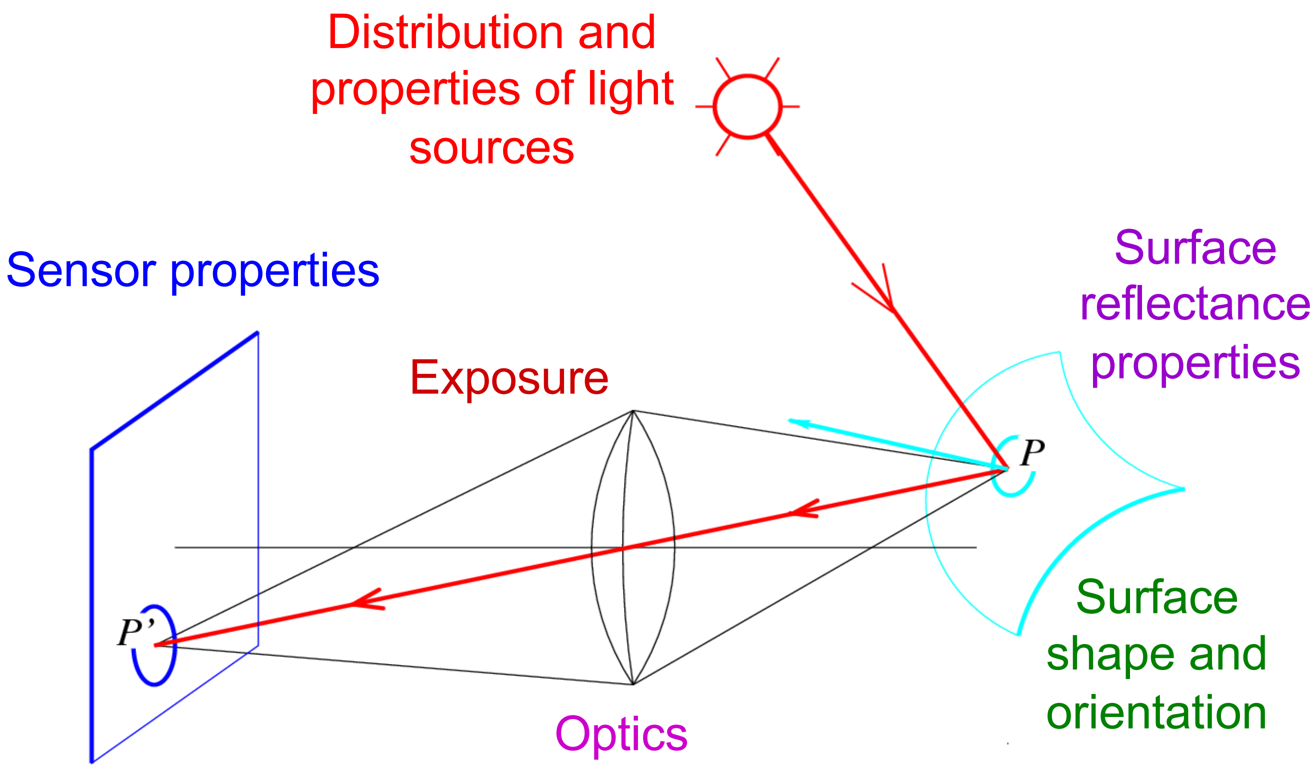
Light and shading

P. Claesz, Still Life with a Skull and a Writing Quill, 1628

Image formation

- What determines the *brightness* of an image pixel?

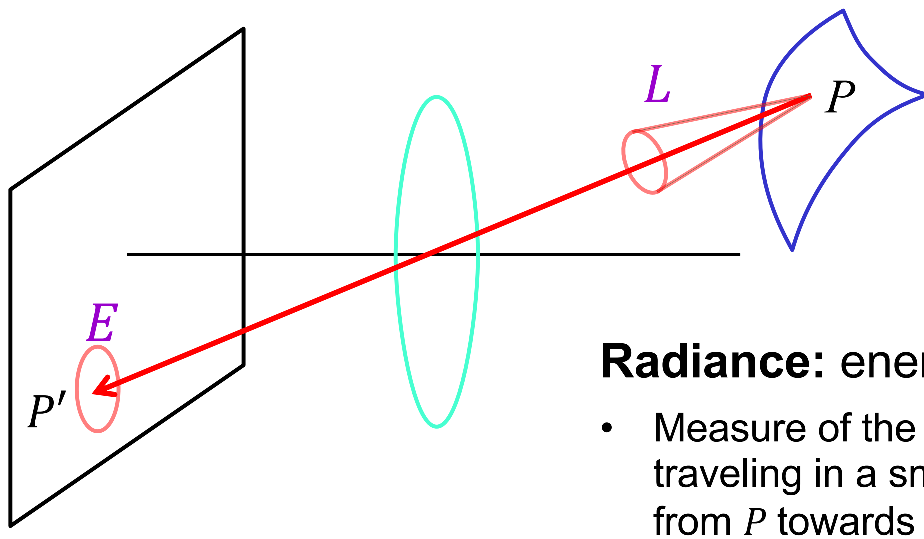


Slide by L. Fei-Fei

Outline

- Small taste of radiometry
- In-camera transformation of light
- Reflectance properties of surfaces
- Diffuse and specular reflection
- Shape from shading
- Estimating direction of light sources

Radiometry of image formation



Irradiance: energy arriving at a surface

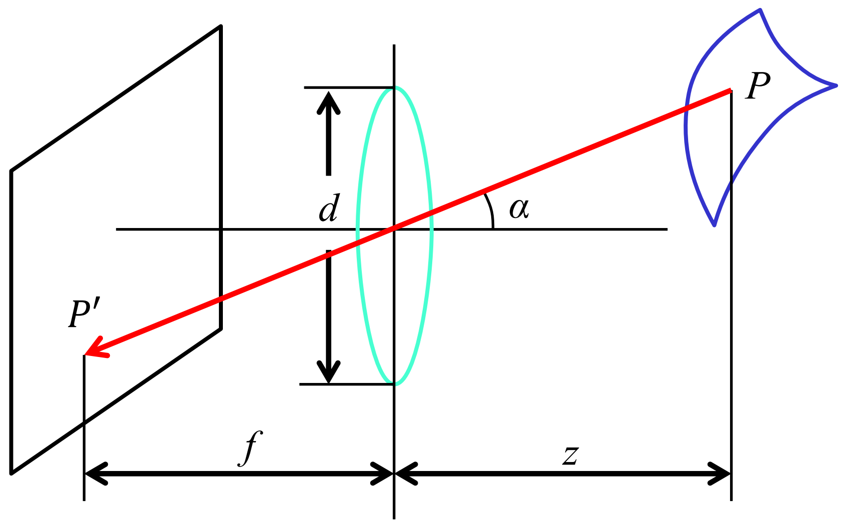
- Incident power per unit area (not foreshortened)
- Units: Watts per square meter

Radiance: energy carried by a ray

- Measure of the density of photons traveling in a small cone of directions from P towards P'
- Power per unit area perpendicular to the direction of travel, per unit solid angle
- Units: Watts per square meter per steradian

What is the relationship between E and L?

