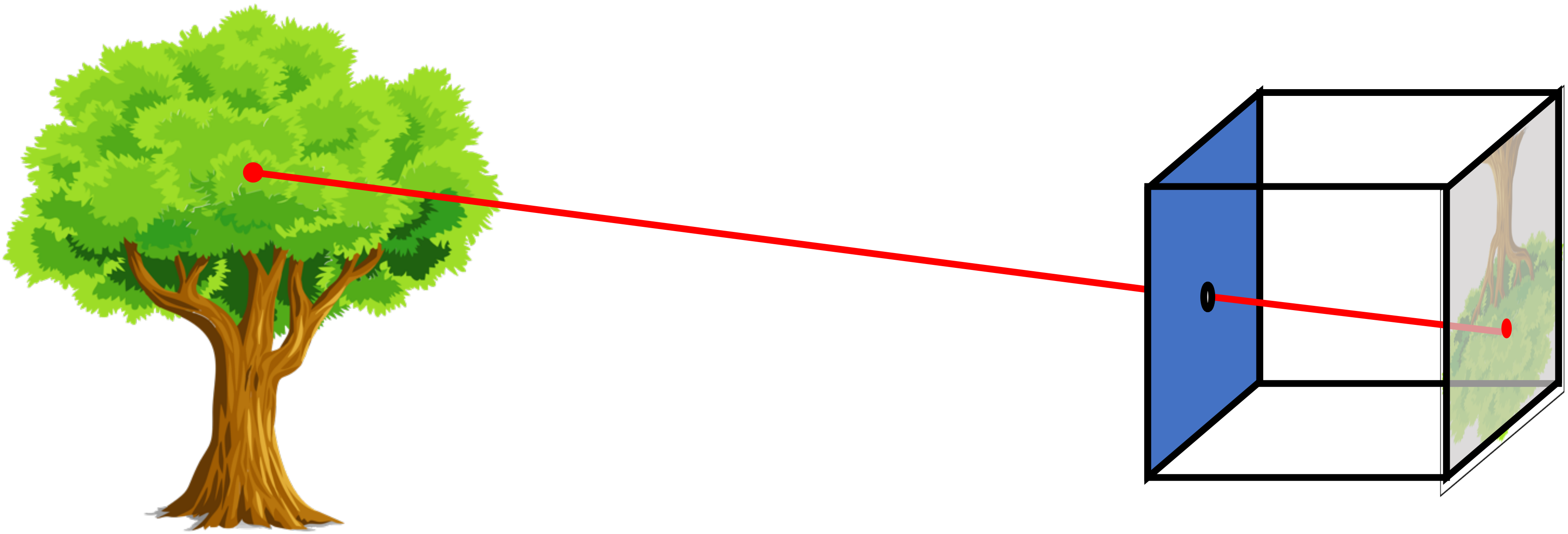
Recap: Geometry of image formation

The pinhole camera



Let's get into the math

Putting everything together

- Change coordinate system so that center of the coordinate system is at pinhole and Z axis is along viewing direction

$$\mathbf{x}'_w = R\mathbf{x}_w + \mathbf{t}$$

- Perspective projection

$$\mathbf{x}'_w \equiv (X, Y, Z)$$

$$\mathbf{x}'_{img} \equiv (x, y)$$

$$x = \frac{X}{Z}$$

$$y = \frac{Y}{Z}$$

Can projection be represented as a matrix multiplication?

Matrix multiplication

$$\begin{bmatrix} a & b & c \\ p & q & r \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} aX + bY + cZ \\ pX + qY + rZ \end{bmatrix}$$

Perspective projection

$$x = \frac{X}{Z}$$

$$y = \frac{Y}{Z}$$

Projective space and homogenous coordinates

- Mapping $\mathbb{R}^2$ to $\mathbb{P}^2$ (Cartesian to homogenous coordinates):

$$(x, y) \rightarrow (x, y, 1)$$

- Mapping $\mathbb{P}^2$ to $\mathbb{R}^2$ (homogenous to cartesian):

$$(x, y, z) \rightarrow (\frac{x}{z}, \frac{y}{z})$$

- A change of coordinates

Homogenous coordinates

- In standard Euclidean coordinates
  - 2D points : (x,y)
  - 3D points : (x,y,z)

- In homogenous coordinates
  - 2D points : (x,y,1)
  - 3D points : (x,y,z,1)

Why homogenous coordinates?

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \equiv \begin{bmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \\ 1 \end{bmatrix}$$

Homogenous coordinates of world point

Homogenous coordinates of image point

Homogenous coordinates

$$\begin{bmatrix} a & b & c & t_x \\ d & e & f & t_y \\ g & h & i & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} aX + bY + cZ + t_x \\ dX + eY + fZ + t_y \\ gX + hY + iZ + t_z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} M & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_w \\ 1 \end{bmatrix} = \begin{bmatrix} M\mathbf{x}_w + \mathbf{t} \\ 1 \end{bmatrix}$$

Perspective projection in homogenous coordinates

$$\vec{\mathbf{x}}_{img} = \begin{bmatrix} I & \mathbf{0} \end{bmatrix} \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \vec{\mathbf{x}}_w$$

$$\vec{\mathbf{x}}_{img} = \begin{bmatrix} R & \mathbf{t} \end{bmatrix} \vec{\mathbf{x}}_w$$

Matrix transformations in 2D

$$\vec{\mathbf{x}}_{img} \equiv K \begin{bmatrix} R & \mathbf{t} \end{bmatrix} \vec{\mathbf{x}}_w$$

$$K = \begin{bmatrix} 1 & 0 & t_u \\ 0 & 1 & t_v \\ 0 & 0 & 1 \end{bmatrix}$$

Translation

$$K = \begin{bmatrix} s_x & 0 & t_u \\ 0 & s_y & t_v \\ 0 & 0 & 1 \end{bmatrix}$$

Scaling of Image x and y (conversion from “meters” to “pixels”)

$$K = \begin{bmatrix} s_x & \alpha & t_u \\ 0 & s_y & t_v \\ 0 & 0 & 1 \end{bmatrix}$$

Added skew if image x and y axes are not perpendicular

Final perspective projection

Camera extrinsics: where your camera is relative to the world. Changes if you move the camera

$$\vec{\mathbf{x}}_{img} \equiv \boxed{K} \boxed{\begin{bmatrix} R & \mathbf{t} \end{bmatrix}} \vec{\mathbf{x}}_w$$

