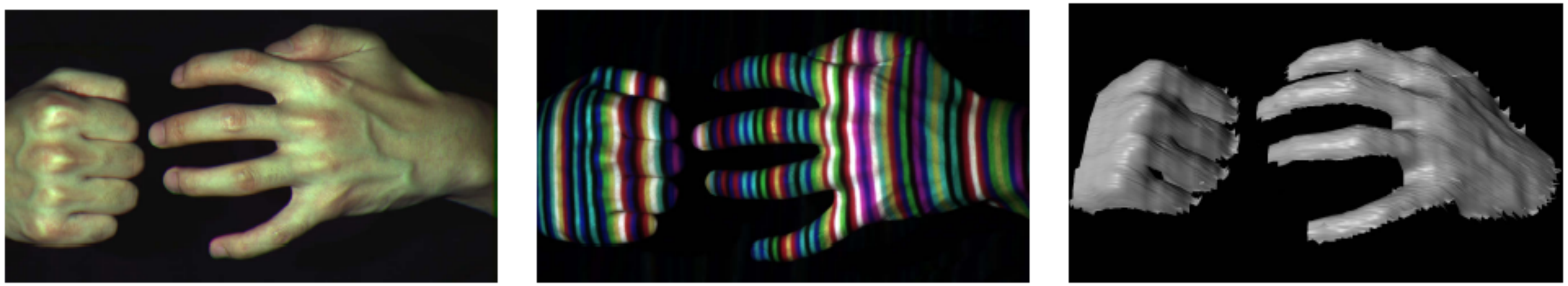
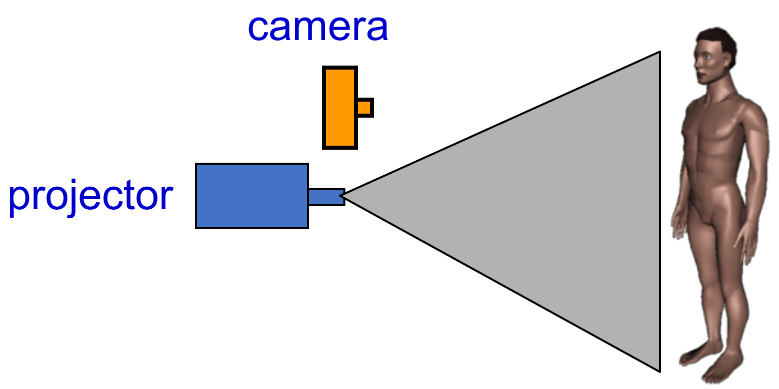
Other approaches to obtaining 3D structure

Active stereo with structured light

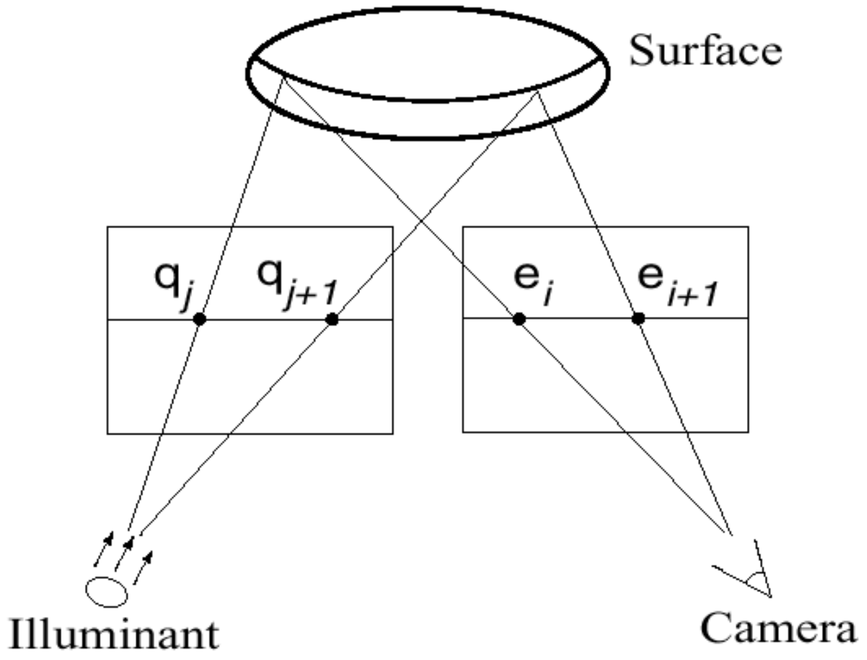


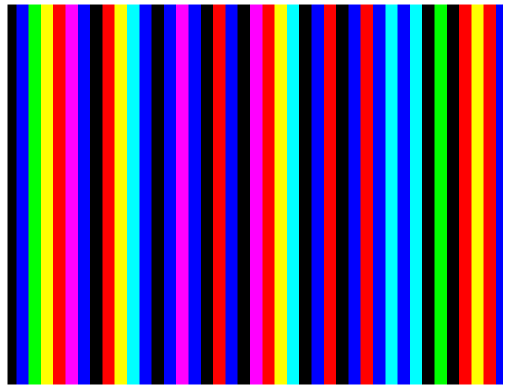
- Project “structured” light patterns onto the object
  - simplifies the correspondence problem
  - Allows us to use only one camera

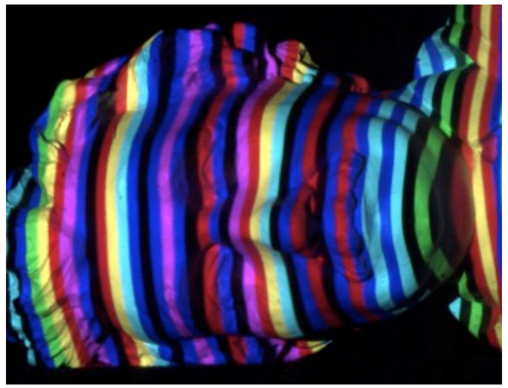


L. Zhang, B. Curless, and S. M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. *3DPVT* 2002

Active stereo with structured light

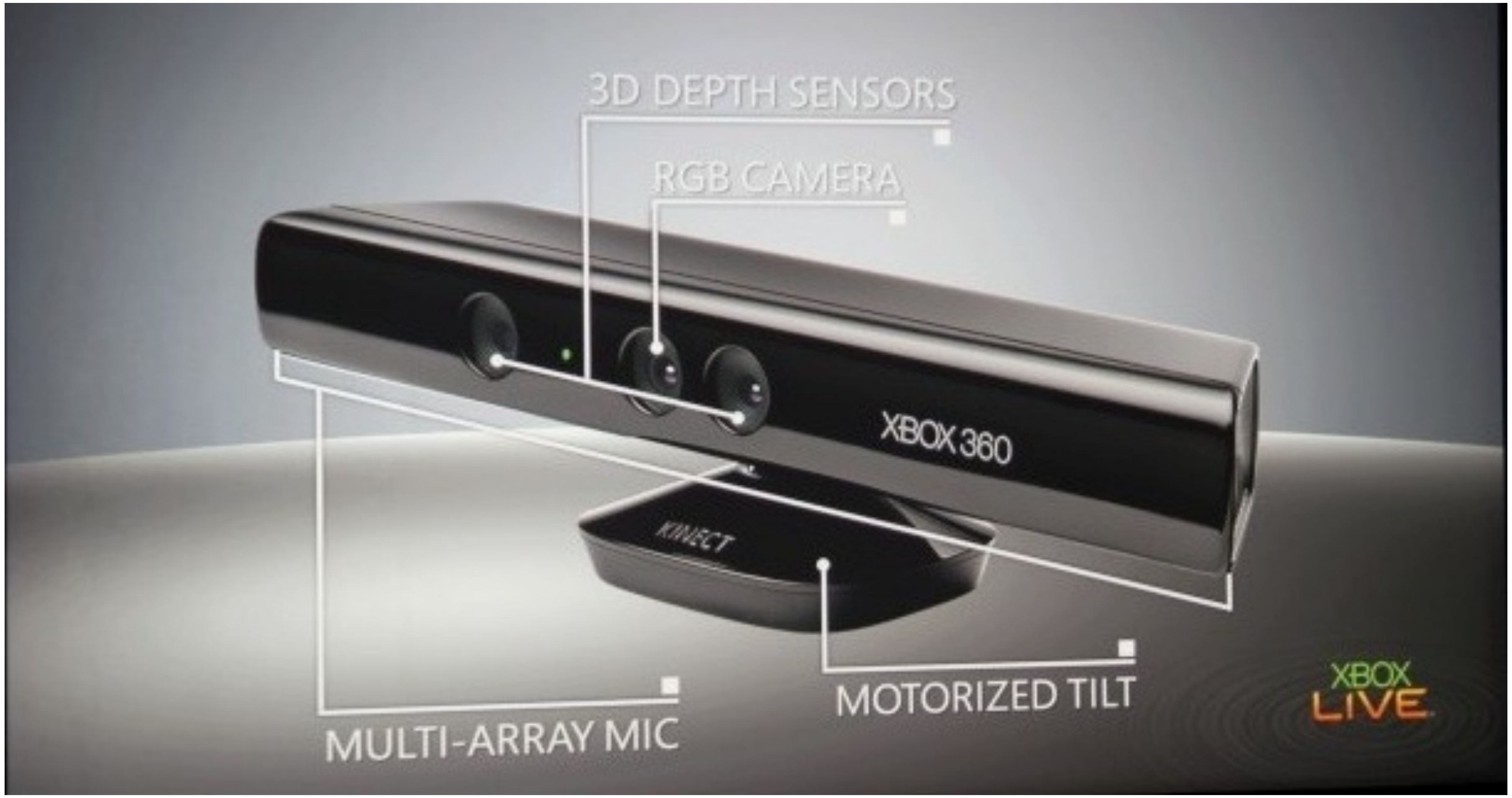






L. Zhang, B. Curless, and S. M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. *3DPVT* 2002

Microsoft Kinect



Light and geometry

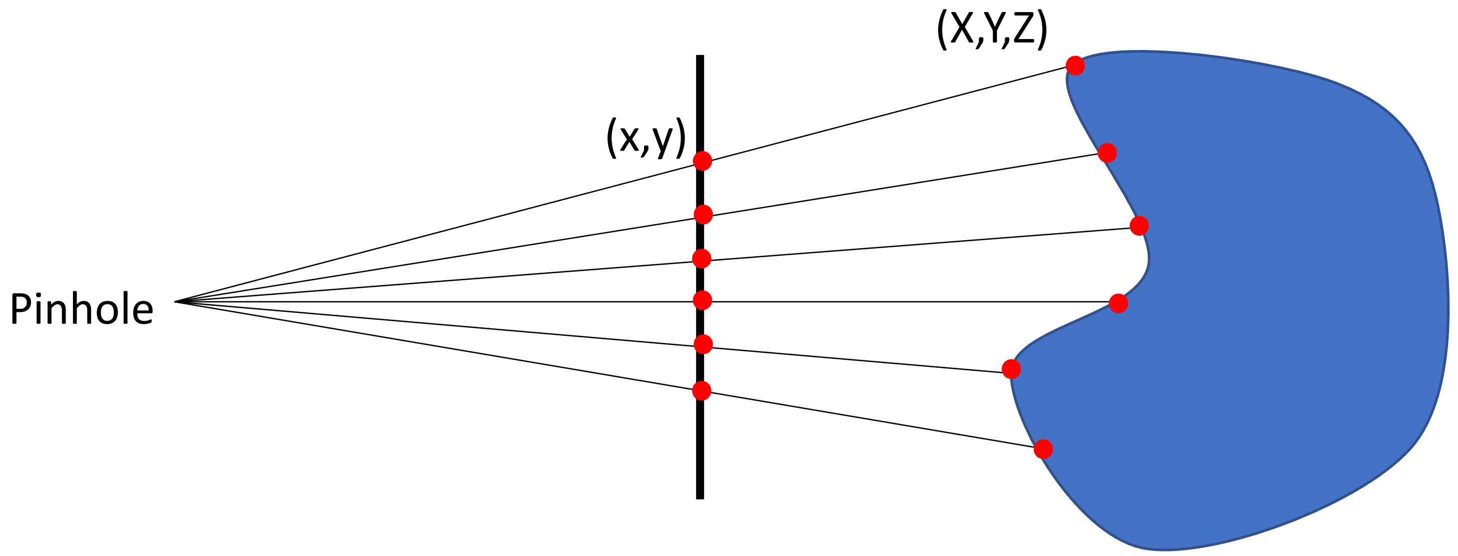
Till now: 3D structure from multiple cameras

- Problems:
  - requires calibrated cameras
  - requires correspondence
- Other cues to 3D structure?