Fundamental radiometric relation

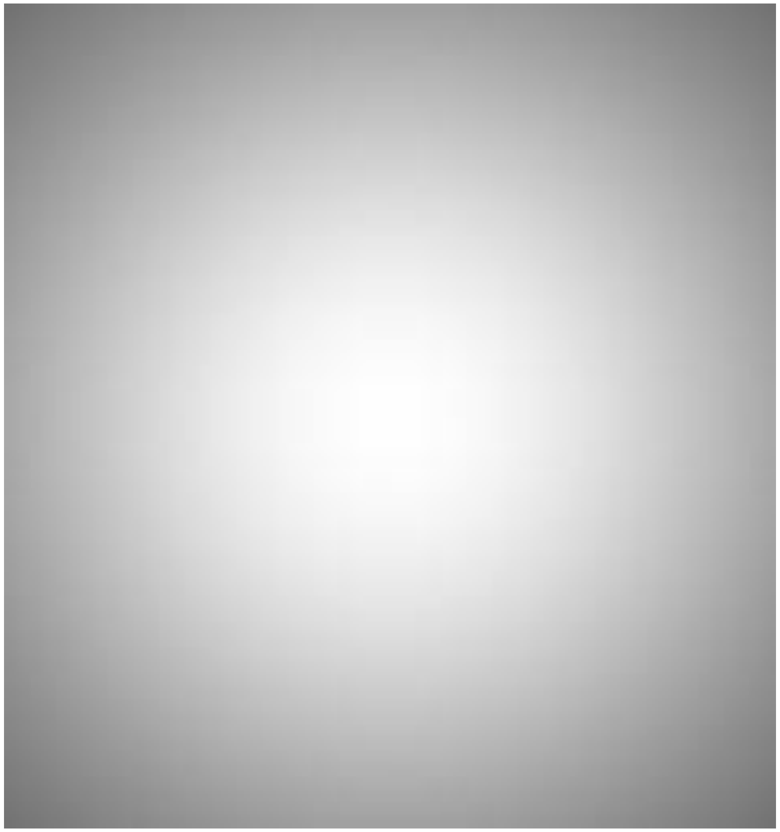


$$E = \left[\frac{\pi}{4}\left(\frac{d}{f}\right)^2 \cos^4\alpha\right] L$$

- Image irradiance (E) is linearly related to scene radiance (L)
- Irradiance is *directly* proportional to the area of the lens ($\frac{\pi d^2}{4}$) and *inversely* proportional to the squared distance between the lens and the image plane (f)
- The irradiance decreases as the angle between the viewing ray and the optical axis (α) increases

For derivation, see, e.g., Szeliski 2.2.3

Fundamental radiometric relation

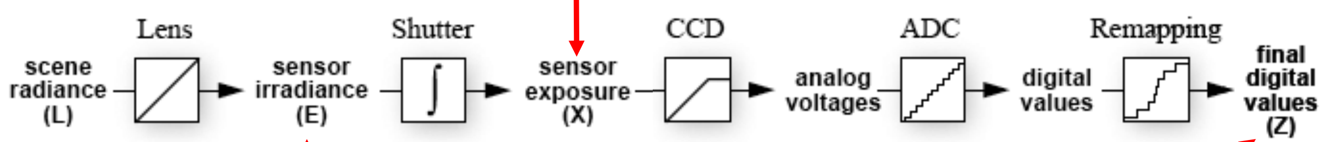


$$E = \left[\frac{\pi}{4}\left(\frac{d}{f}\right)^2 \cos^4\alpha\right] L$$

S. B. Kang and R. Weiss. Can we calibrate a camera using an image of a flat, textureless Lambertian surface? ECCV 2000

From light rays to pixel values

$$X = E \cdot \Delta t$$



$$E = \left[\frac{\pi}{4}\left(\frac{d}{f}\right)^2 \cos^4\alpha\right] L$$

$$Z = f(E \cdot \Delta t)$$

- **Camera response function**: the mapping f from irradiance to pixel values
  - Needed for applications like estimation of scene reflectance properties, creating high dynamic range (HDR) images
  - For further reading: M. Brown, Understanding the In-Camera Image Processing Pipeline for Computer Vision, CVPR 2016 Tutorial

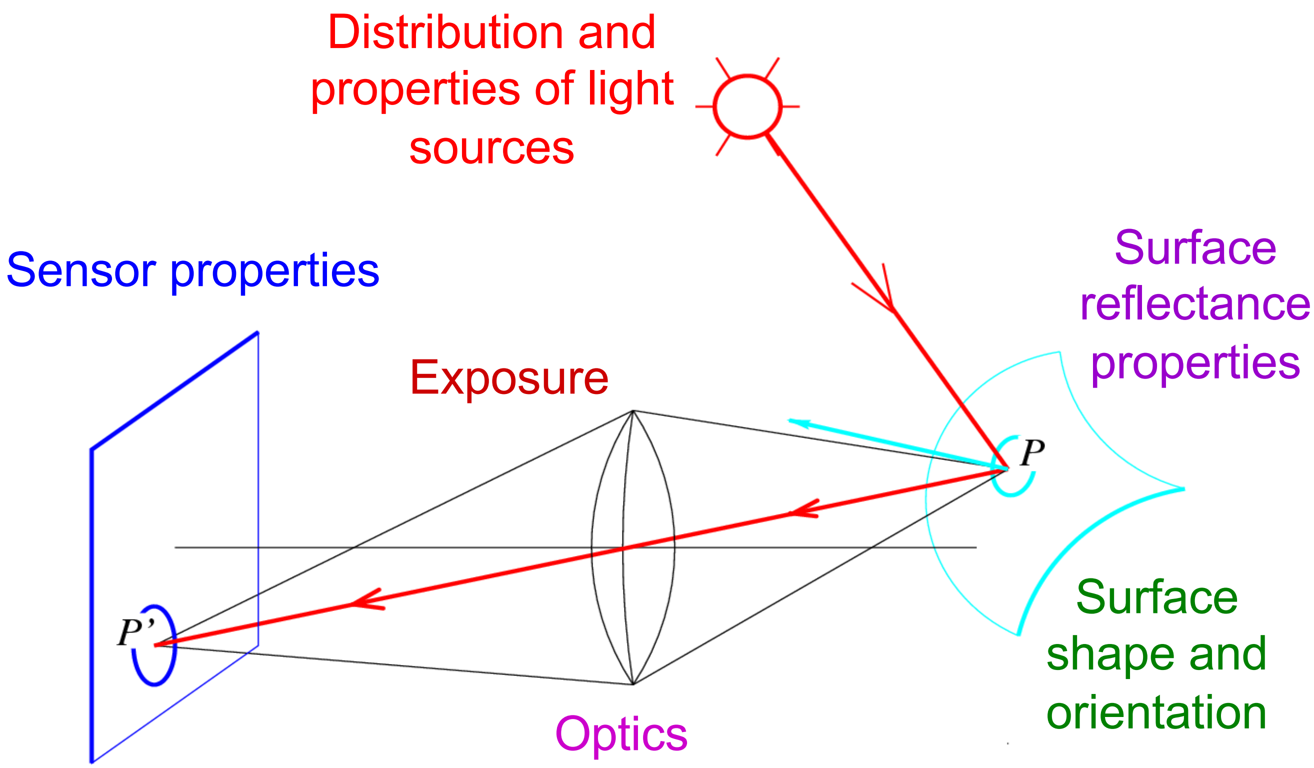
Figure source: P. Debevec and J. Malik. Recovering High Dynamic Range Radiance Maps from Photographs. SIGGRAPH 1997

Outline

- Small taste of radiometry
- In-camera transformation of light
- Reflectance properties of surfaces

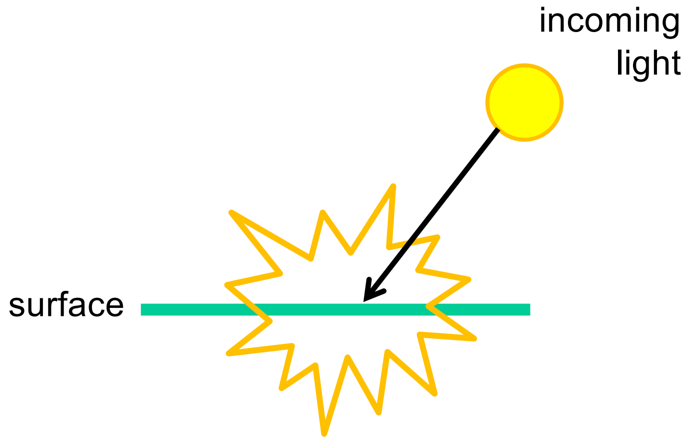
Recall: Image formation

- What determines the brightness of an image pixel?