Camera intrinsics: how your camera handles pixel. Changes if you change your camera

$$\vec{\mathbf{x}}_{img} \equiv P \vec{\mathbf{x}}_w$$

Final perspective projection

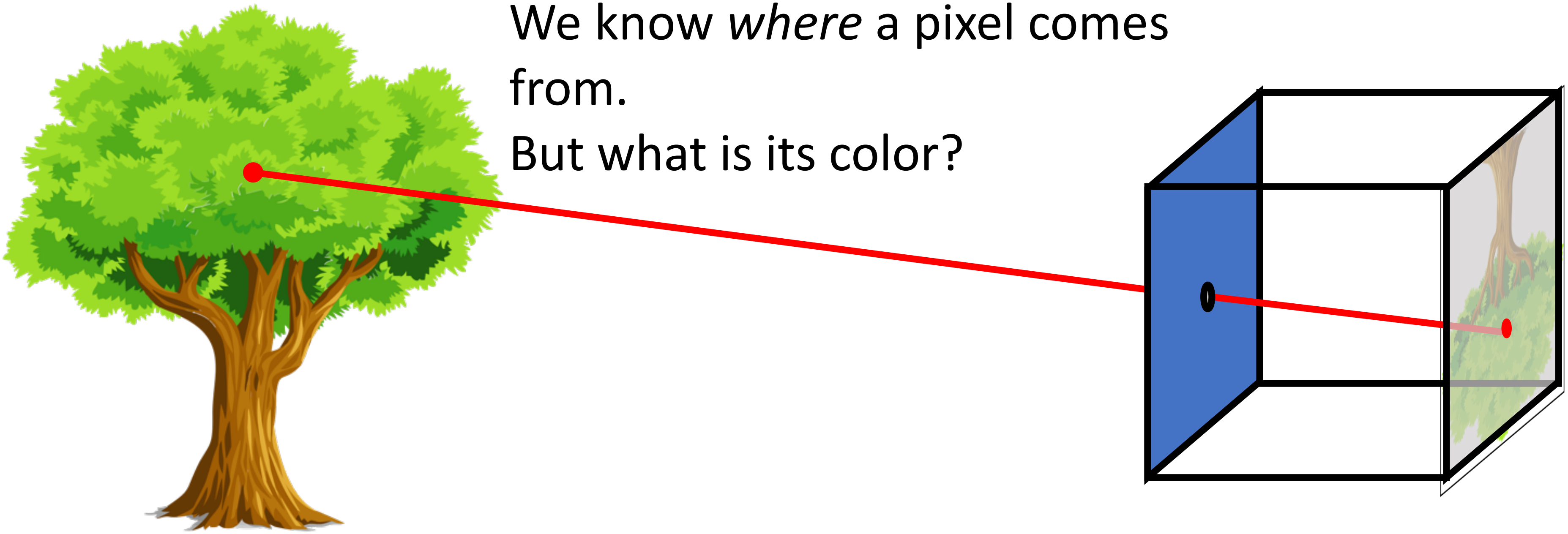
$$\vec{\mathbf{x}}_{img} \equiv \underbrace{K \begin{bmatrix} R & \mathbf{t} \end{bmatrix}}_{\text{Camera parameters}} \vec{\mathbf{x}}_w$$

$$\vec{\mathbf{x}}_{img} \equiv P \vec{\mathbf{x}}_w$$

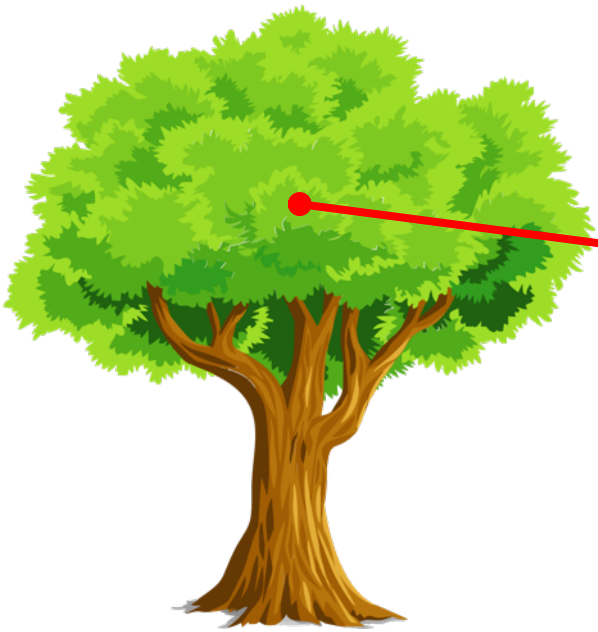
Image Formation - Color

The pinhole camera

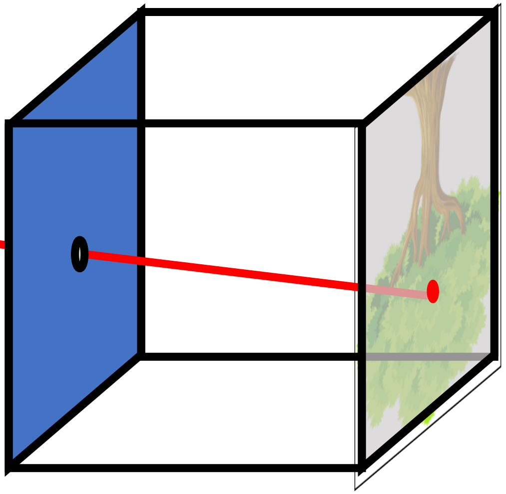
We know *where* a pixel comes from.
But what is its color?



The pinhole camera



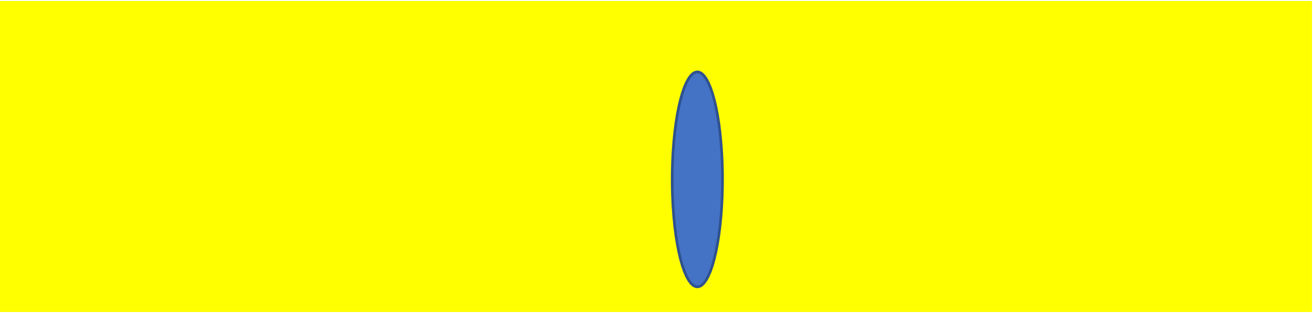
We know *where* a pixel comes from.
But what is its color?



- A pixel is some kind of sensor that measures incident energy
- But what exactly does it measure?