Key Idea: use feature motion to understand shape

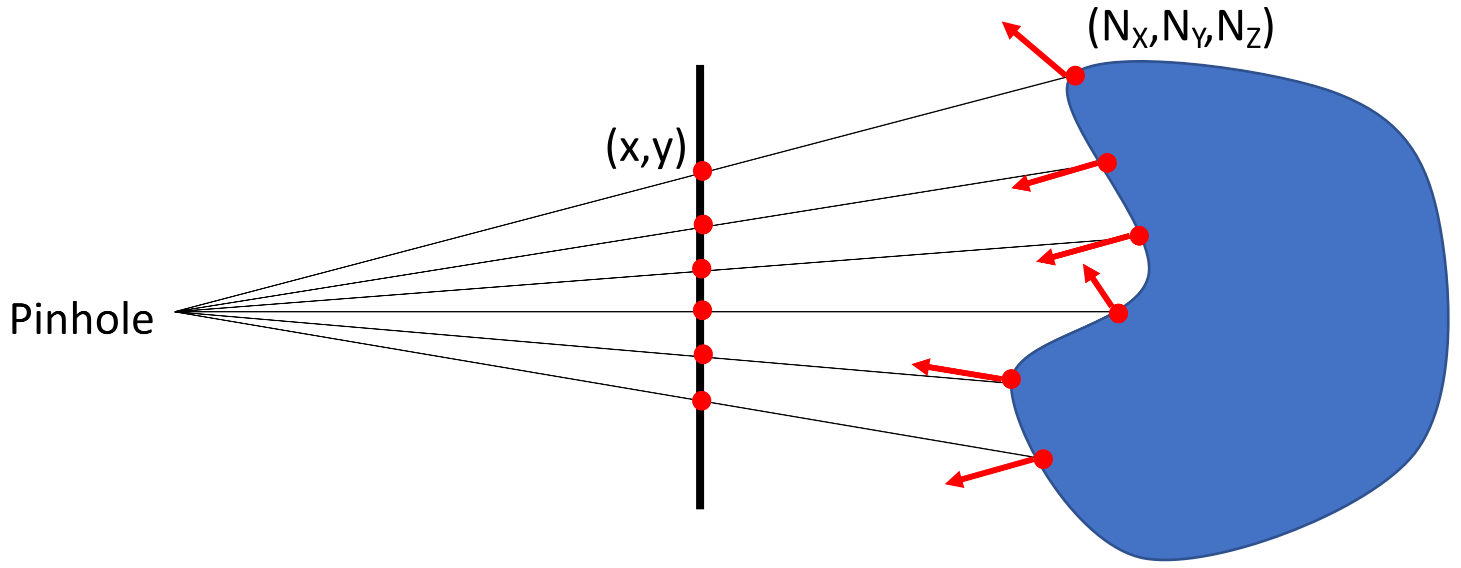
What does 3D structure mean?

- We have been talking about the depth of a pixel



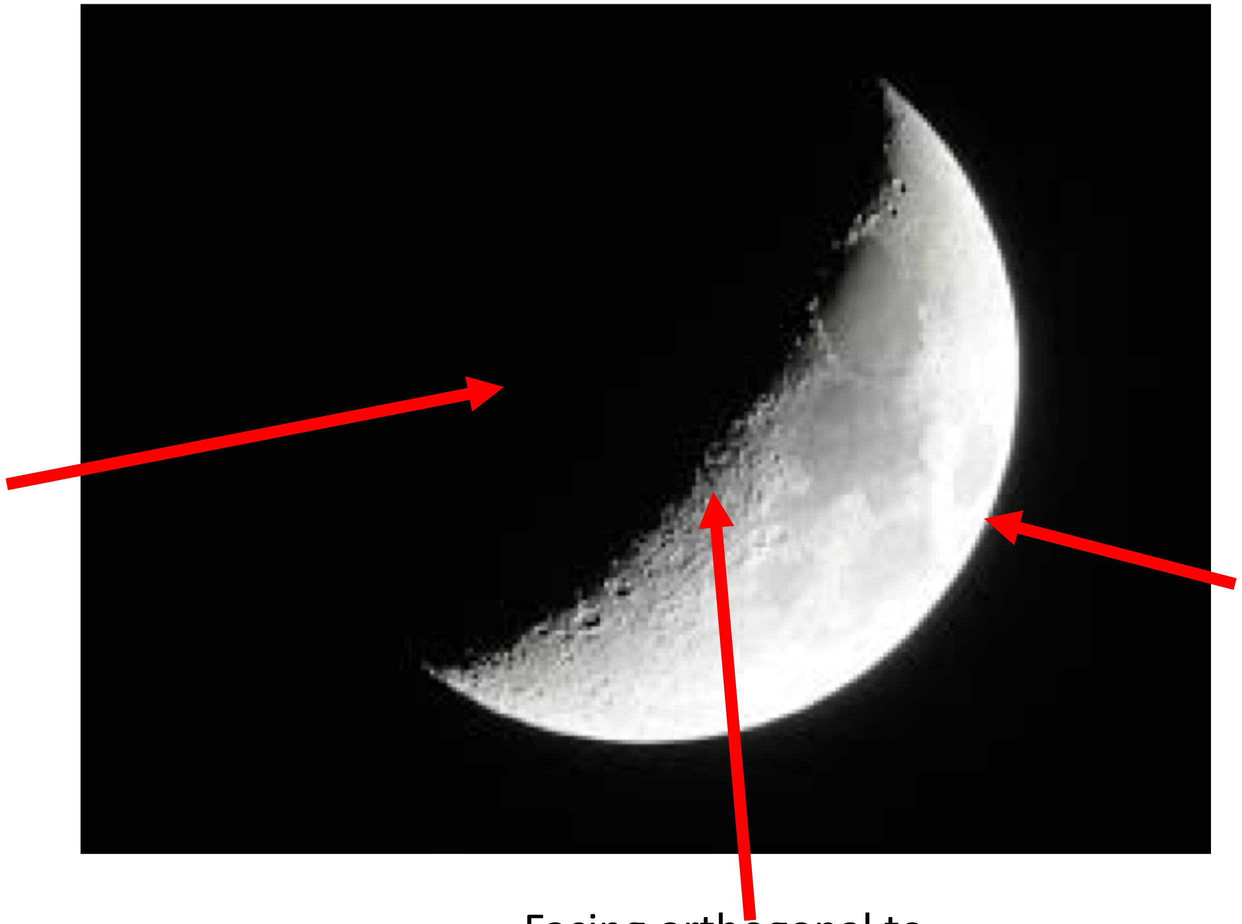
What does 3D structure mean?

- But we can also look at the orientation of the surface at each pixel: *surface normal*



Not enough by itself to reveal absolute locations, but gives enough of a clue to object shape

Shading is a cue to surface orientation


Facing away from the sun, hence dark – "shadow"
Facing the sun, hence bright
Facing orthogonal to the sun, hence dark

Shading is a cue to surface orientation

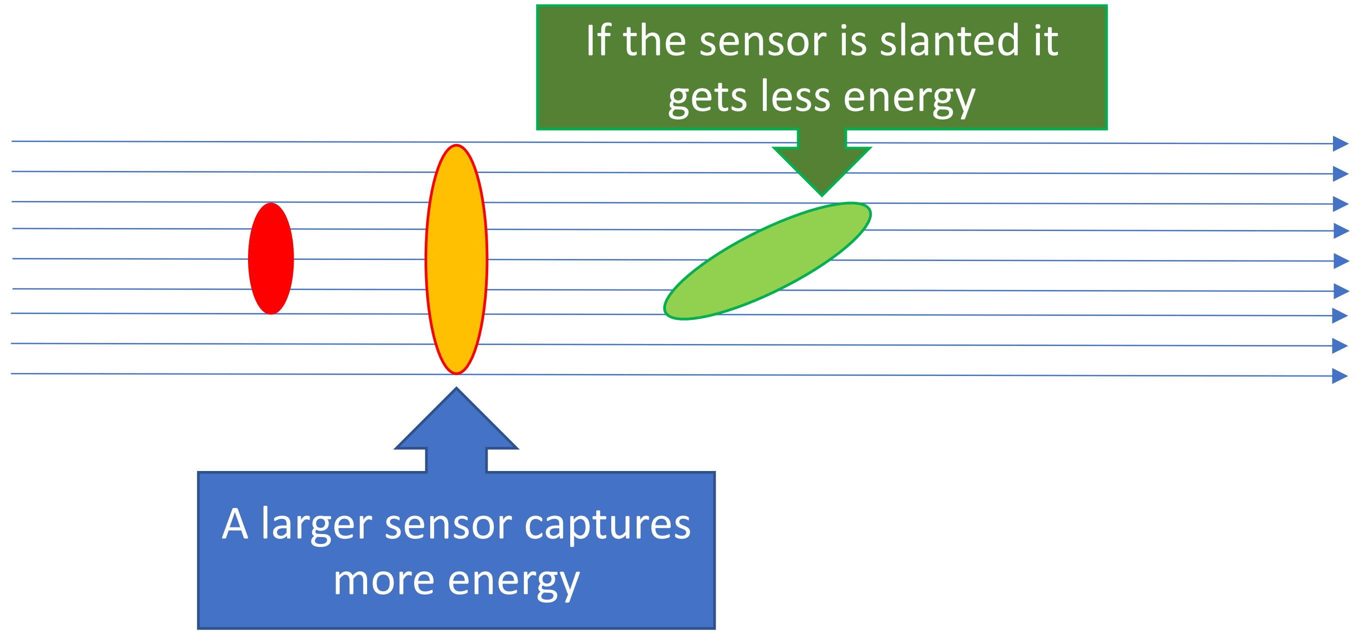
- Till now we have looked at *where a pixel comes from*
- Now: *what is its color?*
- Depends on:
  - Color and amount of lighting
  - Orientation of surface relative to lighting
  - Paint on the surface

How does light interact with the scene?

- Light is a bunch of photons
- Photons are energy packets
- Light starts from the light source, is reflected / absorbed by surfaces and lands on the camera
- Two key questions:
  - What property of light does a camera pixel record? *Radiance*
  - How does the radiance of a pixel depend on lighting, shape and paint?

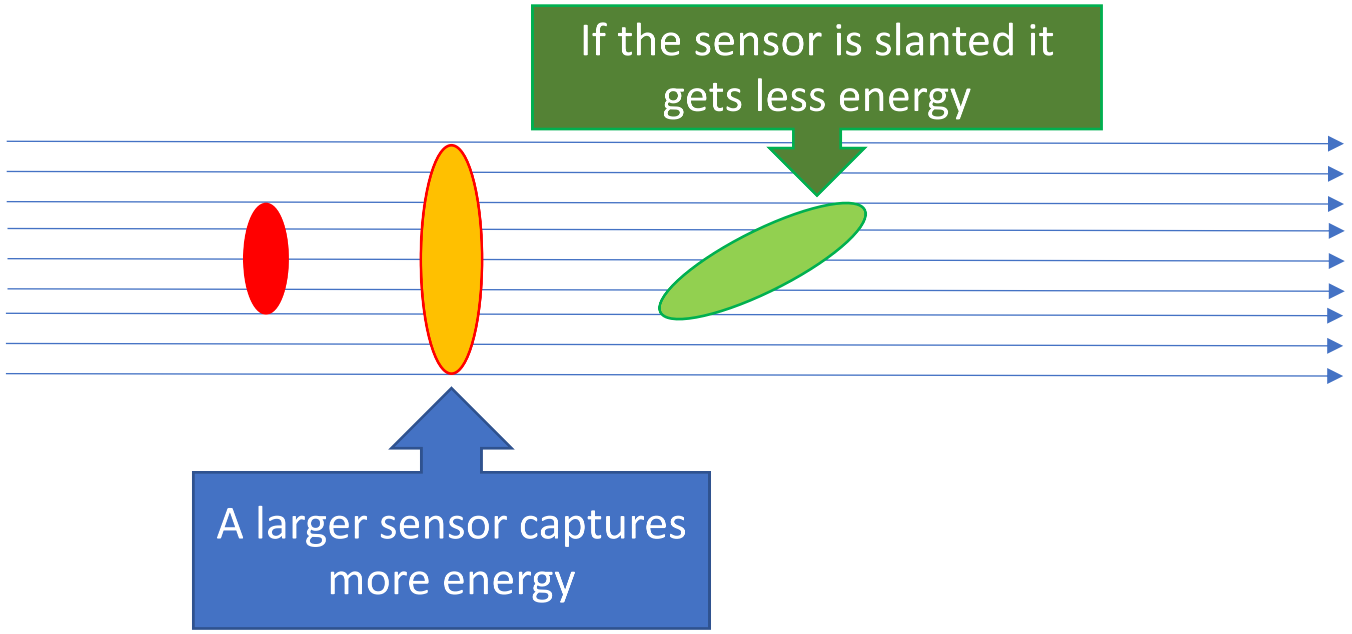
Radiance

- How do we measure the “strength” of a beam of light?
- Idea: put a sensor and see how much energy it gets