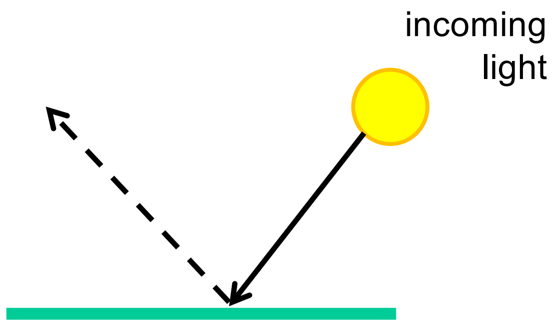
Slide by L. Fei-Fei

What can happen to light when it hits a surface?

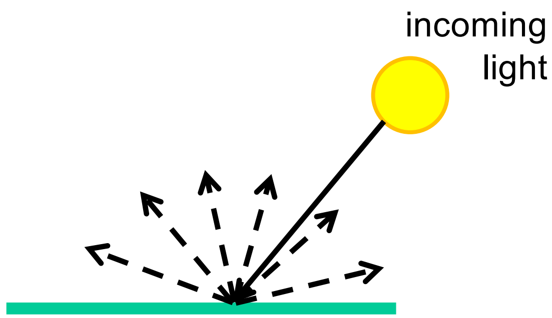


Basic models of reflection

- **Specular reflection:** light is reflected about the surface normal



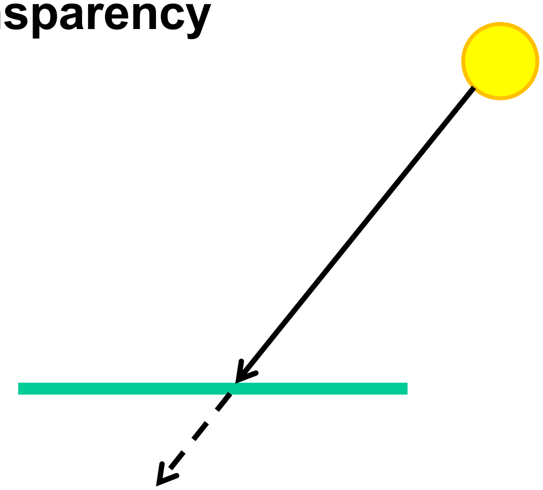
- **Diffuse reflection:** light scatters equally in all directions



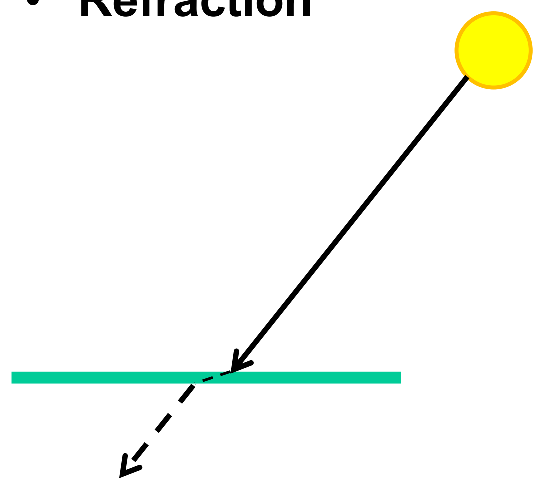
Slide from D. Hoiem

Other possible effects

- **Transparency**



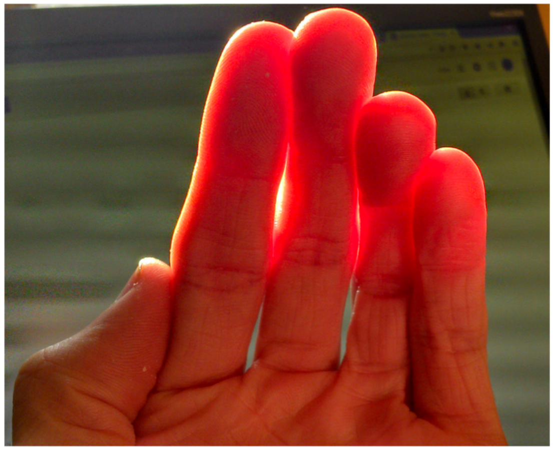
- **Refraction**

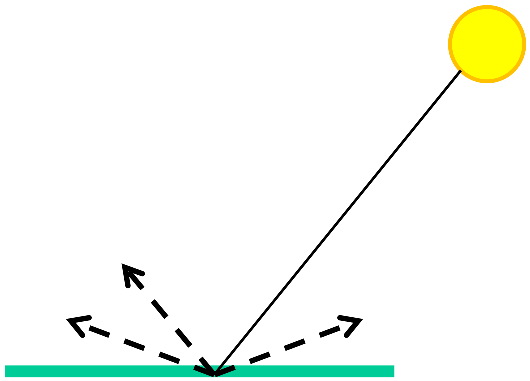


Slide from D. Hoiem

Other possible effects

- **Subsurface scattering**



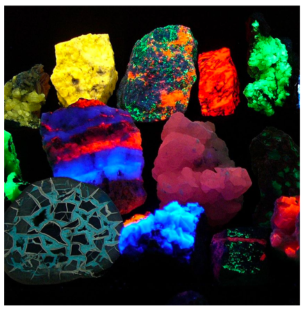


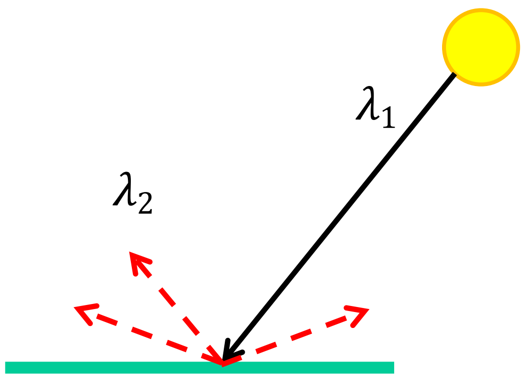
Slide from D. Hoiem

Image source

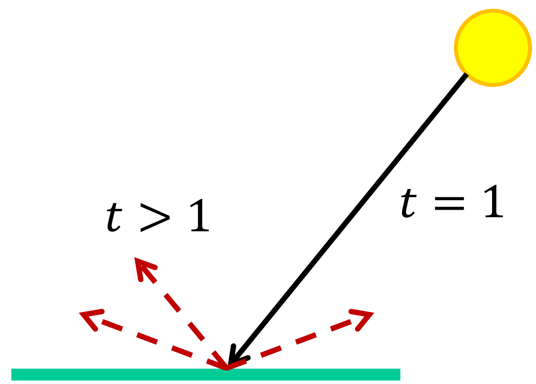
Other possible effects

- **Fluorescence**





- **Phosphorescence**

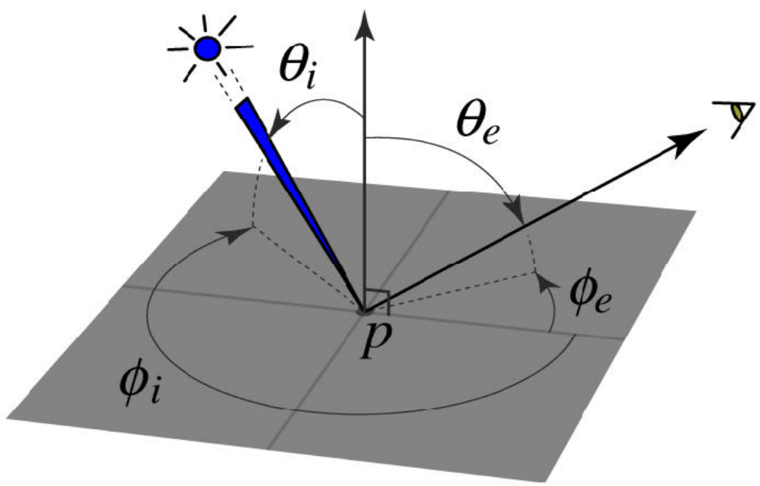


Slide from D. Hoiem

Image source

Bidirectional reflectance distribution function (BRDF)