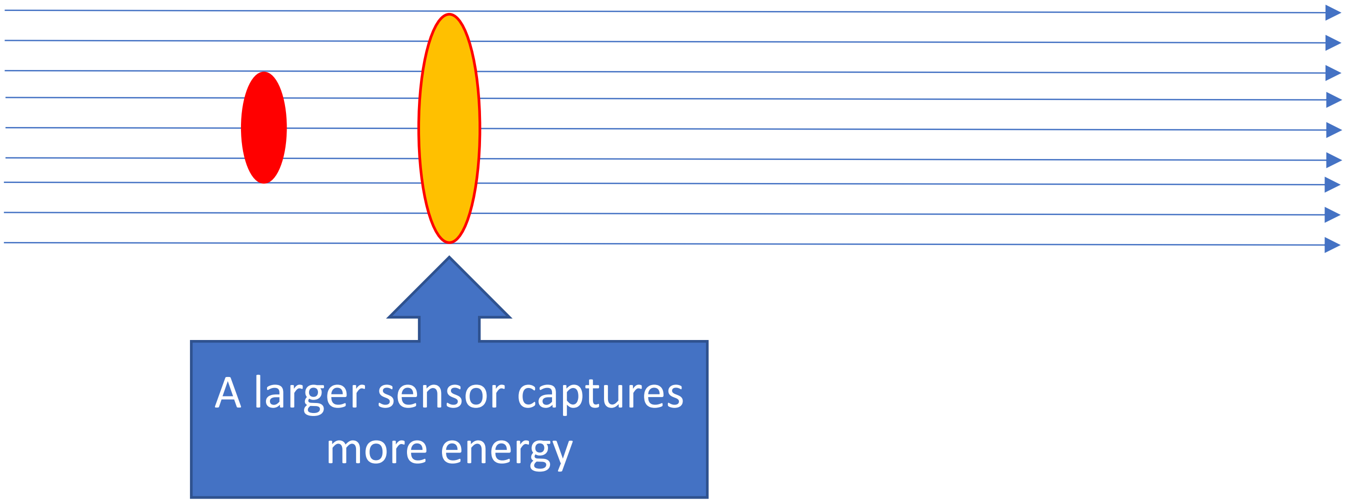
Sensing light

- Consider a sensor placed in a single beam of light.
- How much energy/time does it get?
    - Not enough information



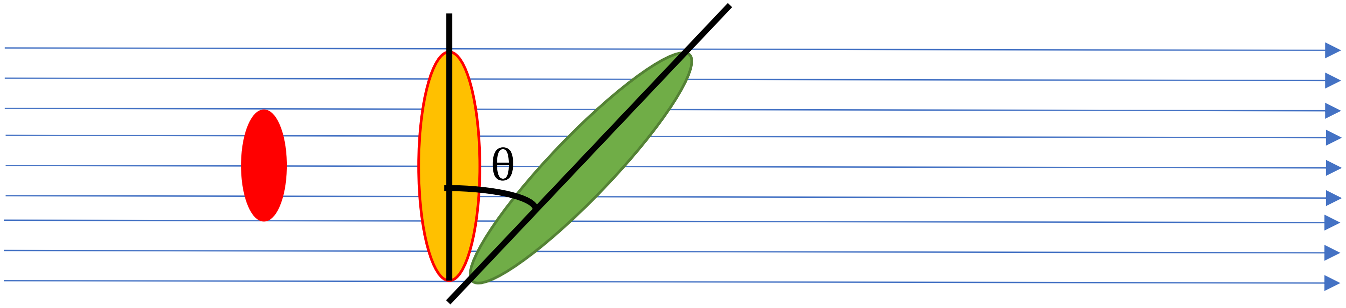
Factor 1: Area

- Larger sensors capture more power
  - Power = LA?
  - L: measure of beam brightness (***radiance***)
  - Radiance is power per unit area?

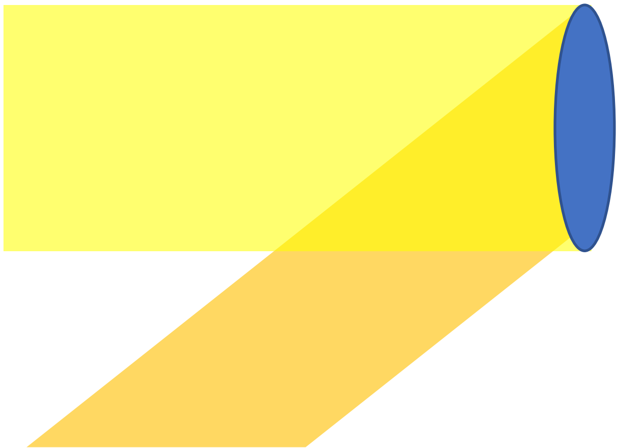


Factor 2: Orientation

- Slanted sensors receive less light
  - Power = $LA\cos\theta$
  - L = Radiance = Power per unit *projected* area



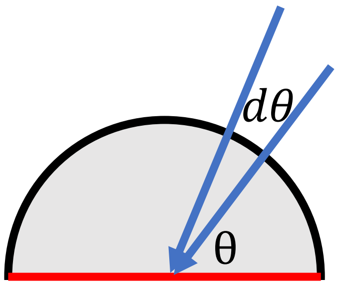
Multiple beams



- Power must be sum of power from each beam
  - Power = $L_1 A \ \cos\theta_1 + L_2 A \ \cos\theta_2$
  - θ_1 and θ_2 are dependent on beam direction
  - Similarly L_1 and L_2
- General case: Light comes from all directions
  - Must *integrate infinitesimal contributions from all directions*

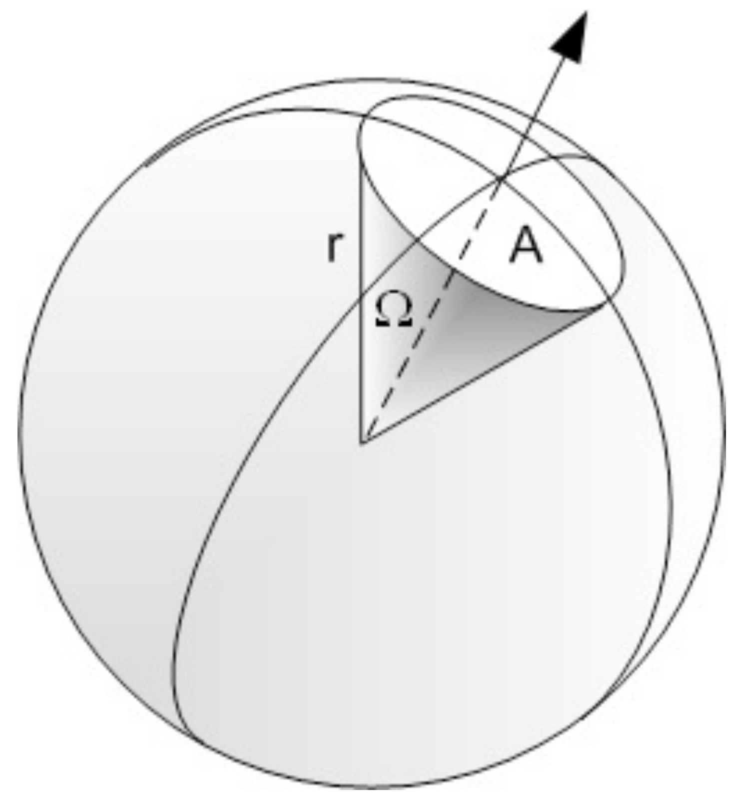
A hemisphere of directions

- In 2D, direction = angle
- Infinitesimal set of directions = infinitesimal angle
- Integrate over all directions = integrate over angle
- 3D?