If the sensor is slanted it gets less energy
A larger sensor captures more energy

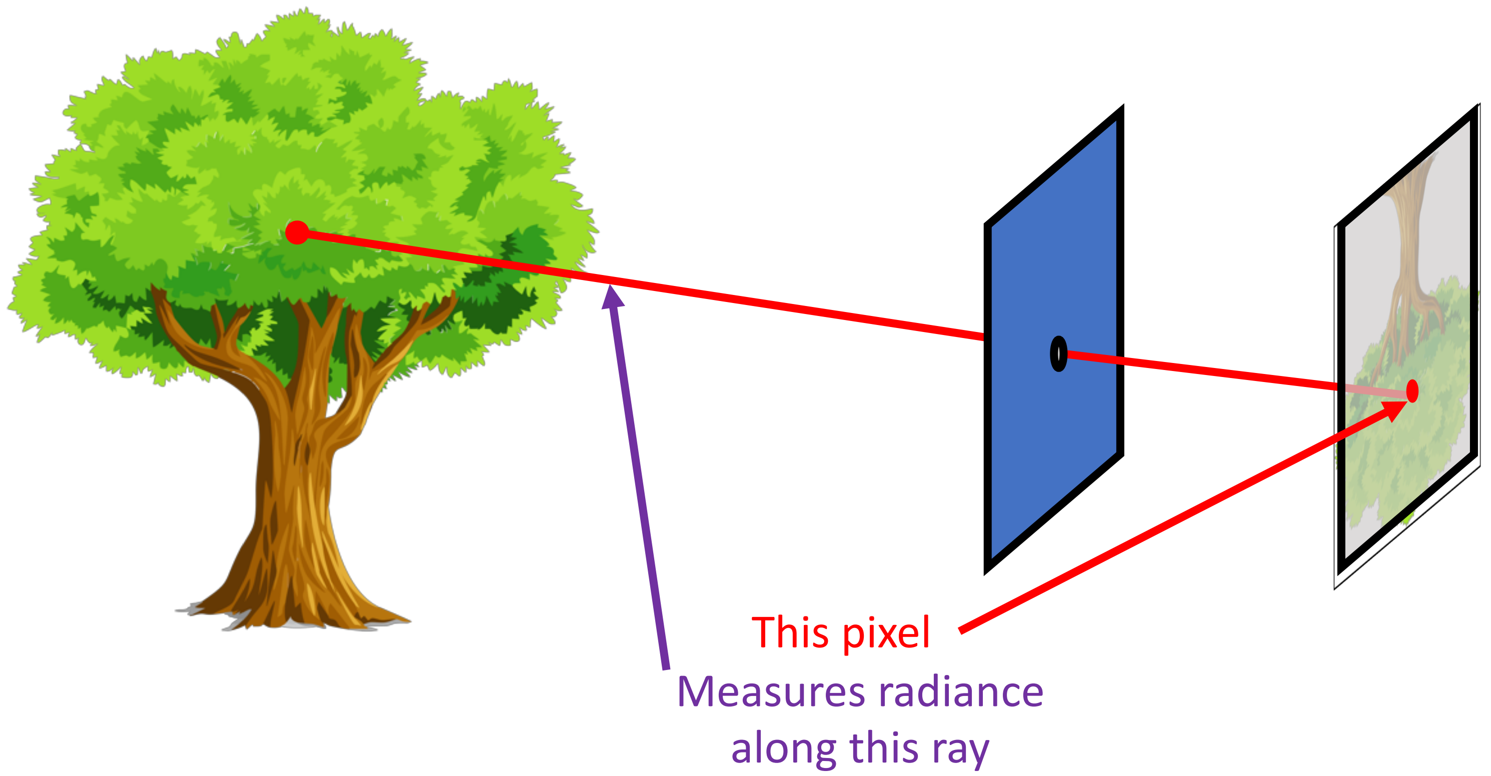
Radiance

- How do we measure the “strength” of a beam of light?
- Radiance: power *in a particular direction* per unit area when surface is orthogonal to direction


If the sensor is slanted it gets less energy
A larger sensor captures more energy

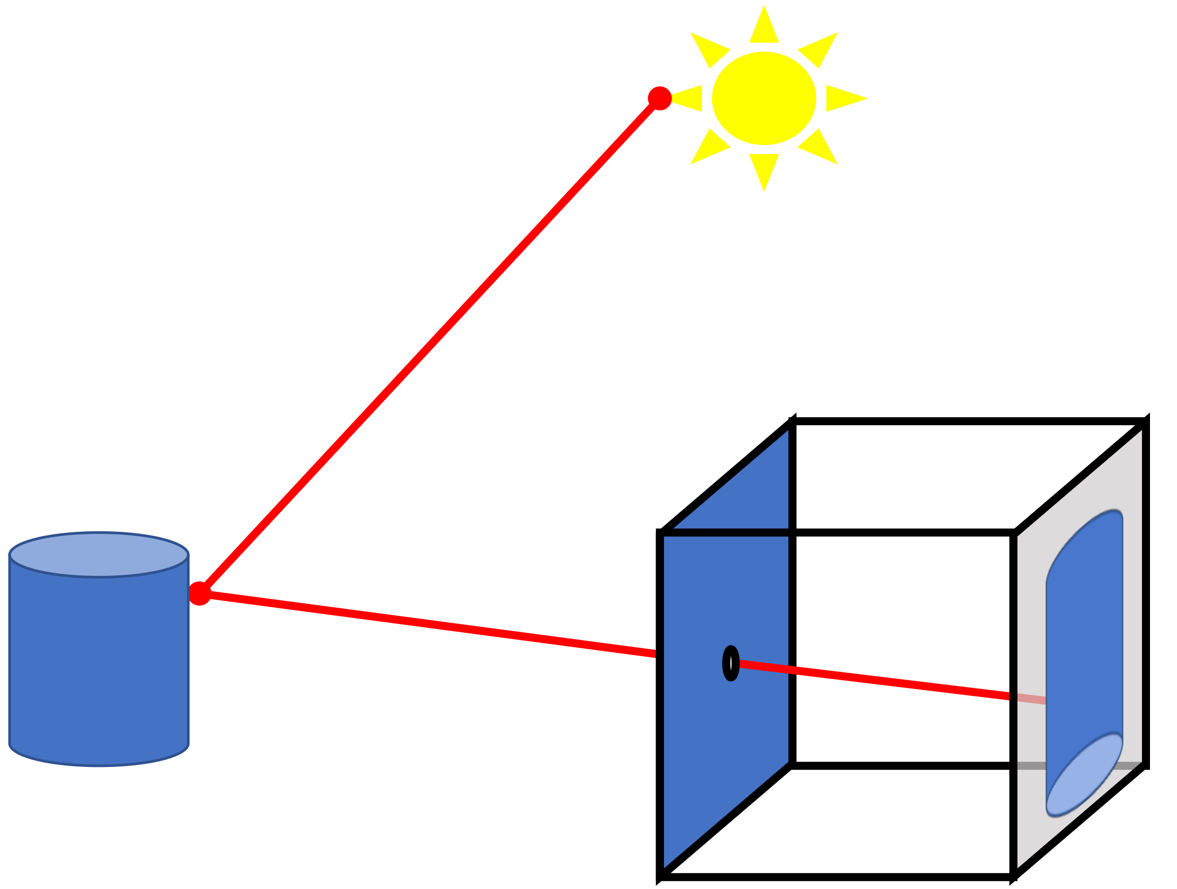
Radiance

- Pixels measure radiance



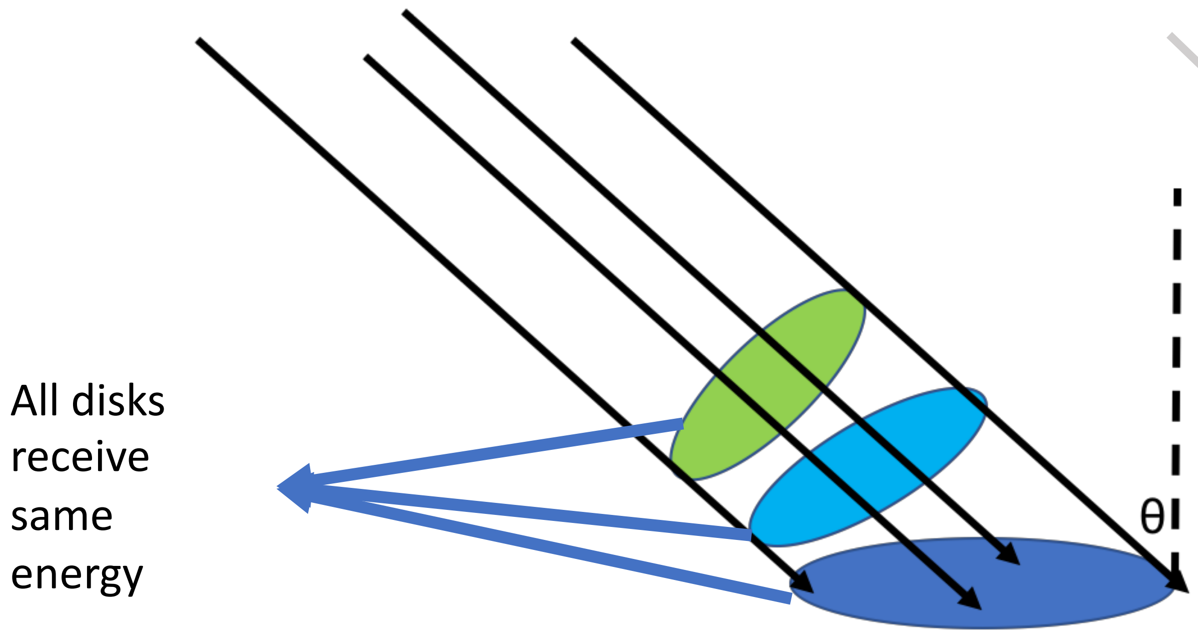
Where do the rays come from?

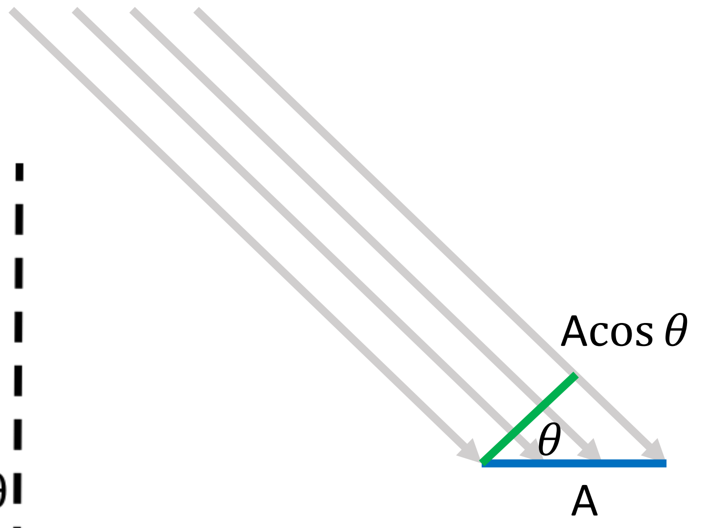
- Rays from the light source “reflect” off a surface and reach camera
- Surface gets some energy from the light source: *irradiance*
- Depending on paint, some of this energy is reflected back