- How bright a surface appears when viewed from one direction when light falls on it from another
- Definition: ratio of the radiance in the emitted direction to irradiance in the incident direction



Function of (at least) four parameters: incident and outgoing θ, ϕ

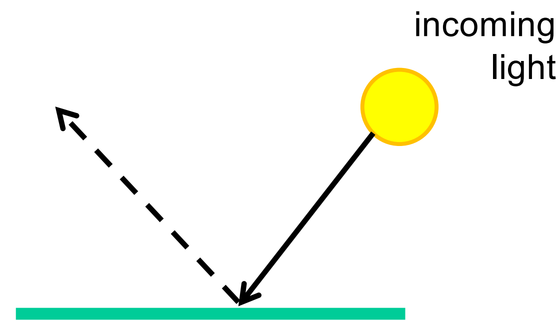
Source: Steve Seitz

Bidirectional reflectance distribution function (BRDF)

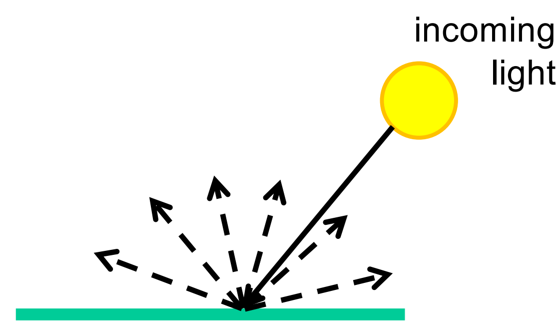
- How bright a surface appears when viewed from one direction when light falls on it from another
- Definition: ratio of the radiance in the emitted direction to irradiance in the incident direction
- Can be incredibly complicated!

Basic models of reflection in detail

- **Specular reflection:** light is reflected about the surface normal



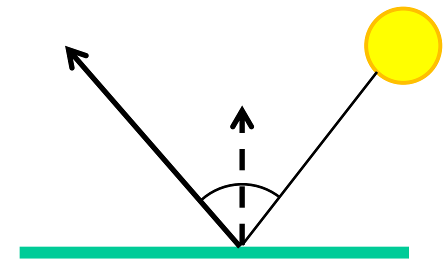
- **Diffuse reflection:** light scatters equally in all directions



Slide from D. Hoiem

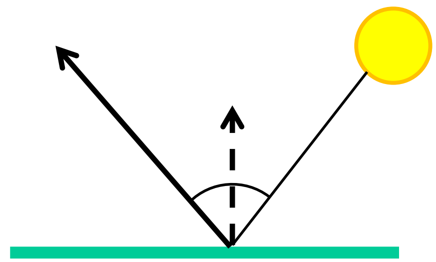
Specular reflection

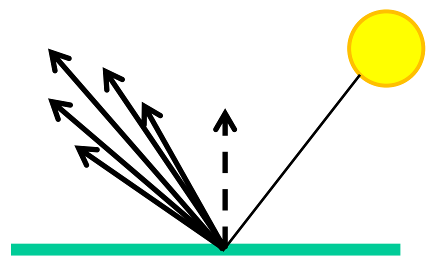
- Radiation arriving along a source direction leaves along the **specular direction** (source direction reflected about normal)



Specular reflection

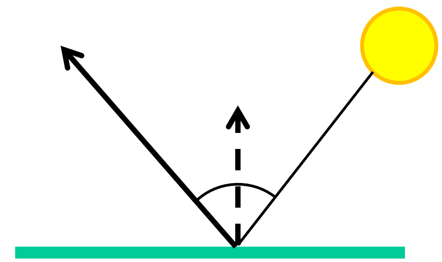
- Radiation arriving along a source direction leaves along the **specular direction** (source direction reflected about normal)
- On real surfaces, energy usually goes into a "lobe" of directions

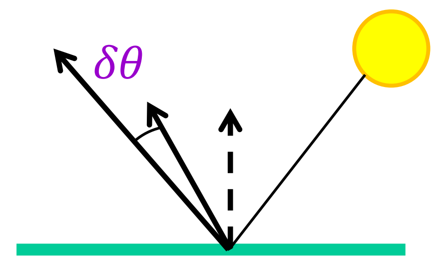




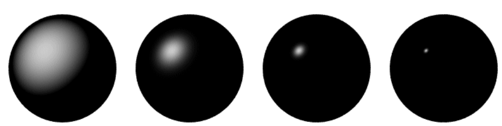
Specular reflection

- Radiation arriving along a source direction leaves along the **specular direction** (source direction reflected about normal)
- On real surfaces, energy usually goes into a “lobe” of directions
- **Phong model:** reflected energy falls of with $\cos^n(\delta\theta)$

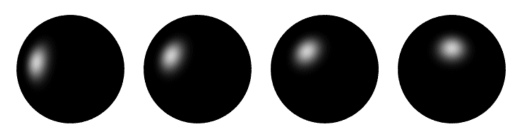




Changing the exponent

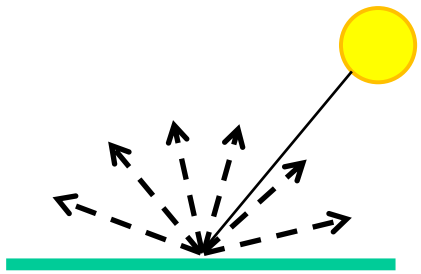


Moving the light source



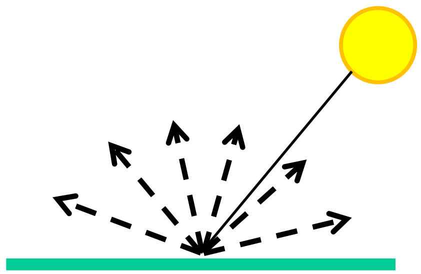
Diffuse reflection

- Light scatters equally in all directions
  - E.g., brick, matte plastic, rough wood

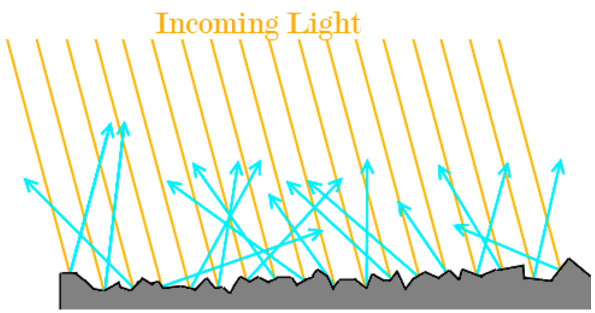


Diffuse reflection

- Light scatters equally in all directions
  - E.g., brick, matte plastic, rough wood



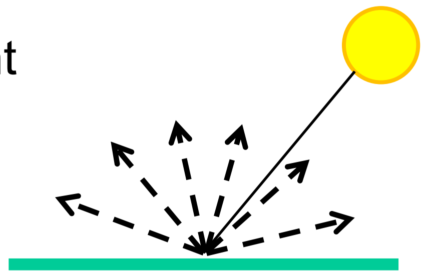
- This happens because of *microfacets* that scatter incoming light randomly



[Image source](#)

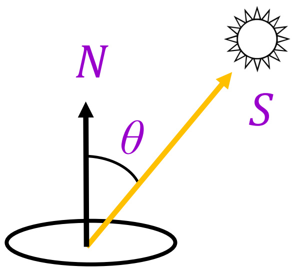
Diffuse reflection

- Light scatters equally in all directions
  - For a fixed incidence angle, BRDF is constant

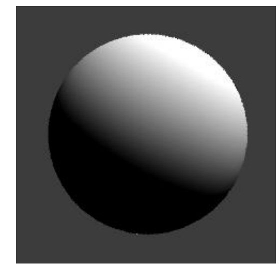


- What if we change the incidence angle?