A hemisphere of directions

- In 3D direction = *solid angle*
- Definition:
  - 2D: angle = arc length / radius
  - 3D: solid angle = *area / radius*^{2}



- $\Omega = \dfrac{A}{r^2}$

Multiple beams

- Integrate incident energy from all directions
- Power = $\int L(\Omega)\, A\ \cos\theta(\Omega)\ d\Omega$
- Radiance = L = ***Power in a particular direction per unit projected area per unit solid angle***

Integrating over area

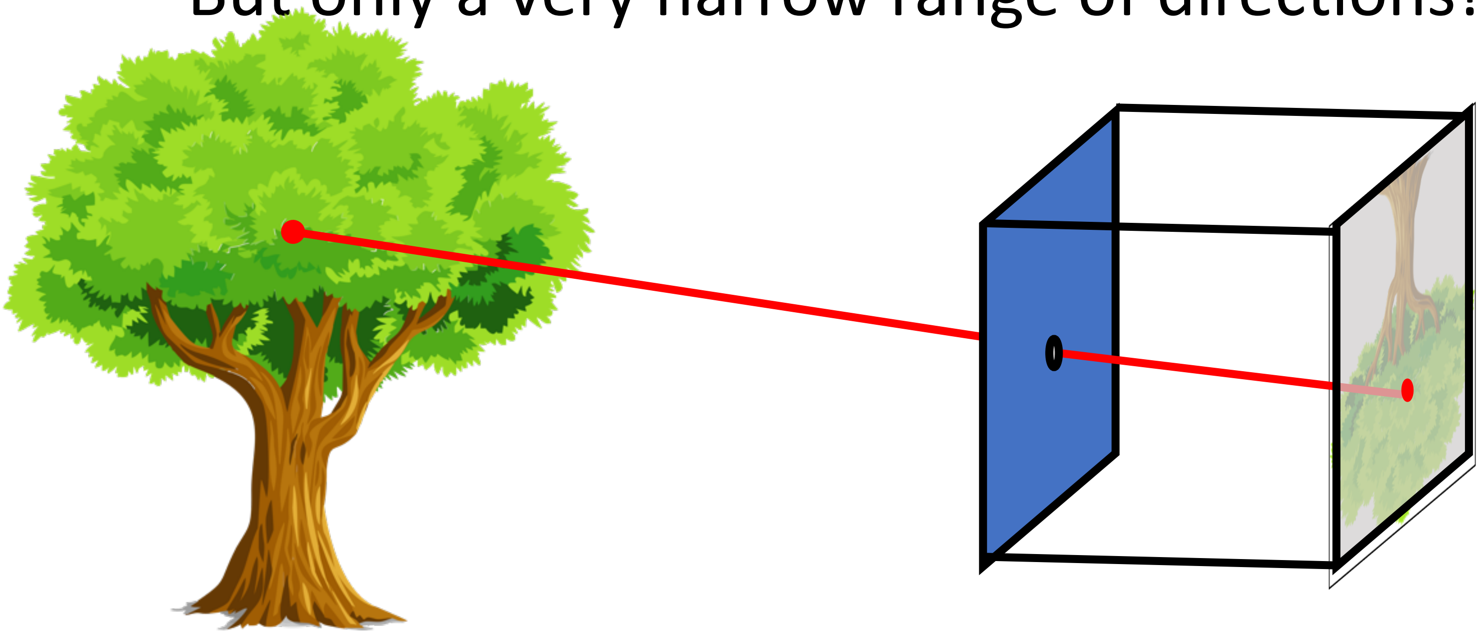
- What if sensor is not flat?
  - Orientation depends on location
- What if parts of the sensor receive less light?
  - L depends on location
- Divide sensor into infinitesimal elements and integrate
  - Power = $\int \int L(x, \Omega) \cos\theta(x, \Omega)\, dA d\Omega$

Radiance

- Power = $\int \int L(x,\Omega) \cos\theta(x,\Omega)\, dA d\Omega$
- $L(x,\Omega)$ is the **Radiance**
  - **Power** at point x
  - in direction Ω
  - per unit projected area
  - per unit solid angle

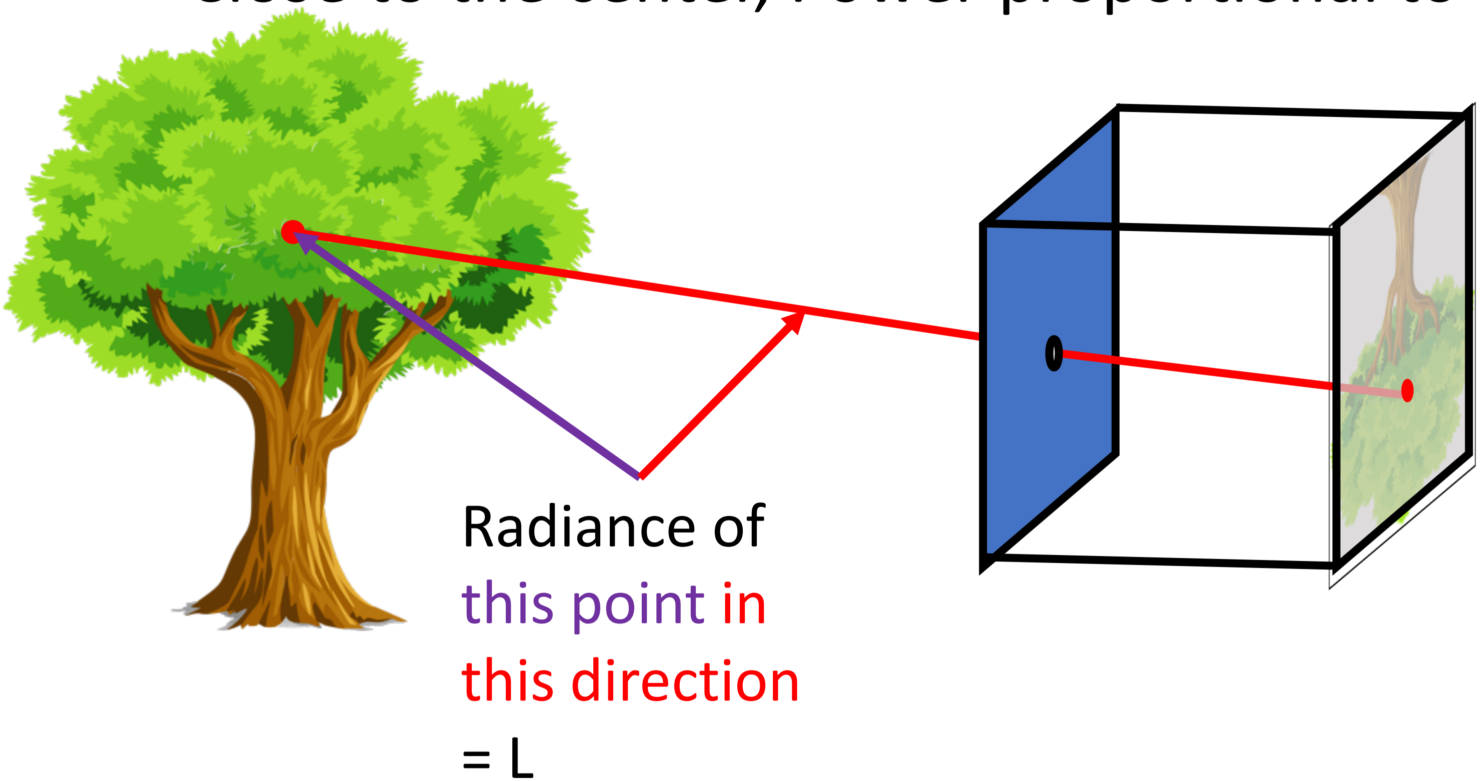
What do pixels measure?