Irradiance

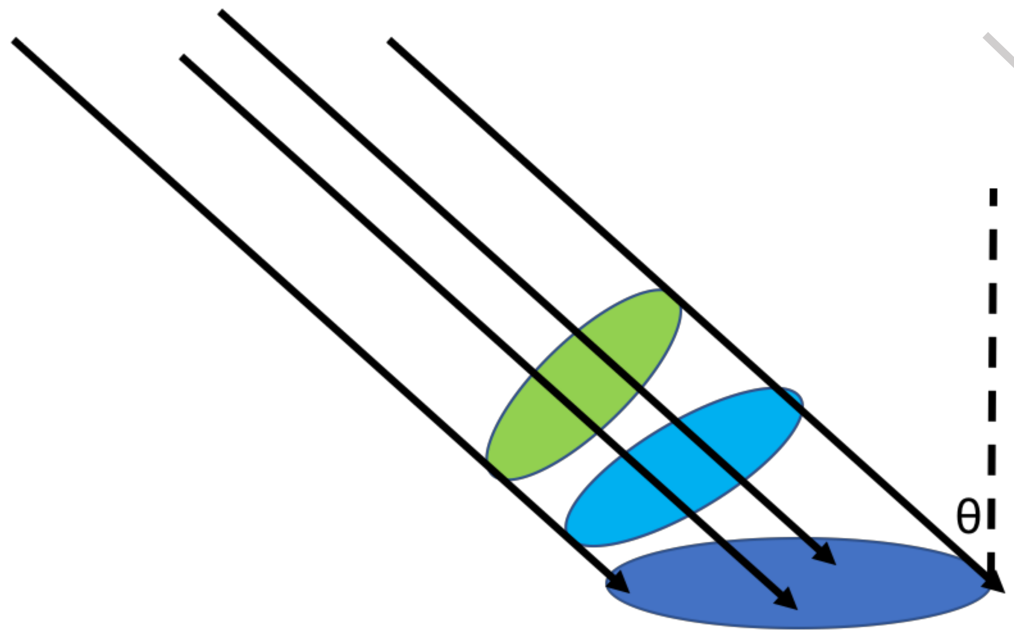
- What is the energy received by a surface from a light source?
- Depends on the area of the surface and its orientation relative to light

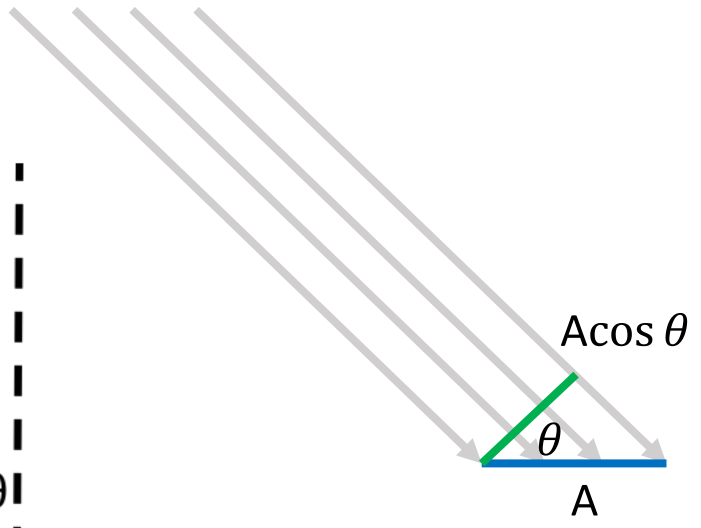




Irradiance

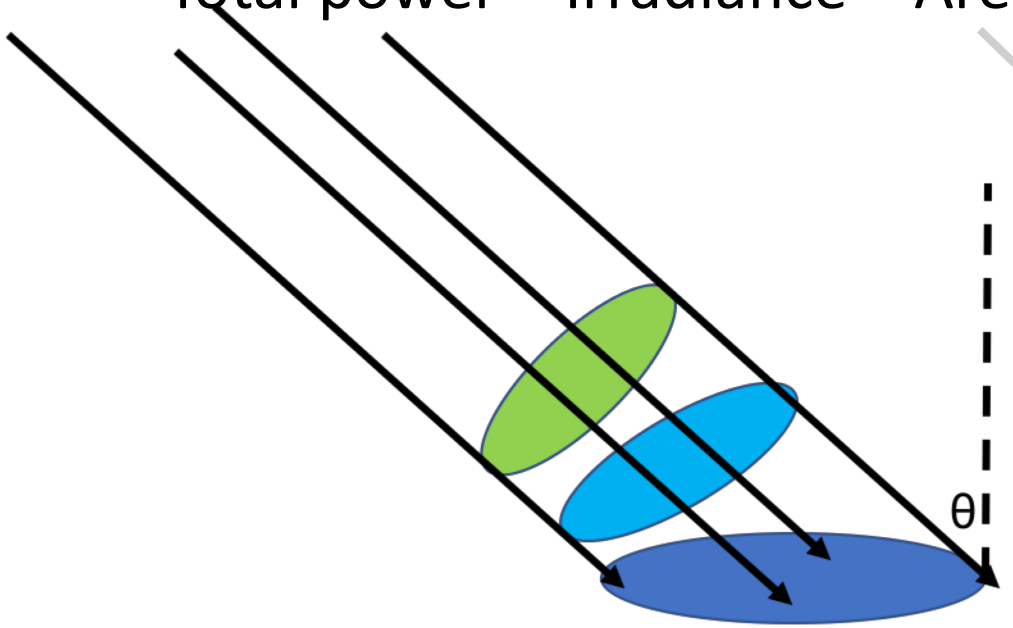
- Power received by a surface patch
    - of area A
    - from a beam of radiance L
    - coming at angle θ= LAcosθ

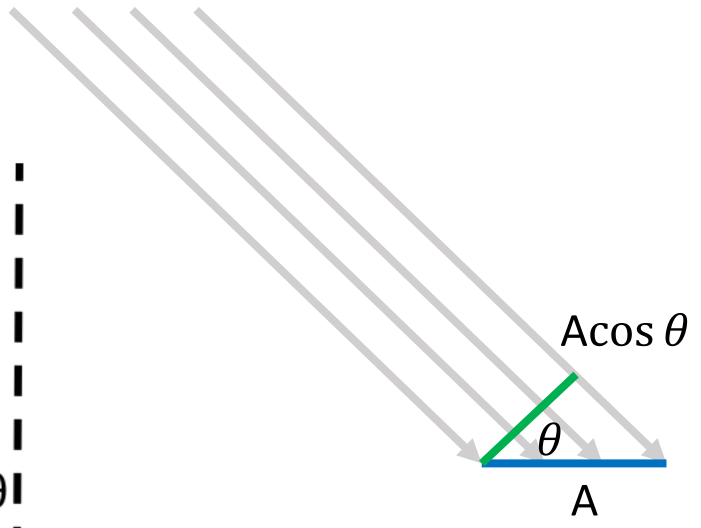




Irradiance

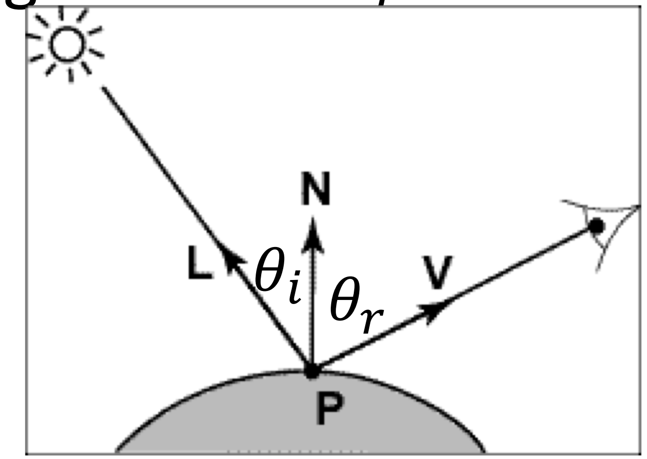
- Power received by a surface patch *of unit area*
  - from a beam of radiance L
  - coming at angle θ= Lcosθ
- Called Irradiance
- Irradiance = Radiance of ray* cosθ
- Total power = Irradiance * Area





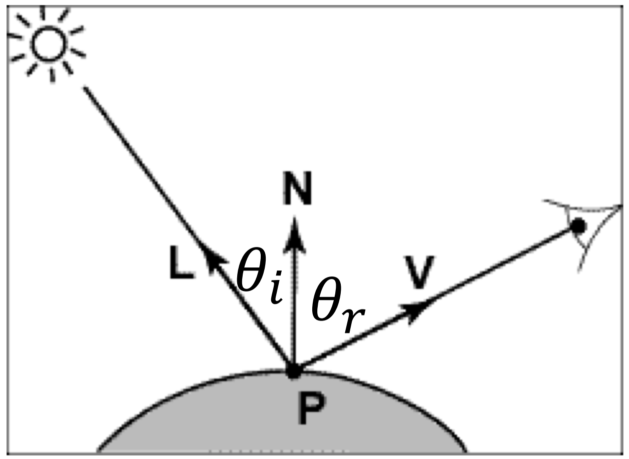
Light rays interacting with a surface

- Light of radiance L_i comes from light source at an incoming direction θ_i : incoming power = $L_i \cos \theta_i$
- Surface absorbs some of this energy and reflects a fraction in the outgoing direction θ_r
  - Fraction might depend on incoming light and outgoing light direction
  - Fraction = $\rho(\theta_i, \theta_r)$
- Outgoing radiance L_r = fraction * incoming power



- **N** is surface normal
- **L** is direction of light, making θ_i with normal
- **V** is viewing direction, making θ_r with normal

Light rays interacting with a surface



- **N** is surface normal
- **L** is direction of light, making θ_i with normal
- **V** is viewing direction, making θ_r with normal

$$L_r = \rho(\theta_i, \theta_r) L_i \cos\theta_i$$

Output radiance along **V**

Incoming irradiance along **L**

Bi-directional reflectance function (BRDF)

Light rays interacting with a surface

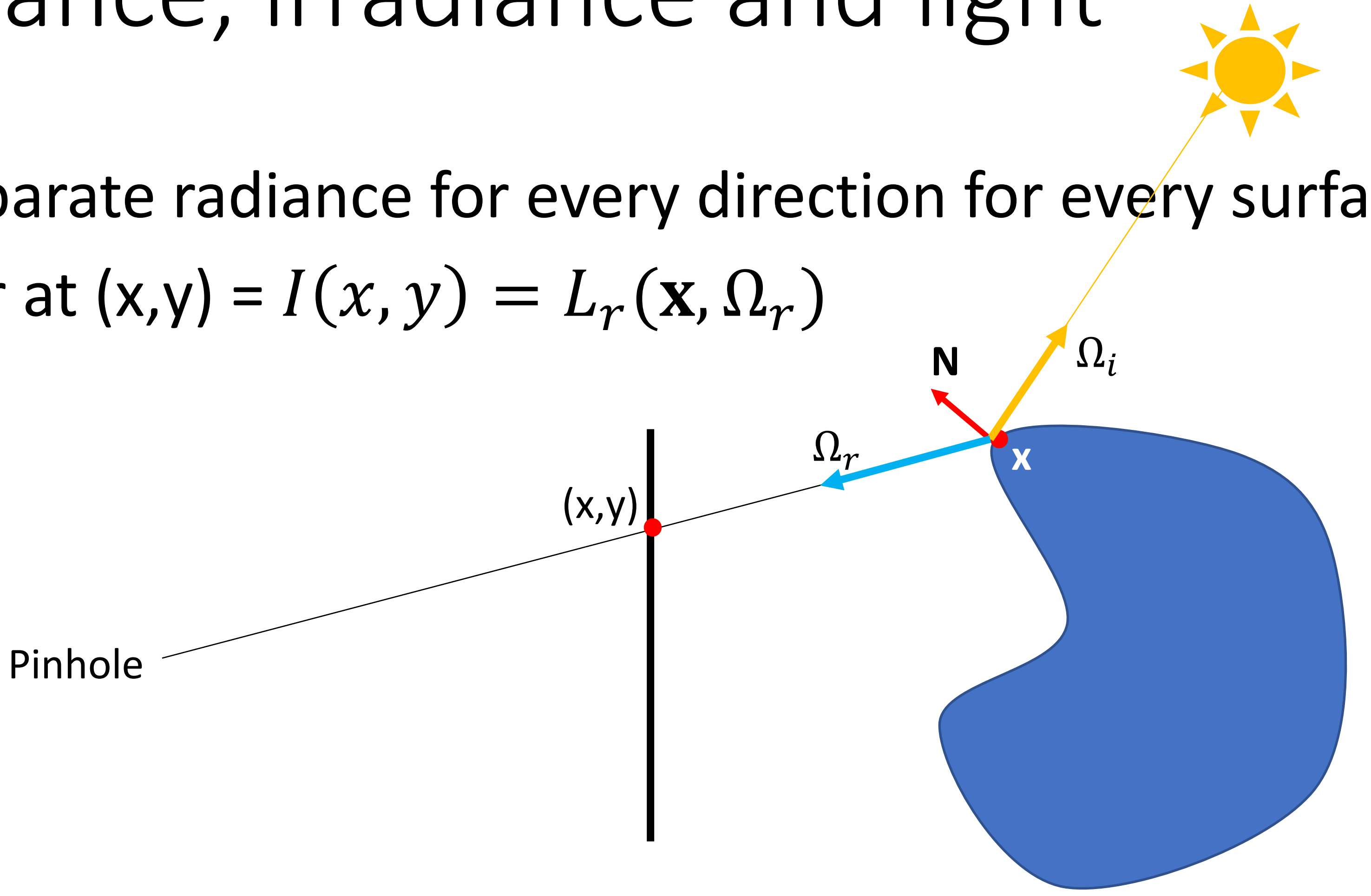
- In reality:
  - World is 3D, so incoming and outgoing directions are not angles θ_i, θ_r but general 3D directions Ω_i, Ω_r (represented by ”solid angles”)
  - Light might come from all directions with different radiance: need to integrate