Diffuse reflection: Lambert's law



$$I = \rho\,(S \cdot N) \\ = \rho \|S\| \cos\theta$$



I: reflected intensity (technically: *radiosity*, or total power leaving the surface per unit area)

ρ: albedo (fraction of incident irradiance reflected by the surface)

S: direction of light source (magnitude proportional to intensity of the source)

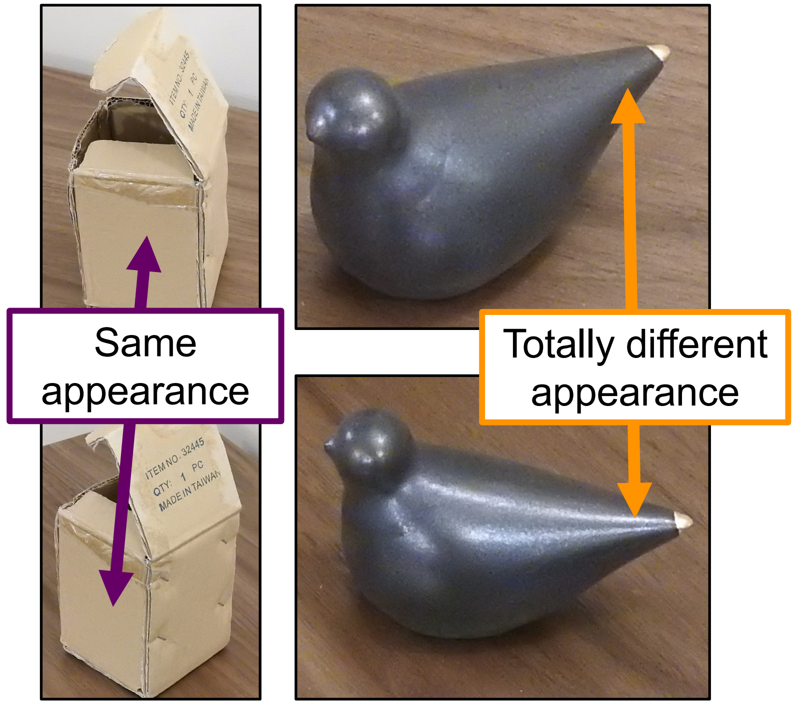
N: unit surface normal

Diffuse vs. specular: Significance for vision applications

Same lighting, as close as possible camera settings, but different **camera position**

Diffuse

Specular



Source: [J. Johnson and D. Fouhey]

Outline

- Small taste of radiometry
- In-camera transformation of light
- Reflectance properties of surfaces
- Diffuse and specular reflection
- Shape from shading

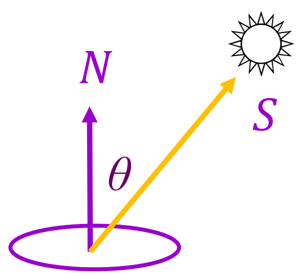
Photometric stereo, or shape from shading

- Can we reconstruct the shape of an object based on shading cues?

Luca della Robbia, *Cantoria*, 1438

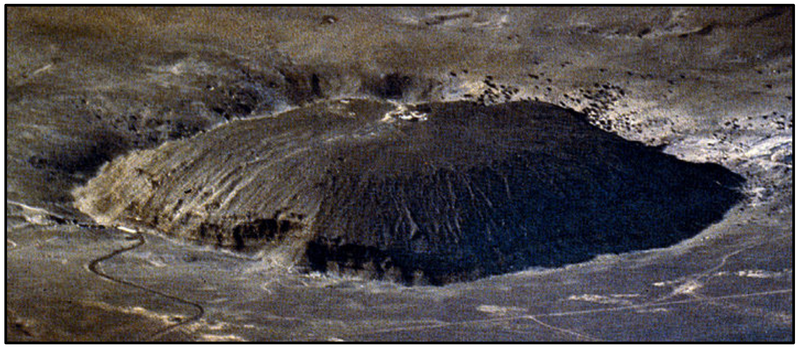
Photometric stereo, or shape from shading

- Can we reconstruct the shape of an object based on shading cues?
- Assuming a Lambertian object, given the image intensity (I), can we recover the light source direction (S) and the surface normal (N)?
- Can we do this from a single image?



$$
\begin{aligned}
I &= \rho\,(S \cdot N) \\
&= \rho \|S\| \cos\theta
\end{aligned}
$$

Shape from shading ambiguity



Source: J. Johnson and D. Fouhey

Image source

Shape from shading ambiguity

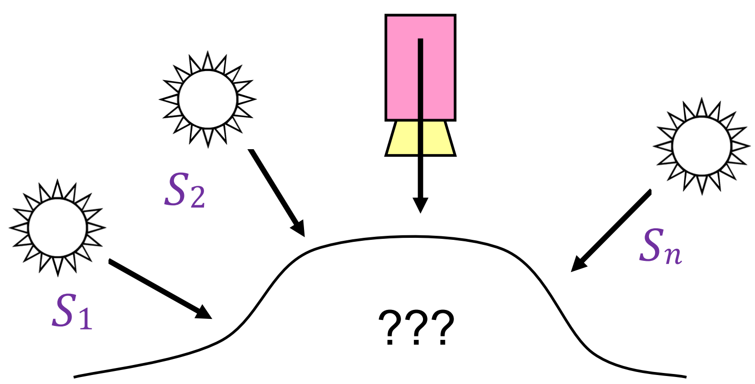
- Humans assume light from above (and the blueness also tells you distance)

Source: [J. Johnson and D. Fouhey]

[Image source]

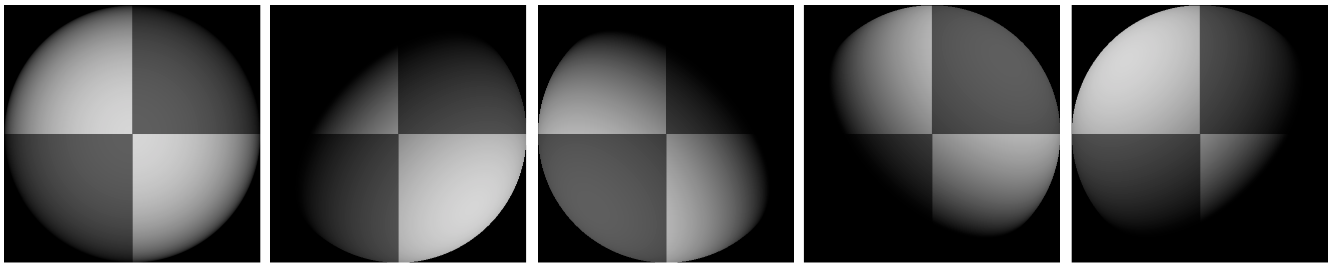
Photometric stereo

- Assume:
  - A Lambertian object
  - A *local shading model* (each point on a surface receives light only from sources visible at that point)
  - A set of *known* light source directions
  - A set of pictures of an object, obtained in exactly the same camera/object configuration but using different sources
  - Orthographic projection
- Goal: reconstruct object shape and albedo