- A pixel measures total power incident on it
- Power = $\int\int L(x,\Omega)\cos\theta(x,\Omega)\, dA\, d\Omega$
- But only a very narrow range of directions!



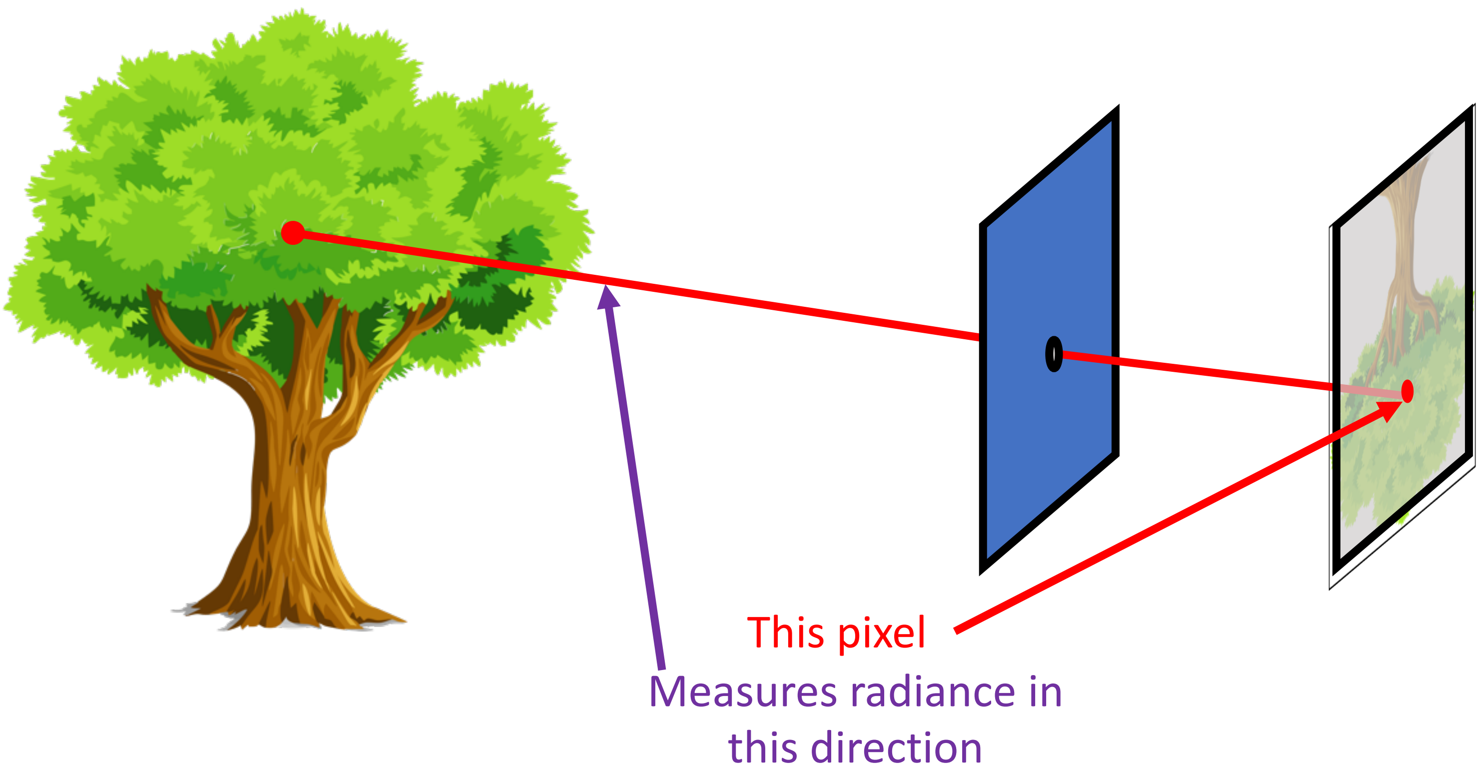
What do pixels measure?

- A pixel measures total power incident on it
- Power = $LA \ \cos\theta$?
- Close to the center, Power proportional to *L*