- Final equation:

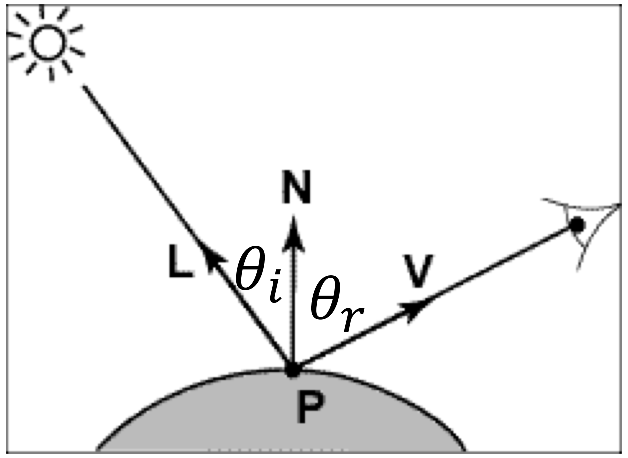
$$\underbrace{L_r(\Omega_r)}_{\text{Outgoing radiance at a point in direction } \Omega_r} = \underbrace{\int_{\Omega_i}}_{\text{Sum / Integral over incoming directions } \Omega_i} \overbrace{\rho(\Omega_i, \Omega_r)}^{\text{BRDF}} \underbrace{L_i \cos\theta_i}_{\text{Incoming irradiance in direction } \Omega_i} d\Omega_i$$

Radiance, irradiance and light

- A separate radiance for every direction for every surface point
- Color at (x,y) = $I(x,y) = L_r(\mathbf{x}, \Omega_r)$



What should BRDF be?

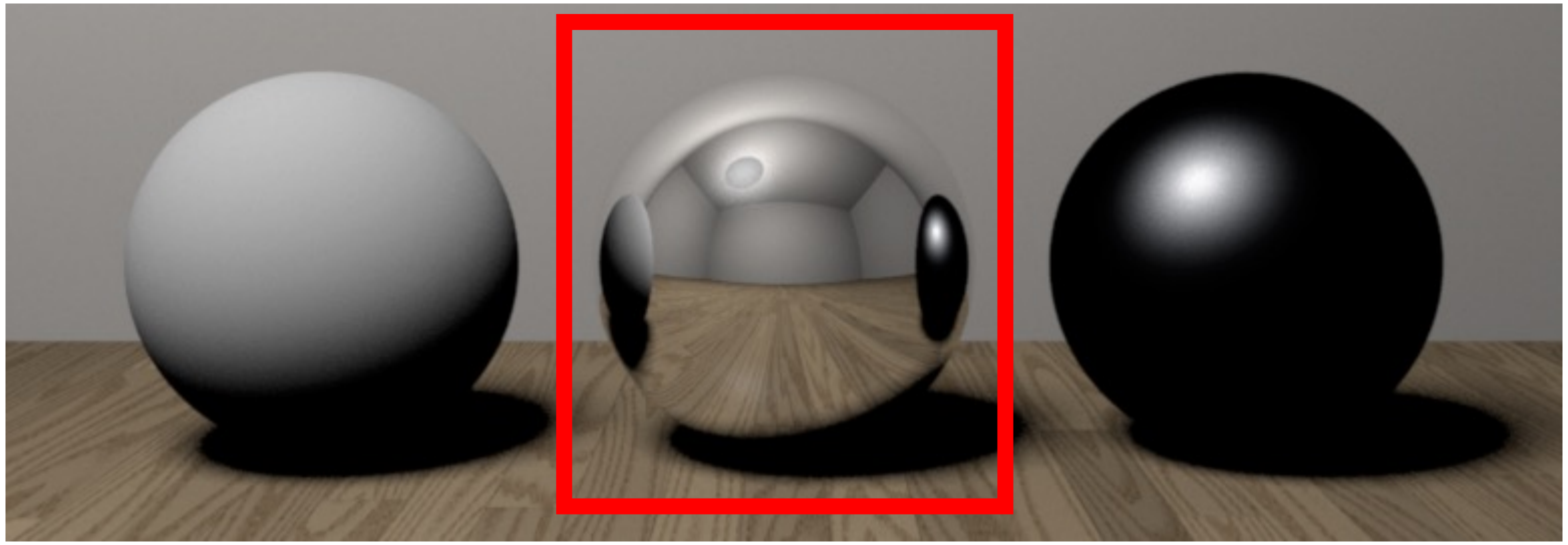


$$L_r = \rho(\theta_i, \theta_r) L_i \cos\theta_i$$

- Special case 1: Perfect mirror
  - $\rho(\theta_i, \theta_r)$ = 0 unless $\theta_i = \theta_r$
- Special case 2: Matte surface
  - $\rho(\theta_i, \theta_r)$ = ρ_0 (constant)

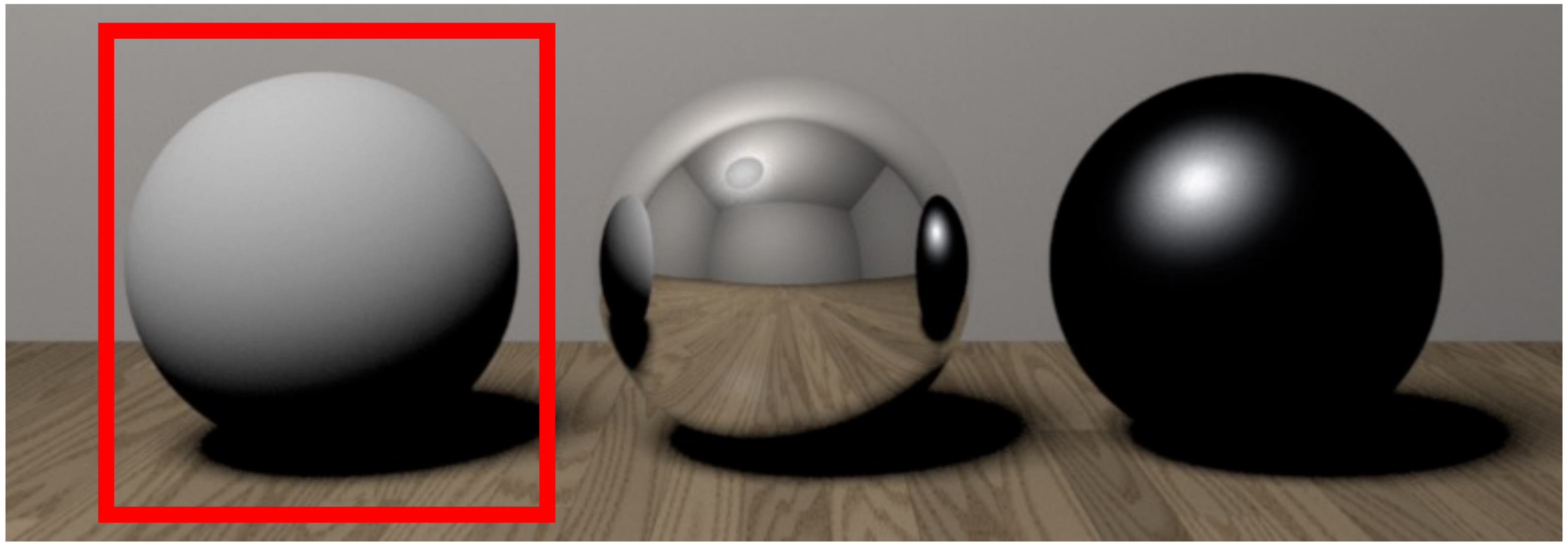
Special case 1: Perfect mirror

- $\rho(\theta_i, \theta_r)$ = 0 unless $\theta_i = \theta_r$
- Also called “Specular surfaces”
- Reflects light in a single, particular direction



Special case 2: Matte surface

- $\rho(\theta_i, \theta_r) = \rho_0$
- Also called “Lambertian surfaces”
- Reflected light is *independent of viewing direction*



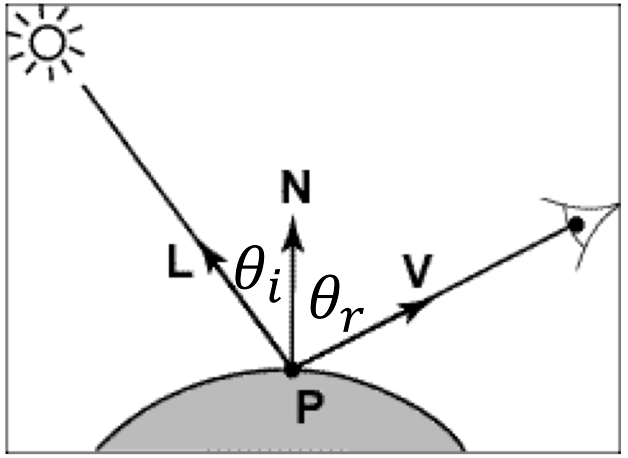
Lambertian surfaces

- For a lambertian surface:

$$L_r = \rho L_i \cos\theta_i$$

$$\Rightarrow L_r = \rho L_i \mathbf{L} \cdot \mathbf{N}$$

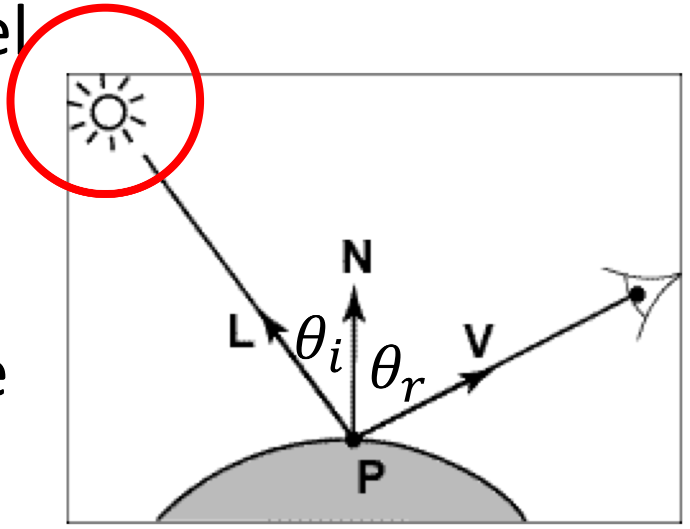
- **L** is direction to light source ($= \Omega_i$)
- L_i is intensity of light
- ρ is called *albedo*
  - Think of this as paint
  - High albedo: white colored surface
  - Low albedo: black surface
  - Varies from point to point



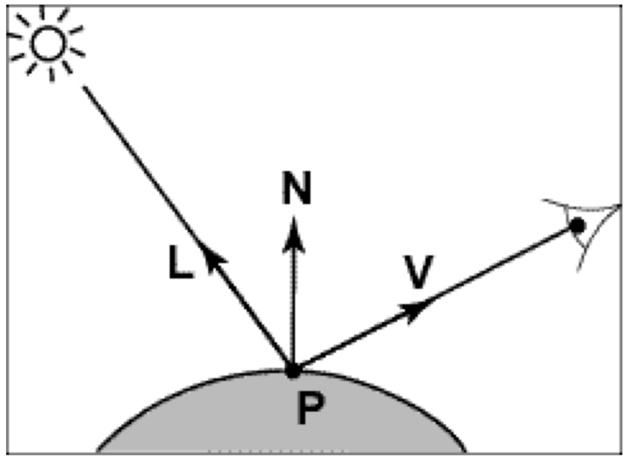
Lambertian surfaces

- Assume the light is directional: all rays from light source are parallel
  - Equivalent to a light source infinitely far away

- All pixels get light from the same direction **L** and of the same intensity L_i