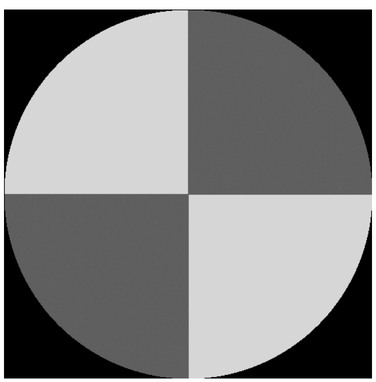


F&P 2nd ed., sec. 2.2.4

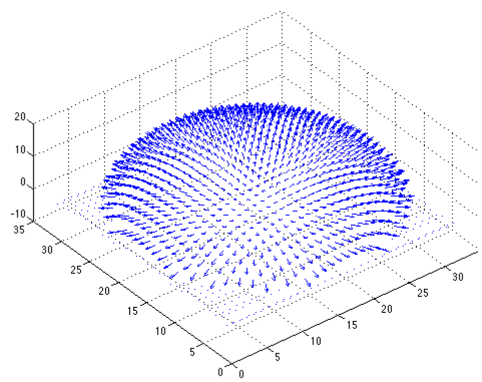
Example 1



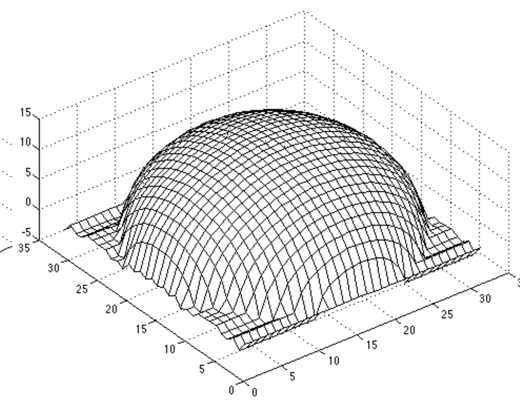
Recovered albedo



Recovered normal field

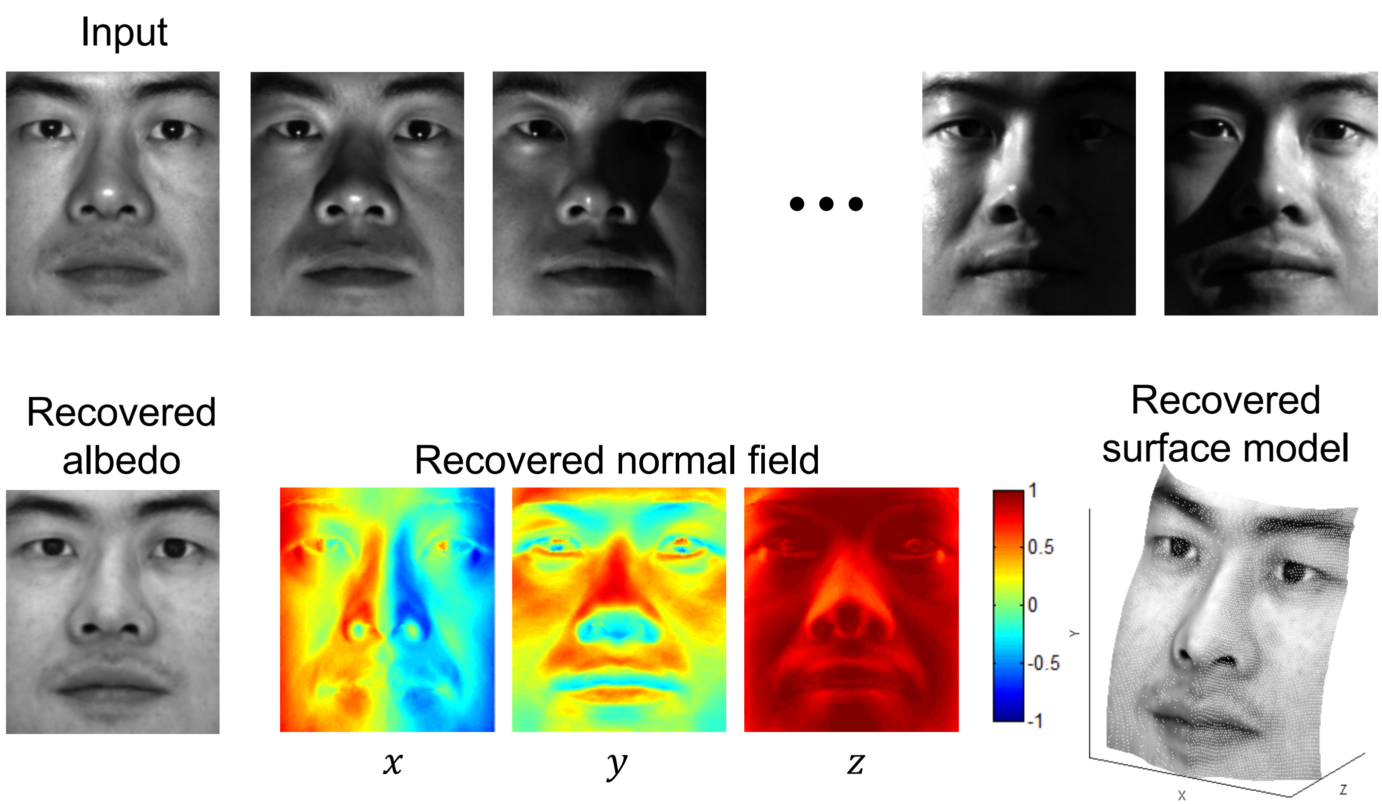


Recovered surface model



F&P 2nd ed., sec. 2.2.4

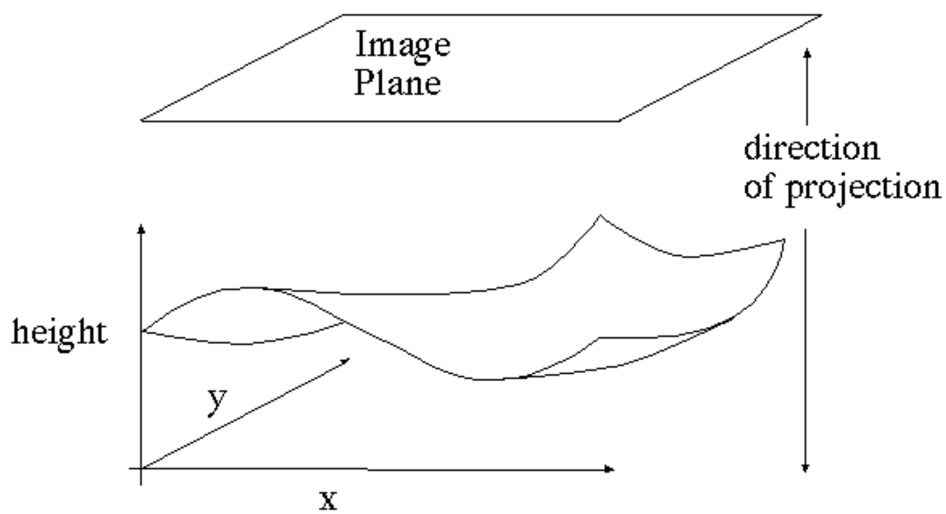
Example 2



Input
• • •
Recovered albedo
Recovered normal field
Recovered surface model
1
0.5
0
-0.5
-1
x
y
z
Y
X
Z

Image model

- **Known:** source vectors S_j and pixel values $I_j(x, y)$
- **Unknown:** surface normal $N(x, y)$ and albedo $\rho(x, y)$



F&P 2nd ed., sec. 2.2.4

Image model

- **Known:** source vectors S_j and pixel values $I_j(x,y)$
- **Unknown:** surface normal $N(x,y)$ and albedo $\rho(x,y)$
- Assume that the response function of the camera is a linear scaling by a factor of k
- Lambert's law:

$$
\begin{aligned}
I_j(x,y) &= k\,\rho(x,y)\big(N(x,y)\cdot S_j\big) \\
&= \big(\rho(x,y)N(x,y)\big)\cdot\big(k\,S_j\big) \\
&= g(x,y)\cdot V_j
\end{aligned}
$$