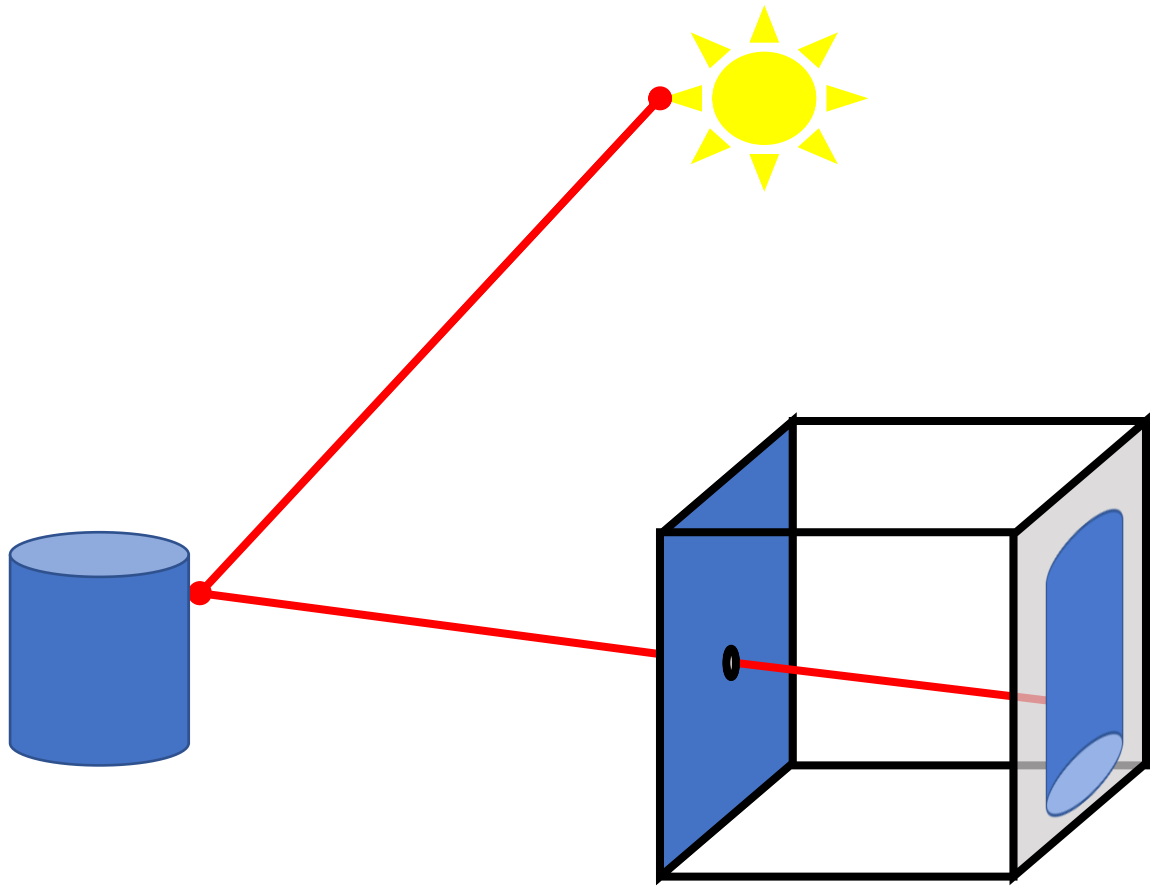
Radiance

- Pixels measure ***radiance***

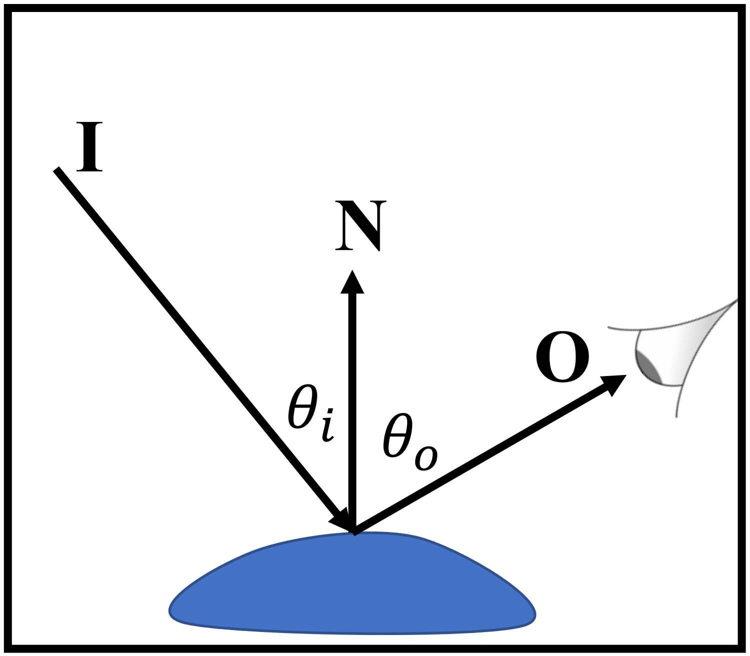


Where do the rays come from?

- Rays from the light source "reflect" off a surface and reach camera
- Reflection: Surface absorbs light energy and radiates it back

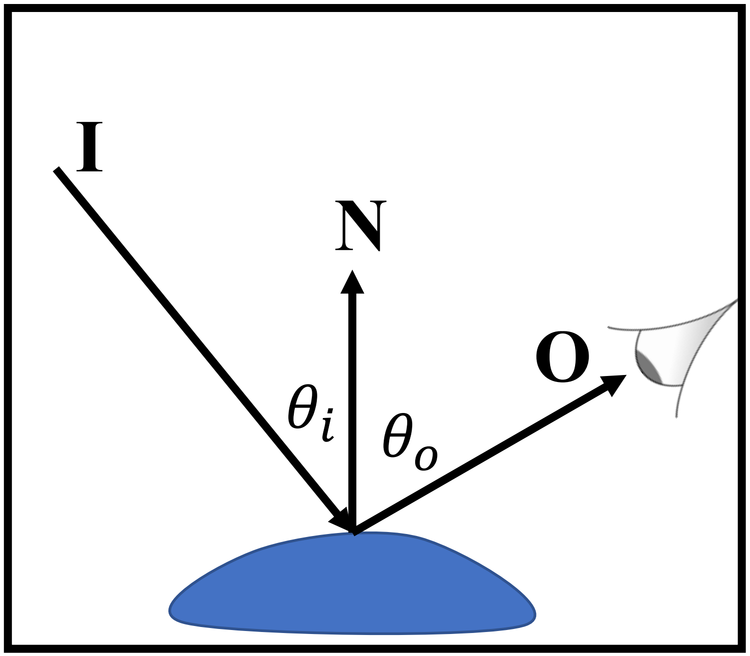


Light rays interacting with a surface



- **I** : Incoming light direction (only one direction)
- **O** : Outgoing light direction (viewing direction)
- **N** : Surface normal
- L_i: Incoming light radiance
- L_o: Outgoing light radiance

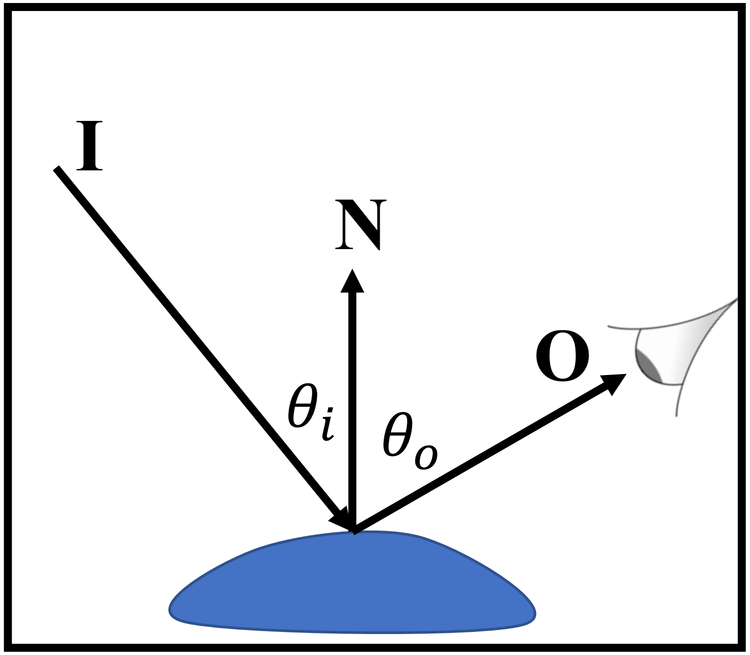
Light rays interacting with a surface



- Consider a surface patch of unit area
- How much power does it receive?
- $E_i = L_i \cos\theta_i$
- Some fraction of this will be emitted
- Fraction might depend on I, O