Lambertian surfaces



Intrinsic Image Decomposition

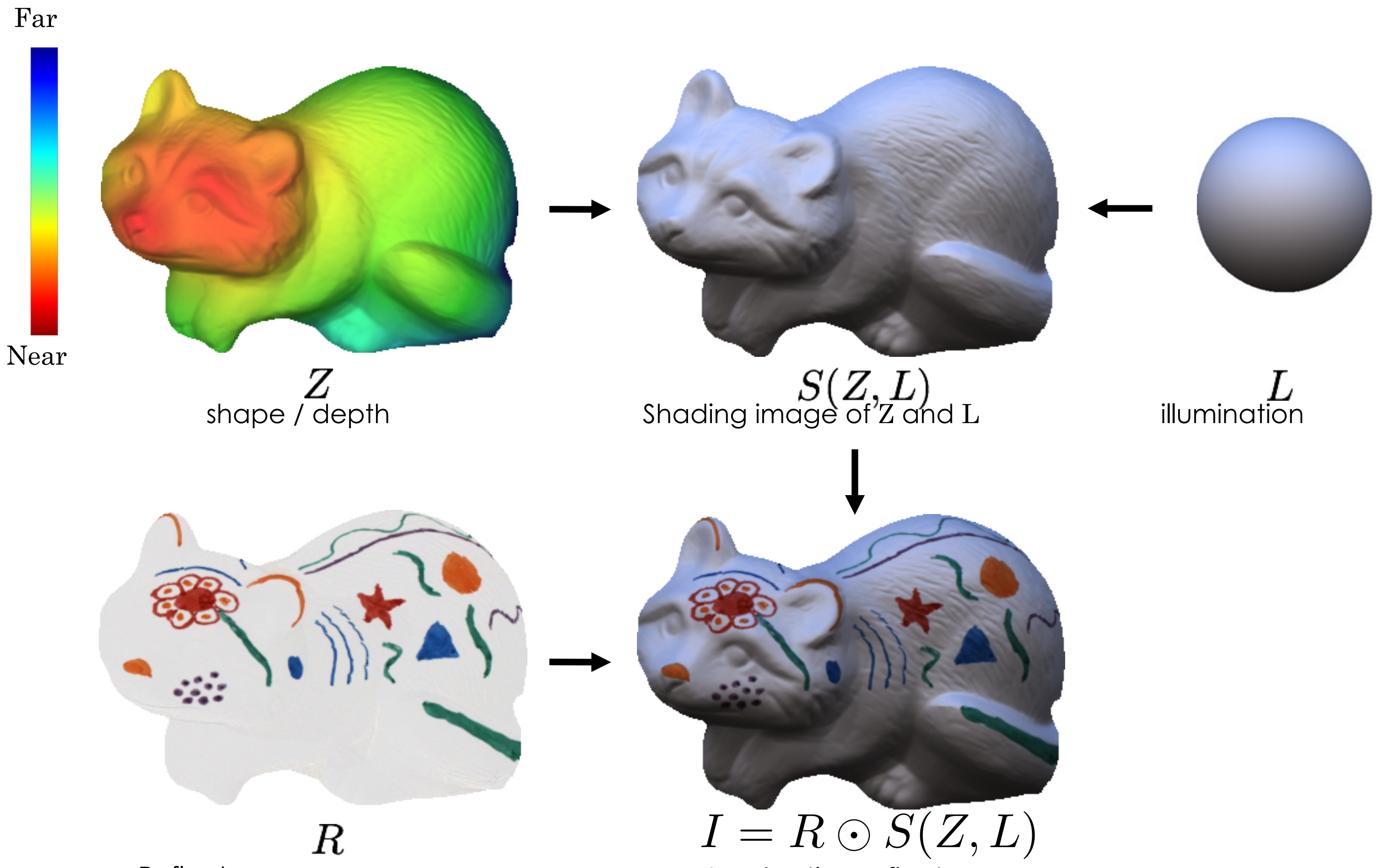
$$I(x,y) = \rho(x,y) L_i \mathbf{L} \cdot \mathbf{N}(x,y)$$

Reflectance image: albedo of surface corresponding to each pixel

Shading image: dot product between light and normal direction at each pixel

Lambertian surfaces

Lambertian surfaces



Reconstructing Lambertian surfaces

$$I(x, y) = \rho(x, y) L_i \mathbf{L} \cdot \mathbf{N}(x, y)$$

- Equation is a constraint on albedo and normals
- Can we solve for albedo and normals?

Recovery from multiple images

$$I(x,y) = \rho(x,y) L_i \mathbf{L} \cdot \mathbf{N}(x,y)$$

- Represents an equation in the albedo and normals
- Multiple images give constraints on albedo and normals
- Solve for albedo and normals
- Called *Photometric Stereo*

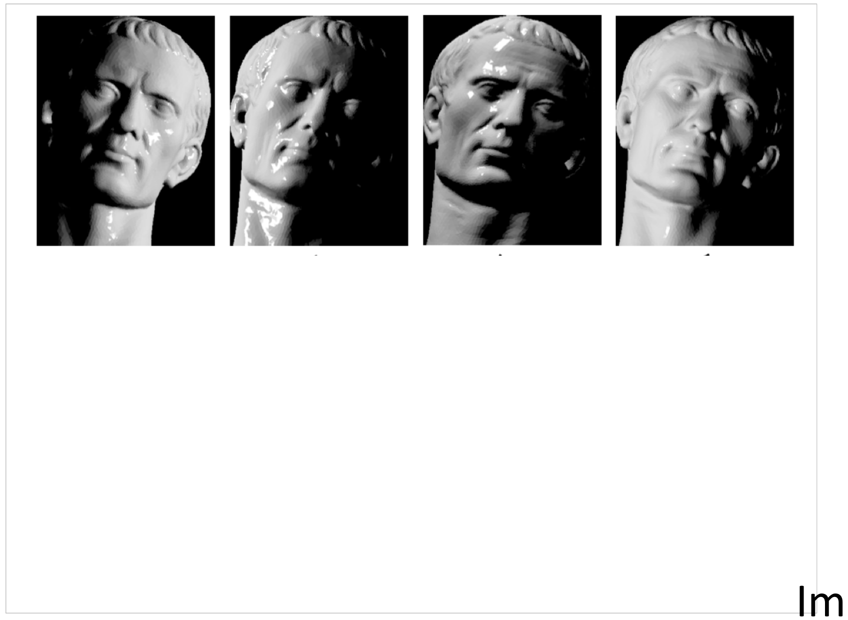
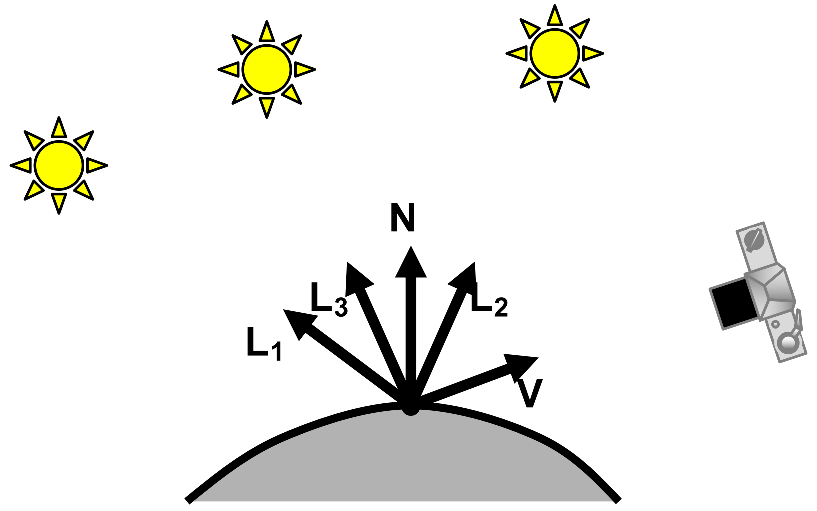


Image credit: Wikipedia

Multiple Images: Photometric Stereo



Photometric stereo - the math

$$I(x,y) = \rho(x,y) L_i \mathbf{L} \cdot \mathbf{N}(x,y)$$

- Consider single pixel
- Assume $L_i = 1$

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

$$I = \rho \mathbf{N}^T \mathbf{L}$$

- Write $\mathbf{G} = \rho \mathbf{N}$
- $\mathbf{G}$ is a 3-vector
    - Norm of **G** = ρ
    - Direction of **G** = **N**

Photometric stereo - the math

- Consider single pixel
- Assume $L_i = 1$

$$I = \rho \mathbf{N}^T \mathbf{L}$$

- Write $\mathbf{G} = \rho \mathbf{N}$
- $\mathbf{G}$ is a 3-vector
  - Norm of **G** = ρ
  - Direction of **G** = **N**

$$I = \mathbf{G}^T \mathbf{L} = \mathbf{L}^T \mathbf{G}$$

Photometric stereo - the math

$$I = \mathbf{L}^T \mathbf{G}$$

- Multiple images with different light sources but same viewing direction?

$$I_1 = \mathbf{L}_1^T \mathbf{G}$$

$$I_2 = \mathbf{L}_2^T \mathbf{G}$$

$$\vdots$$

$$I_k = \mathbf{L}_k^T \mathbf{G}$$

Photometric stereo - the math

$$I_1 = \mathbf{L}_1^T \mathbf{G}$$

$$I_2 = \mathbf{L}_2^T \mathbf{G}$$

$$\vdots$$

$$I_k = \mathbf{L}_k^T \mathbf{G}$$

- Assume lighting directions are known
- Each is a linear equation in G
- Stack everything up into a massive linear system of equations!

Photometric stereo - the math

$$
\begin{aligned}
I_1 &= \mathbf{L}_1^T \mathbf{G} \\
I_2 &= \mathbf{L}_2^T \mathbf{G} \\
&\vdots \\
I_k &= \mathbf{L}_k^T \mathbf{G}
\end{aligned}
\quad \Rightarrow \quad
\underbrace{\mathbf{I}}_{\text{k x 1 vector of intensities}} = \overbrace{\mathbf{L}^T}^{\text{k x 3 matrix of lighting directions}} \underbrace{\mathbf{G}}_{\text{3x1 vector of unknowns}}
$$

Photometric stereo - the math

$$\underset{k \times 1}{\mathbf{I}} = \underset{k \times 3}{\mathbf{L}^T} \underset{3 \times 1}{\mathbf{G}}$$

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

- What is the minimum value of k to allow recovery of G?
- How do we recover G if the problem is overconstrained?

Photometric stereo - the math

- How do we recover G if the problem is overconstrained?
    - More than 3 lights: more than 3 images
- Least squares

$$\min_{\mathbf{G}} \|\mathbf{I} - \mathbf{L}^T\mathbf{G}\|^2$$

- Solved using normal equations

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

Normal equations

$$\|\mathbf{I} - \mathbf{L}^T\mathbf{G}\|^2 = \mathbf{I}^T\mathbf{I} + \mathbf{G}^T\mathbf{L}\mathbf{L}^T\mathbf{G} - 2\mathbf{G}^T\mathbf{L}\mathbf{I}$$

- Take derivative with respect to **G** and set to 0

$$2\mathbf{L}\mathbf{L}^T\mathbf{G} - 2\mathbf{L}\mathbf{I} = 0$$

$$\Rightarrow \mathbf{G} = (\mathbf{L}\mathbf{L}^T)^{-1}\mathbf{L}\mathbf{I}$$

Estimating normals and albedo from **G**

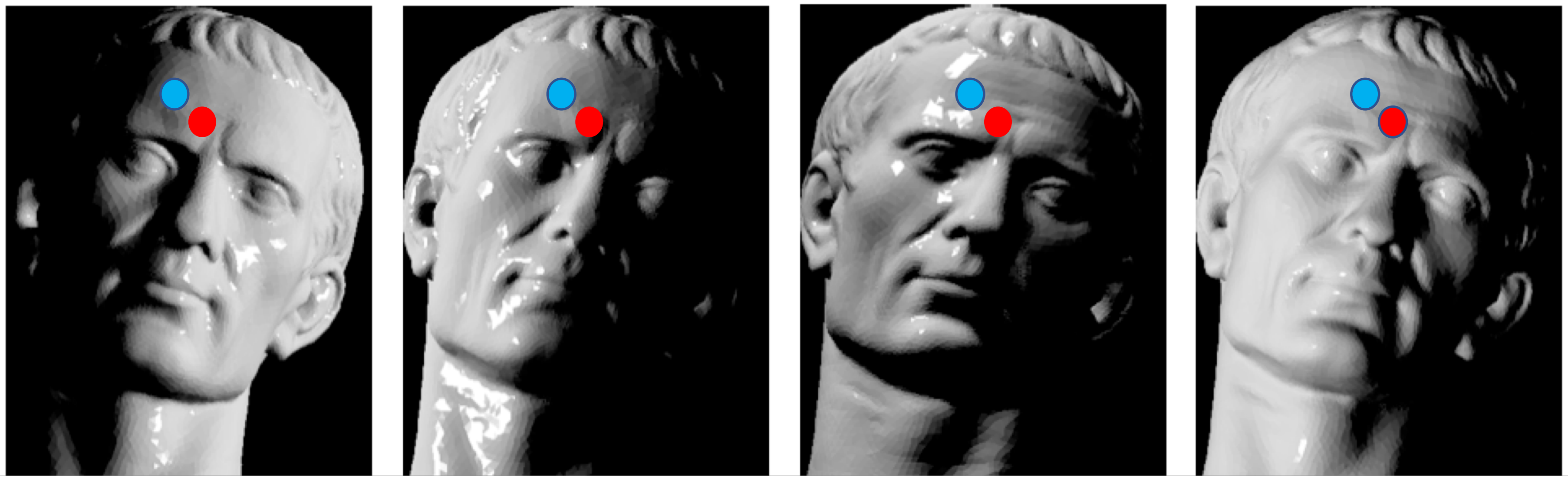
- Recall that $\mathbf{G} = \rho\mathbf{N}$

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

$$\frac{\mathbf{G}}{\|\mathbf{G}\|} = \mathbf{N}$$

Multiple pixels

- We've looked at a single pixel till now
- How do we handle multiple pixels?
- Essentially independent equations!