F&P 2nd ed., sec. 2.2.4

Least squares problem

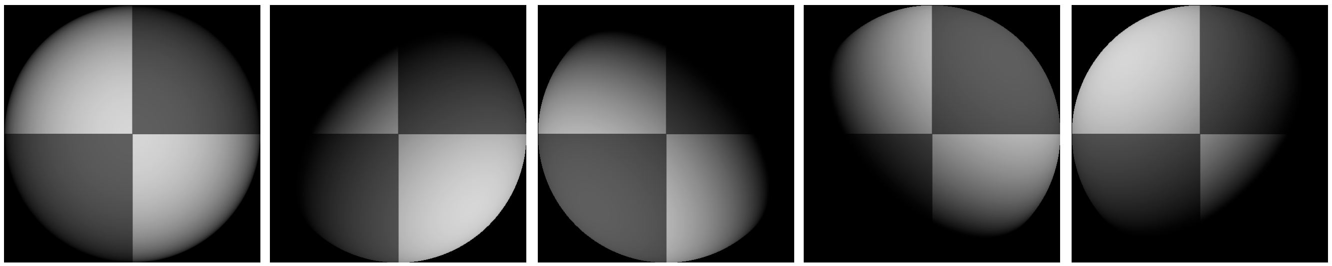
- For each pixel, set up a linear system:

$$\underbrace{\begin{bmatrix} V_1^T \\ V_2^T \\ \vdots \\ V_n^T \end{bmatrix}}_{\substack{n \times 3 \\ \text{known}}} \underbrace{g(x,y)}_{\substack{3 \times 1 \\ \text{unknown}}} = \underbrace{\begin{bmatrix} I_1(x,y) \\ I_2(x,y) \\ \vdots \\ I_n(x,y) \end{bmatrix}}_{\substack{n \times 1 \\ \text{known}}}$$

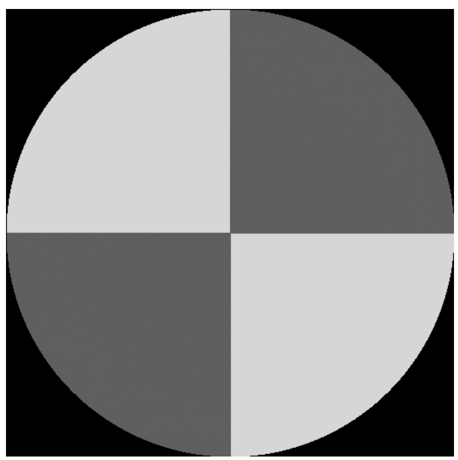
- Obtain least-squares solution for $g(x,y)$, which we defined as $\rho(x,y)N(x,y)$
- Since $N(x,y)$ is the *unit* normal, $\rho(x,y)$ is given by the magnitude of $g(x,y)$
- Finally, $N(x,y) = \frac{1}{\rho(x,y)} g(x,y)$

F&P 2nd ed., sec. 2.2.4

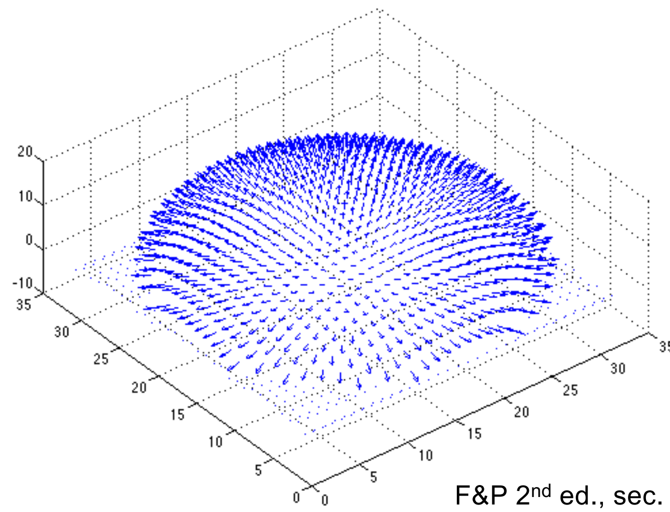
Synthetic example



Recovered albedo



Recovered normal field



F&P 2nd ed., sec. 2.2.4

Recovering a surface from normals

- Recall: the surface is written as

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

- This means the unit normal has the following form:

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

- Write the estimated vector g as

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

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

$$f_x(x, y) = \frac{g_1(x, y)}{g_3(x, y)}$$

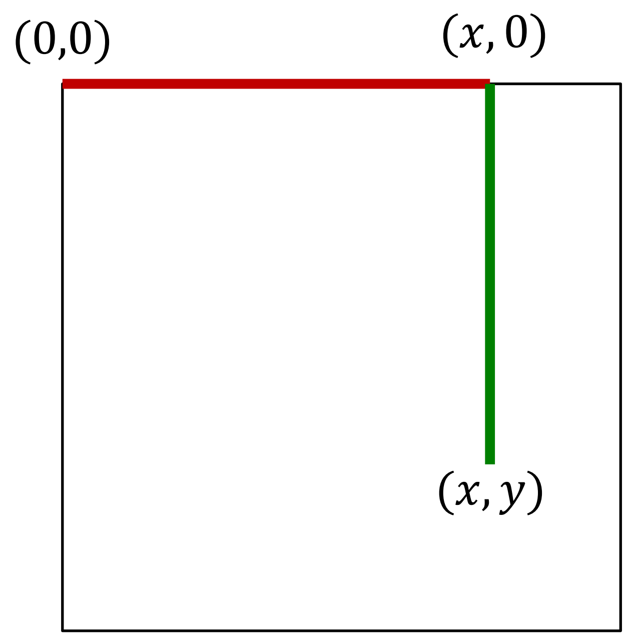
$$f_y(x, y) = \frac{g_2(x, y)}{g_3(x, y)}$$

F&P 2nd ed., sec. 2.2.4

Recovering a surface from normals

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

$$f(x, y) =$$
$$\int_0^x f_x(s, 0)ds + \int_0^y f_y(x, t)dt + C$$



- For robustness, it is better to take integrals over many different paths and average the results

F&P 2nd ed., sec. 2.2.4

Recovering a surface from normals

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

$$f(x, y) = \int_0^x f_x(s, 0)ds + \int_0^y f_y(x, t)dt + C$$

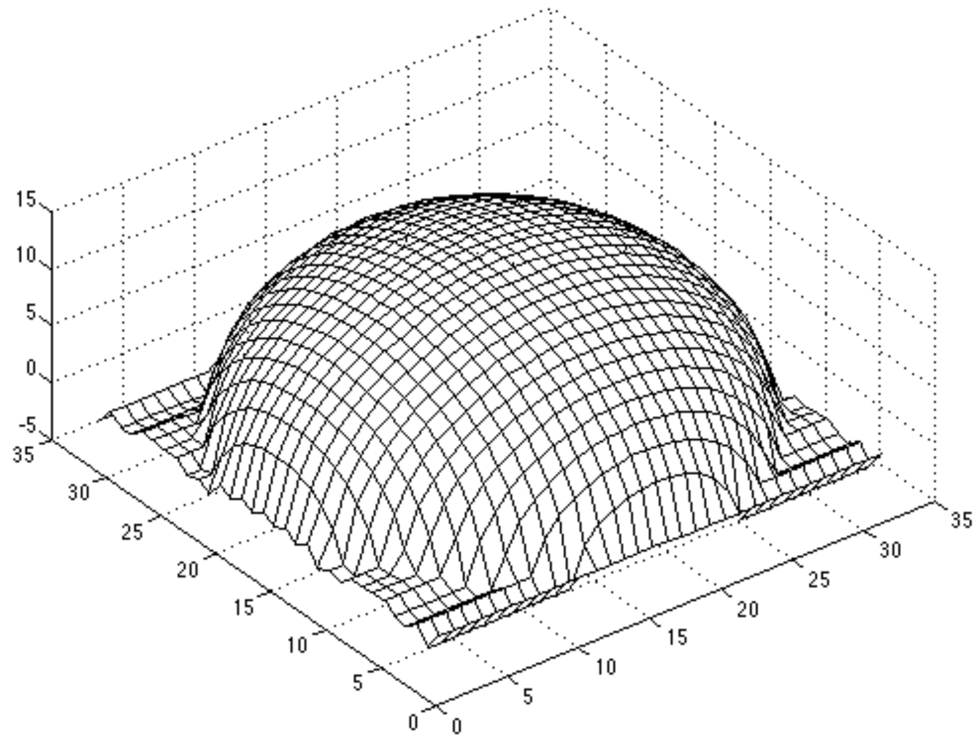
- For robustness, it is better to take integrals over many different paths and average the results