$$L_o = \rho(I,O)E_i$$
$$= \rho(I,O)L_i \cos\theta_i$$

Light rays interacting with a surface



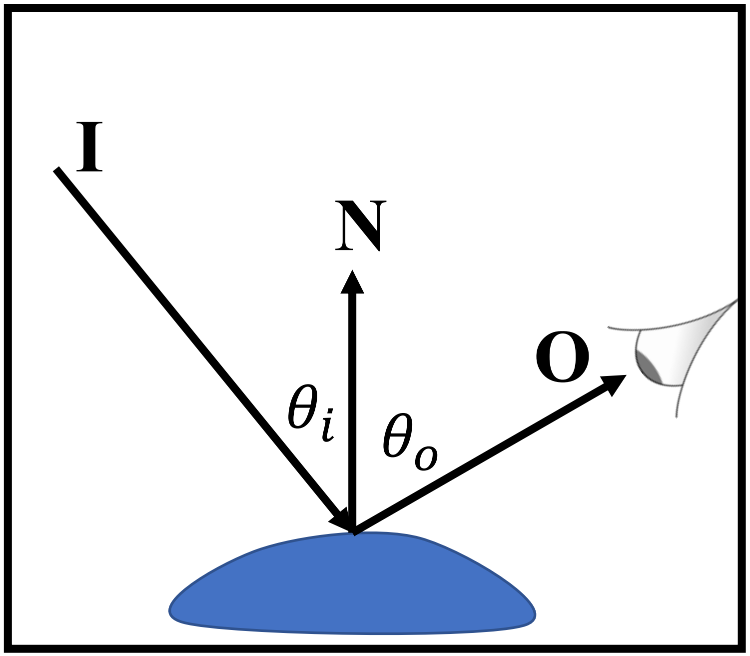
Incoming energy (**Irradiance**)

$$L_o = \rho(I, O) L_i \cos\theta_i$$

BRDF: Bidirectional reflectance distribution function

Note does not capture phenomena like sub-surface scattering

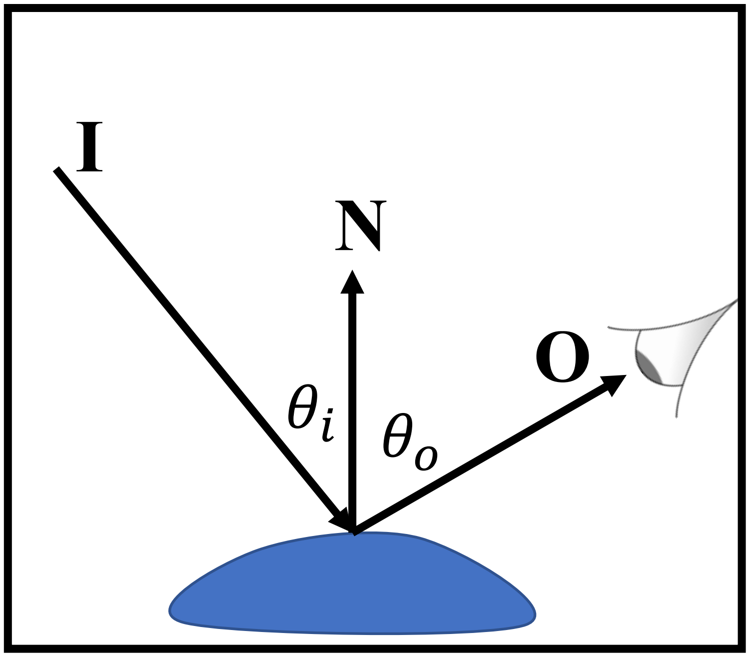
Light rays interacting with a surface



$$L_o = \rho(I, O) L_i \cos\theta_i$$

- Special case 1: Specular surfaces
  - All light reflected in a single direction
  - $\rho(I, O) = 0$ unless $\theta_i = \theta_r$

Light rays interacting with a surface



$$L_o = \rho(I, O) L_i \cos\theta_i$$

- Special case 2: Matte surfaces
  - Light reflected equally in all directions
  - $\rho(I, O) = \rho$ (constant)
  - ρ is **albedo** : amount of paint
  - These are also called **Lambertian surfaces**

Lambertian surface

- $L_o = \rho L_i \cos \theta_i$
- Outgoing radiance does not depend on viewing direction
- Given same light, pixel looks the same from all views
- Frequent assumption in computer vision

Intrinsic image decomposition

- Consider a lambertian scene lit with directional light
- Image pixel (x,y) corresponds to point in scene with
  - albedo $\rho(x,y)$
  - surface normal making angle $\theta_i(x,y)$ with light direction
- Pixel color:

$$\underbrace{I(x,y)}_{\text{Image}} = \underbrace{\rho(x,y)}_{\text{"Reflectance" image}} \underbrace{L_i \cos\theta_i(x,y)}_{\text{"Shading" Image}}$$

Intrinsic image decomposition

- Consider a lambertian scene lit with directional light
- Pixel color:

$$\underbrace{I(x,y)}_{\text{Image}} = \underbrace{\rho(x,y)}_{\text{"Reflectance" image}} \underbrace{L_i \cos\theta_i(x,y)}_{\text{"Shading" Image}}$$

- Reflectance image depends only on object paint
- Shading image depends only on light and object shape (normals)

Integrating over incoming light

- General case

$$L_o = \int \rho(I,O) L\_i(I) \cos\theta_i(I)\, d\Omega$$

- Lambertian case

$$L_o = \rho \int L\_i(I) \cos\theta_i(I)\, d\Omega$$

Extension to color

- General case

$$L_o(\lambda) = \int \rho(I, O, \lambda) L\_i(I, \lambda) \cos\theta_i(I)\, d\Omega$$

- Lambertian case

$$L_o(\lambda) = \rho(\lambda) \int L\_i(I, \lambda) \cos\theta_i(I)\, d\Omega$$

Intrinsic image decomposition

$$I(x, y, \lambda) = \rho(x, y, \lambda) \int L_i(I, \lambda) \cos \theta_i(x, y, I) \ d\Omega$$