Multiple pixels: matrix form

- Note that all pixels share the same set of lights

$$\mathbf{I}^{(1)} = \mathbf{L}^T \mathbf{G}^{(1)}$$

$$\mathbf{I}^{(2)} = \mathbf{L}^T \mathbf{G}^{(2)}$$

$$\vdots$$

$$\mathbf{I}^{(n)} = \mathbf{L}^T \mathbf{G}^{(n)}$$

Multiple pixels: matrix form

- Can stack these into *columns* of a matrix

$$\mathbf{I}^{(1)} = \mathbf{L}^T \mathbf{G}^{(1)}$$

$$\mathbf{I}^{(2)} = \mathbf{L}^T \mathbf{G}^{(2)}$$

$$\vdots$$

$$\mathbf{I}^{(n)} = \mathbf{L}^T \mathbf{G}^{(n)}$$

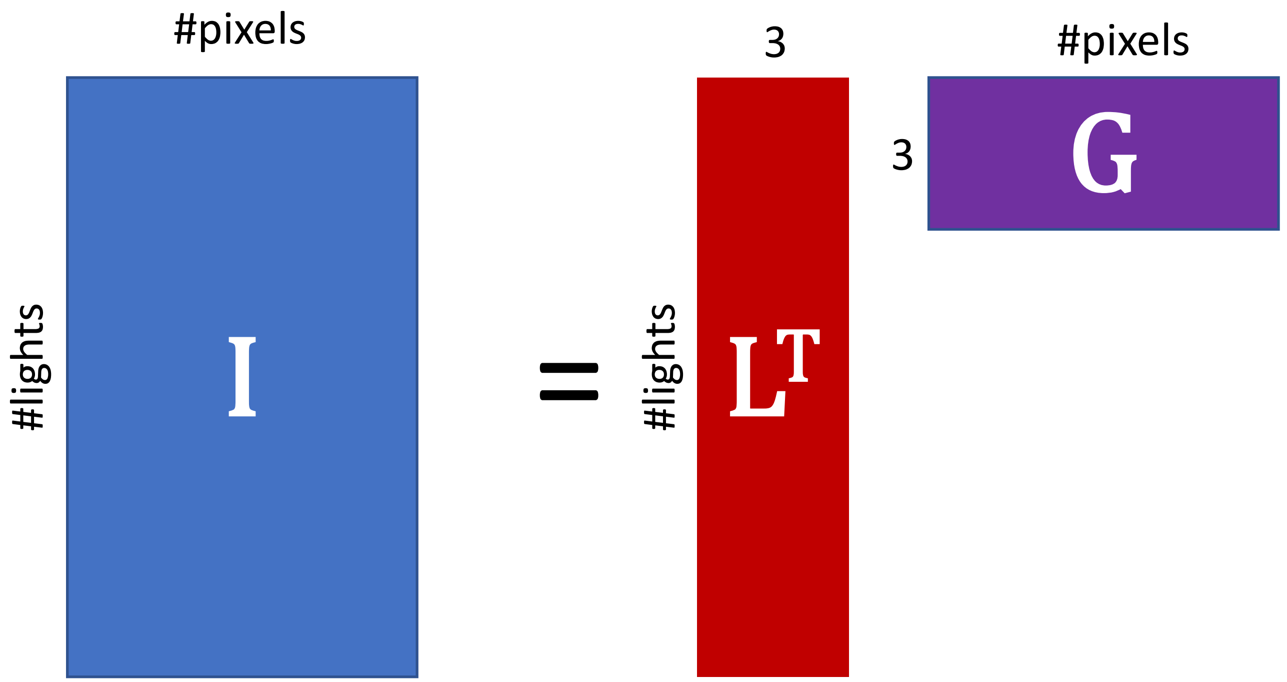
$$\Big\Downarrow$$

$$\begin{bmatrix}\mathbf{I}^{(1)} & \mathbf{I}^{(2)} & \cdots & \mathbf{I}^{(n)}\end{bmatrix} = \mathbf{L}^T \begin{bmatrix}\mathbf{G}^{(1)} & \mathbf{G}^{(2)} & \cdots & \mathbf{G}^{(n)}\end{bmatrix}$$

$$\boxed{\mathbf{I} = \mathbf{L}^T \mathbf{G}}$$

Multiple pixels: matrix form

$$\mathbf{I} = \mathbf{L}^T \mathbf{G}$$

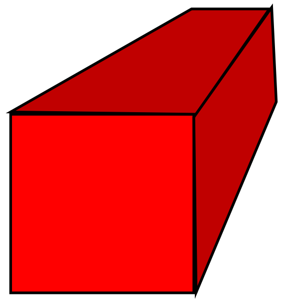


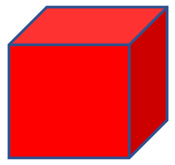
Estimating depth from normals

- So we got surface normals, can we get depth?
- Yes, given *boundary conditions*
- Normals provide information about the derivative

Brief detour: Orthographic projection

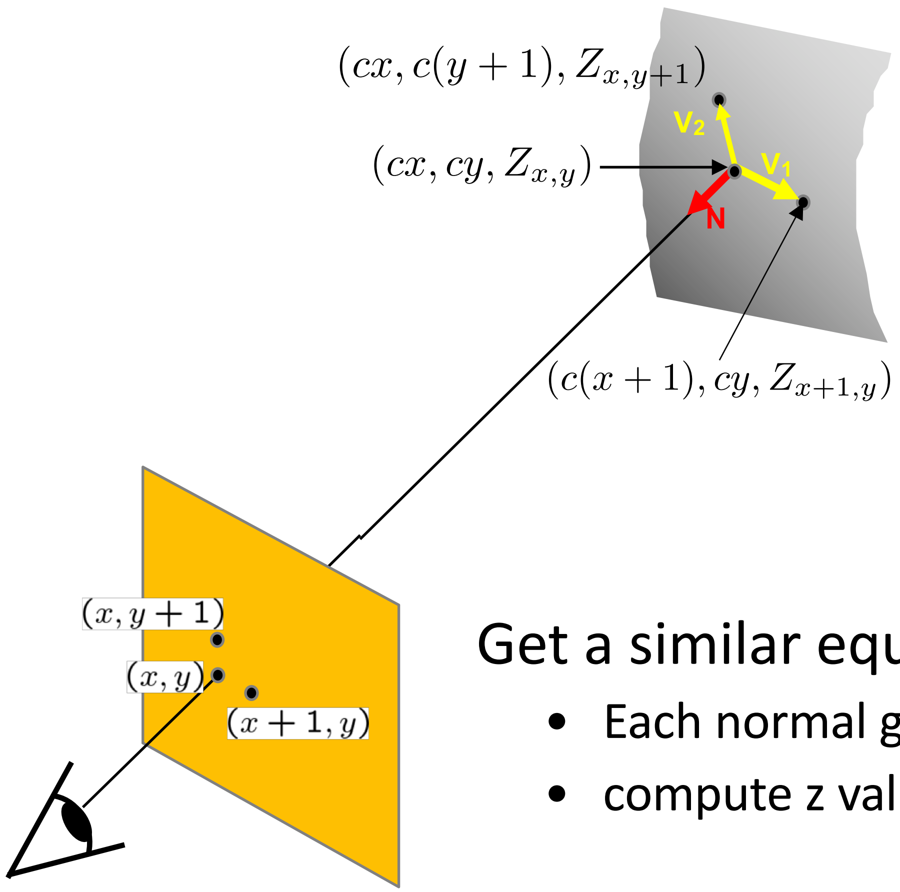
- Perspective projection
  - $x = \frac{X}{Z}, y = \frac{Y}{Z}$
- If all points have similar depth
  - $Z \approx Z_0$
  - $x \approx \frac{X}{Z_0}, y \approx \frac{Y}{Z_0}$
  - $x \approx cX, y \approx cY$
- A scaled version of orthographic projection
  - $x = X, y = Y$


Perspective


Scaled orthographic

Depth Map from Normal Map

- We now have a surface normal, but how do we get depth?



Assume a smooth surface

$$\begin{aligned} V_1 &= (c(x+1), cy, Z_{x+1,y}) - (cx, cy, Z_{x,y}) \\ &= (c, 0, Z_{x+1,y} - Z_{x,y}) \end{aligned}$$

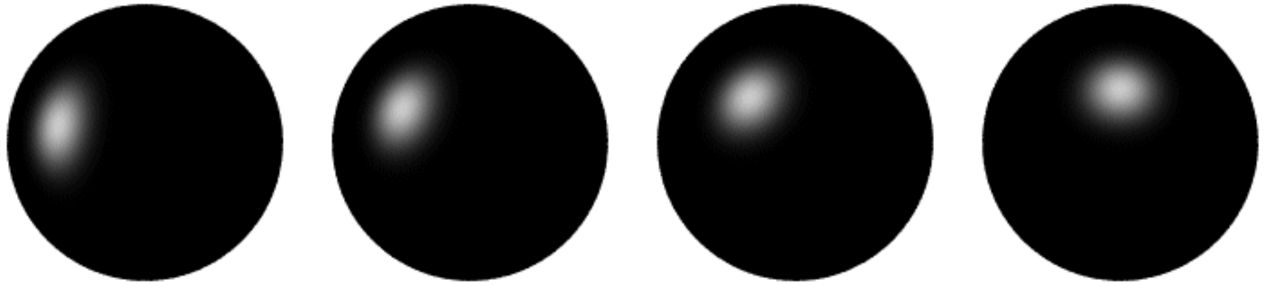
$$\begin{aligned} 0 &= N \cdot V_1 \\ &= (n_x, n_y, n_z) \cdot (c, 0, Z_{x+1,y} - Z_{x,y}) \\ &= cn_x + n_z(Z_{x+1,y} - Z_{x,y}) \end{aligned}$$

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

Determining Light Directions

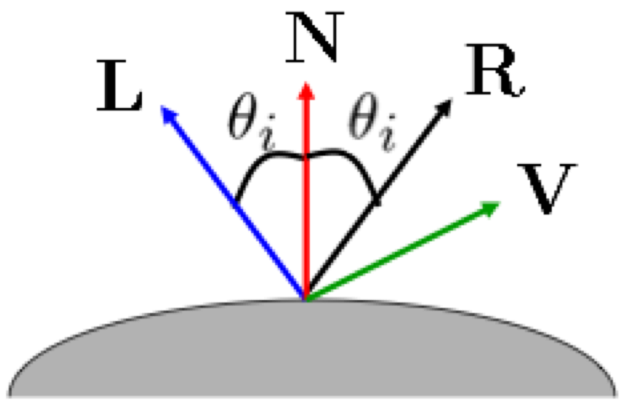
- Trick: Place a mirror ball in the scene.