- Note: *integrability* must be satisfied: for the surface f to exist, the mixed second partial derivatives must be equal (or at least similar in practice):

$$\frac{\partial}{\partial y}\left(\frac{g_1(x, y)}{g_3(x, y)}\right) = \frac{\partial}{\partial x}\left(\frac{g_2(x, y)}{g_3(x, y)}\right)$$

F&P 2nd ed., sec. 2.2.4

Surface recovered by integration



F&P 2nd ed., sec. 2.2.4

Limitations of model

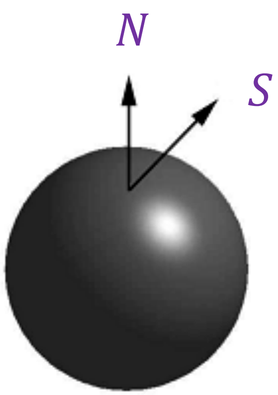
- Orthographic camera model
- Simplistic reflectance and lighting model
- No shadows
- No interreflections
- No missing data
- Integration is tricky

Outline

- Small taste of radiometry
- In-camera transformation of light
- Reflectance properties of surfaces
- Diffuse and specular reflection
- Shape from shading
- Estimating direction of light sources

Finding the direction of the light source

$$I(x,y) = N(x,y) \cdot S(x,y)$$



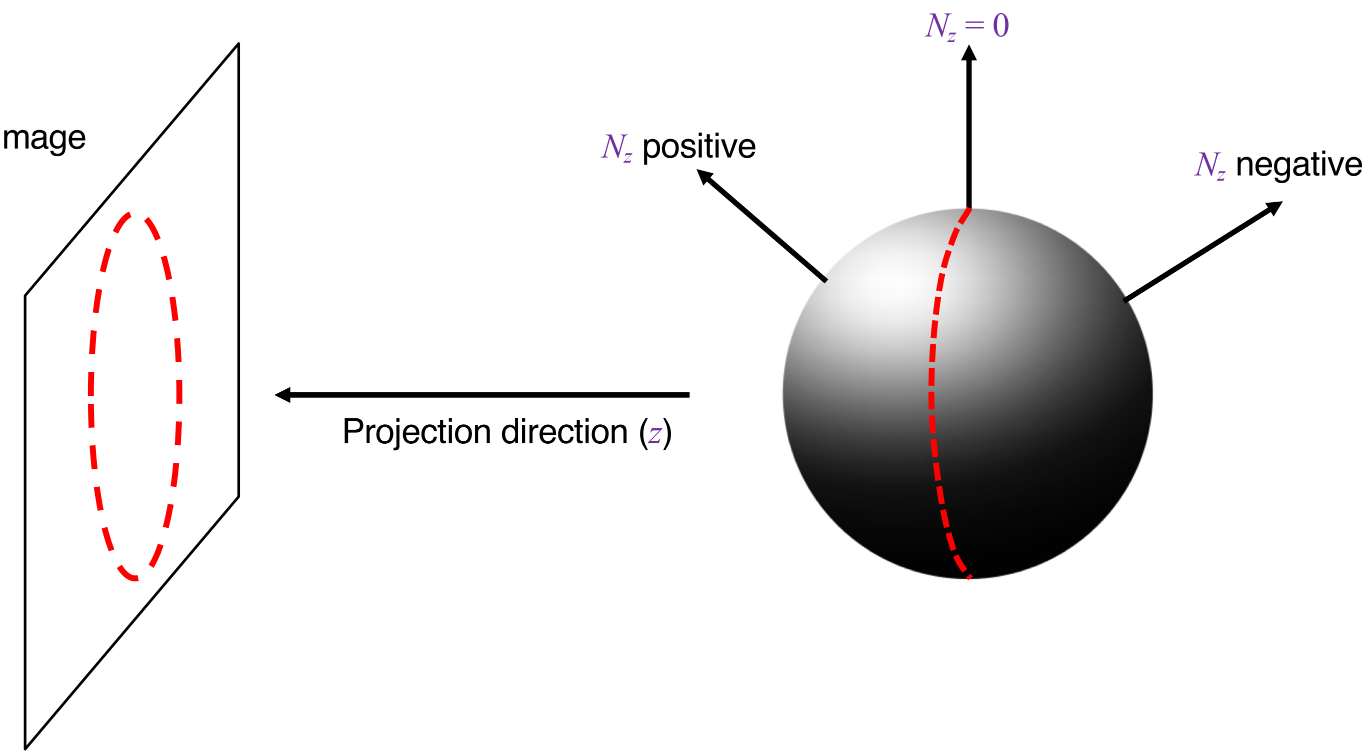
- Full 3D case:

$$\begin{bmatrix} N_x(x_1,y_1) & N_y(x_1,y_1) & N_z(x_1,y_1) \\ N_x(x_2,y_2) & N_y(x_2,y_2) & N_z(x_2,y_2) \\ \vdots & \vdots & \vdots \\ N_x(x_n,y_n) & N_y(x_n,y_n) & N_z(x_n,y_n) \end{bmatrix} \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} = \begin{bmatrix} I(x_1,y_1) \\ I(x_2,y_2) \\ \vdots \\ I(x_n,y_n) \end{bmatrix}$$

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

Finding the direction of the light source

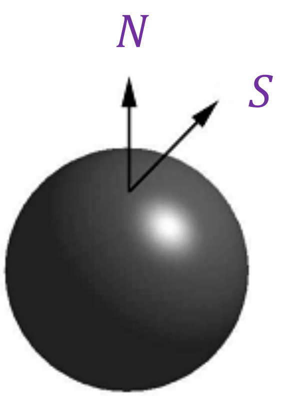
Consider points on the *occluding contour*:



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

Finding the direction of the light source

$$I(x,y) = N(x,y) \cdot S(x,y)$$



- Full 3D case:

$$\begin{bmatrix} N_x(x_1,y_1) & N_y(x_1,y_1) & N_z(x_1,y_1) \\ N_x(x_2,y_2) & N_y(x_2,y_2) & N_z(x_2,y_2) \\ \vdots & \vdots & \vdots \\ N_x(x_n,y_n) & N_y(x_n,y_n) & N_z(x_n,y_n) \end{bmatrix} \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} = \begin{bmatrix} I(x_1,y_1) \\ I(x_2,y_2) \\ \vdots \\ I(x_n,y_n) \end{bmatrix}$$

- For points on the occluding contour ($N_z = 0$):

$$\begin{bmatrix} N_x(x_1,y_1) & N_y(x_1,y_1) \\ N_x(x_2,y_2) & N_y(x_2,y_2) \\ \vdots & \vdots \\ N_x(x_n,y_n) & N_y(x_n,y_n) \end{bmatrix} \begin{bmatrix} S_x \\ S_y \end{bmatrix} = \begin{bmatrix} I(x_1,y_1) \\ I(x_2,y_2) \\ \vdots \\ I(x_n,y_n) \end{bmatrix}$$

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

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

Real photo

Fake photo

M. K. Johnson and H. Farid. Exposing Digital Forgeries by Detecting Inconsistencies in Lighting.
ACM Multimedia and Security Workshop, 2005