- The location of the highlight is determined by the light source direction.
- Can relate the direction of highlight mathematically to direction of light source

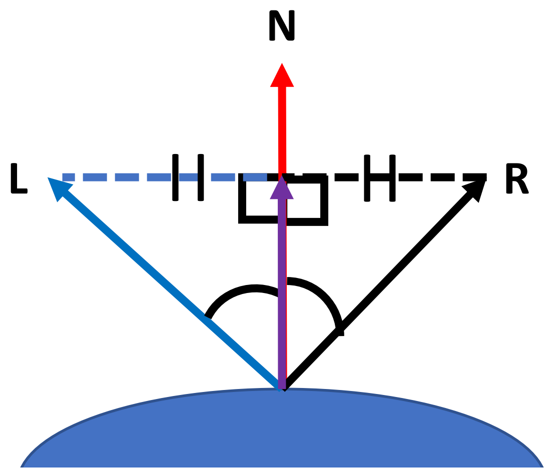
Optional: Determining Light Directions

- For a perfect mirror, the light is reflected across N:



$$I_e = \begin{cases} I_i & \text{if } \mathbf{V} = \mathbf{R} \\ 0 & \text{otherwise} \end{cases}$$

Optional: Determining Light Directions



$$\longrightarrow = (N \cdot R)N$$

$$\text{-------} = R - (N \cdot R)N$$

$$\text{-------} = R - (N \cdot R)N$$

$$\longrightarrow = \longrightarrow - 2 \ \text{-------}$$

$$= R - 2(R - N \cdot R)N$$

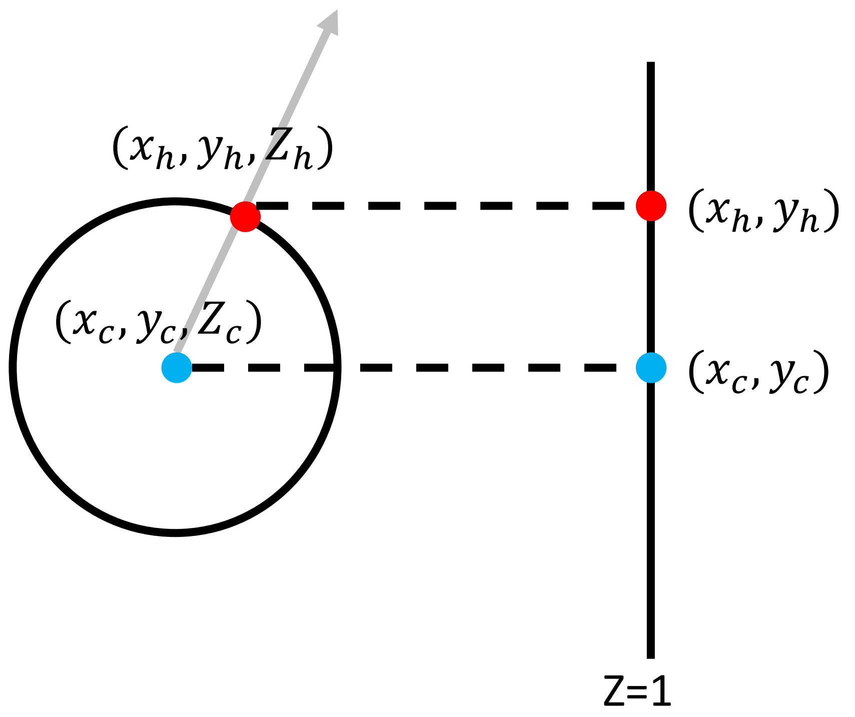
$$= 2(N \cdot R)N - R$$

So the light source direction is given by:

$$L = 2(N \cdot R)N - R$$

Optional: Determining Light Directions

- Assume orthographic projection
- Viewing direction R = [0,0,-1]
- Normal?



Z_h and Z_c are unknown, but:

$$(x_h - x_c)^2 + (y_h - y_c)^2 + (Z_h - Z_c)^2 = r^2$$

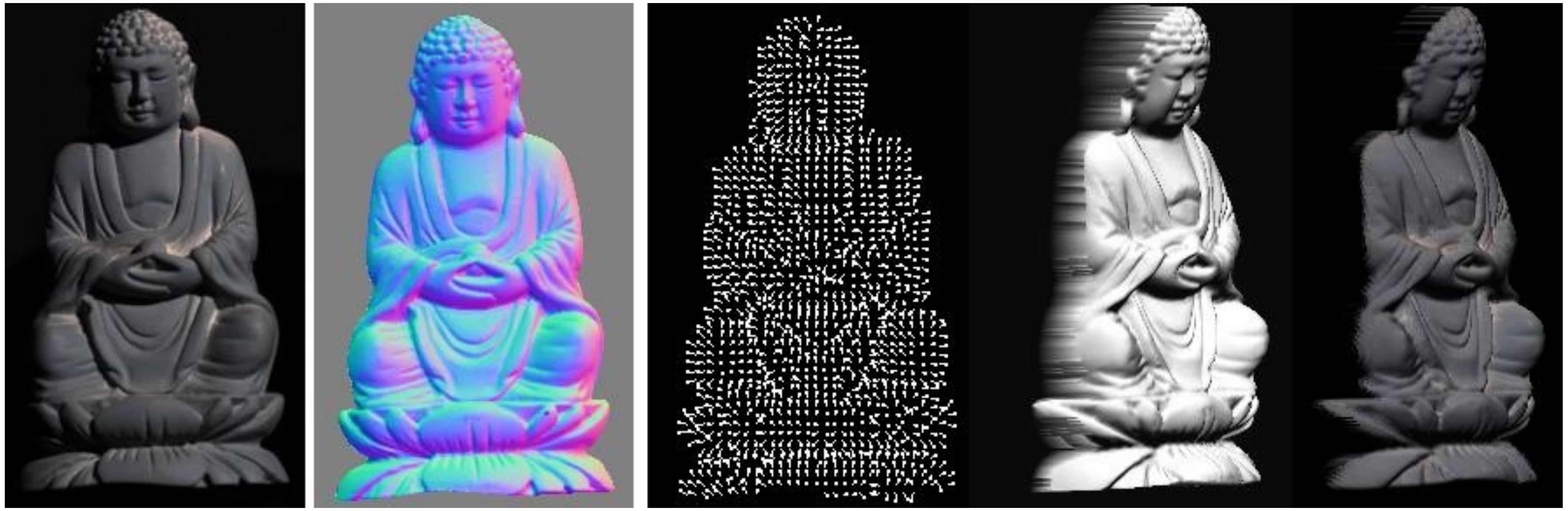
$(Z_h - Z_c)$ can be computed

$(x_h - x_c, y_h - y_c, Z_h - Z_c)$ is the normal

$$L = 2(N \cdot R)N - R$$

Photometric Stereo

What results can you get?



Input
(1 of 12)

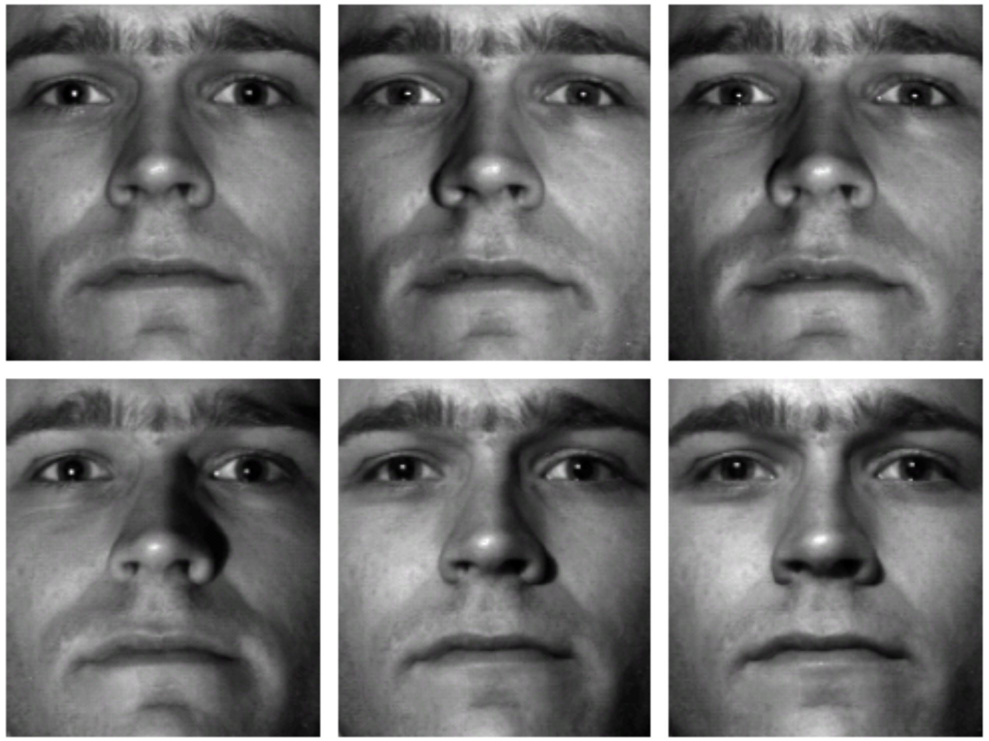
Normals (RGB
colormap)

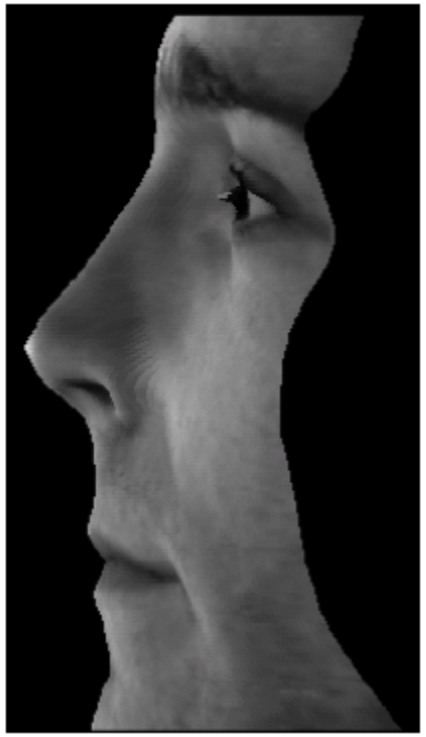
Normals (vectors)

Shaded 3D
rendering

Textured 3D
rendering

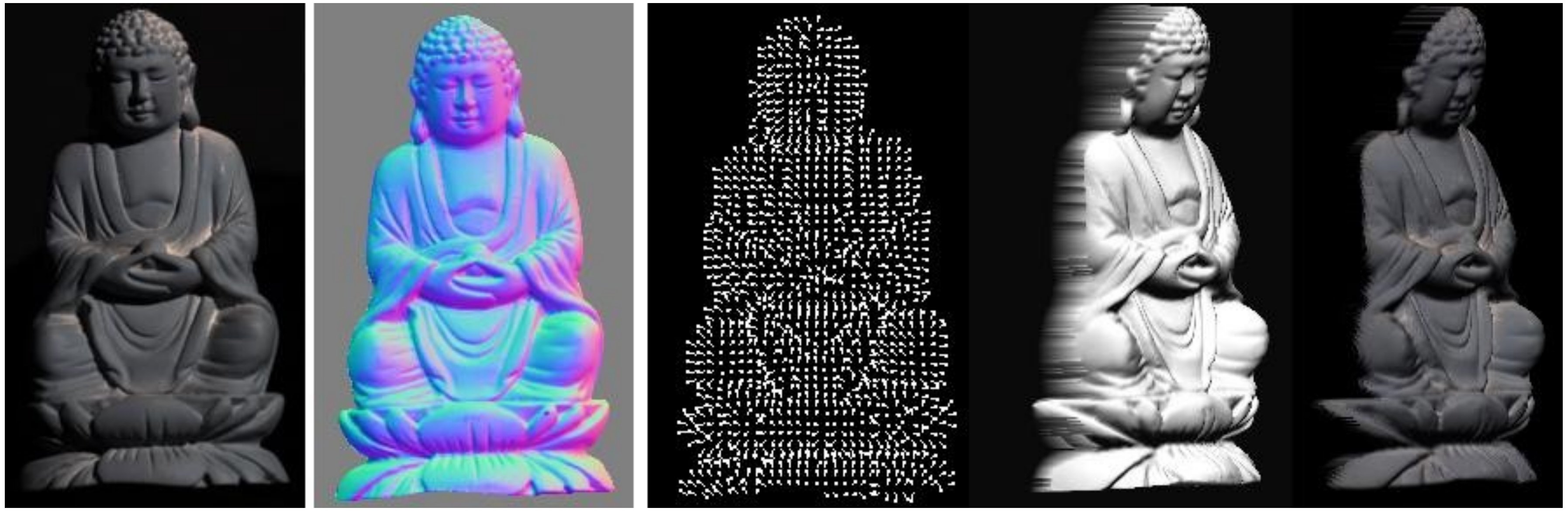
Results





from Athos Georghiades

Results



Input
(1 of 12)

Normals (RGB
colormap)

Normals (vectors)

Shaded 3D
rendering

Textured 3D
rendering