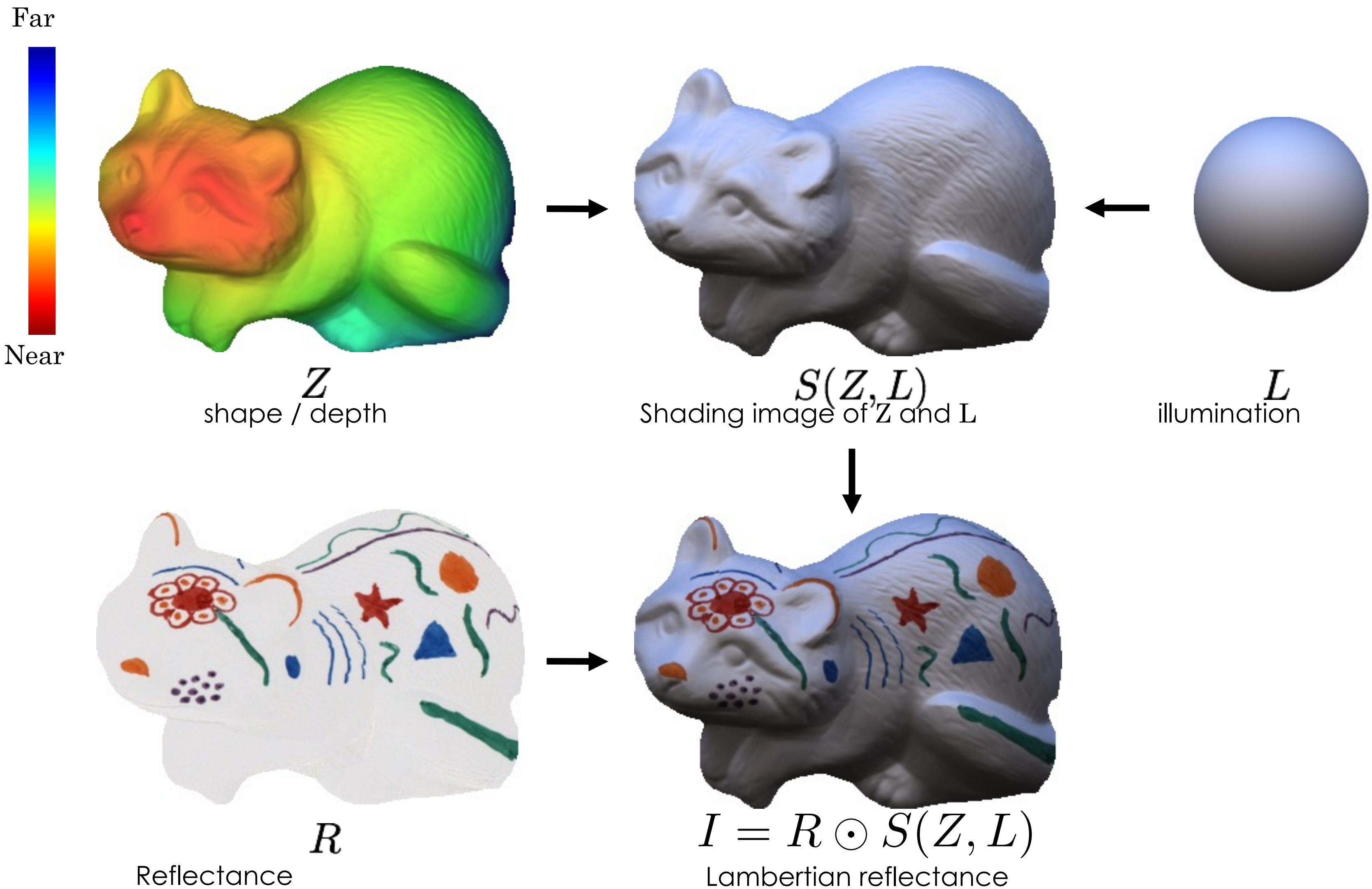
Image

"Reflectance" image, depends on paint only

"Shading" image depends on shape, lighting

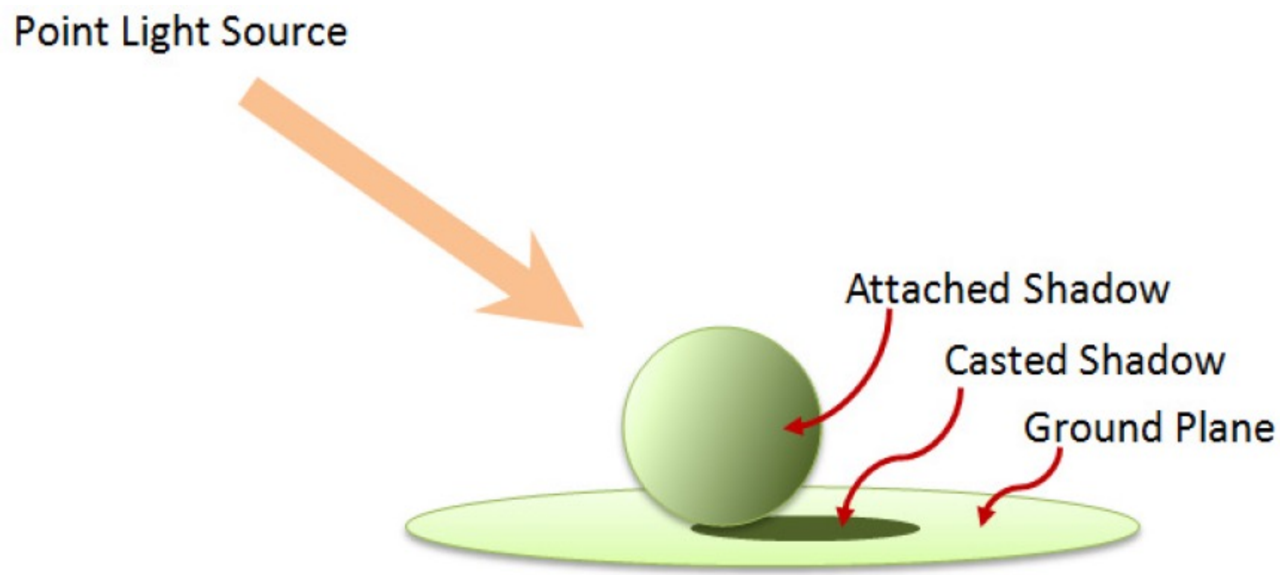
Lambertian surfaces

Lambertian surfaces



Slide credit: Jon Barron

Other lighting effects



How to create an image

- Create objects
  - Pick shape
  - Pick material
    - Is it Lambertian?
    - Pick albedo
- Place objects in coordinate system
- Place lights
- Place camera
- Take image

Photometric stereo

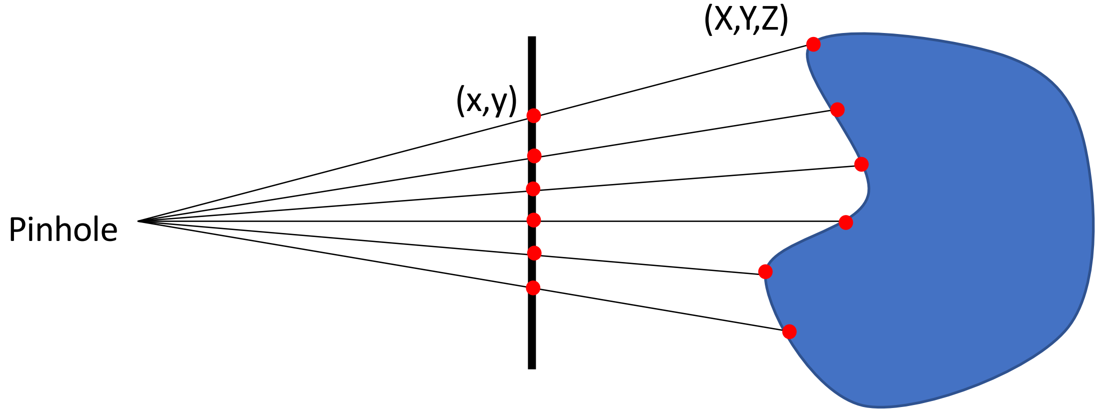
Till now: 3D structure from multiple cameras

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

Key Idea: use feature motion to understand shape

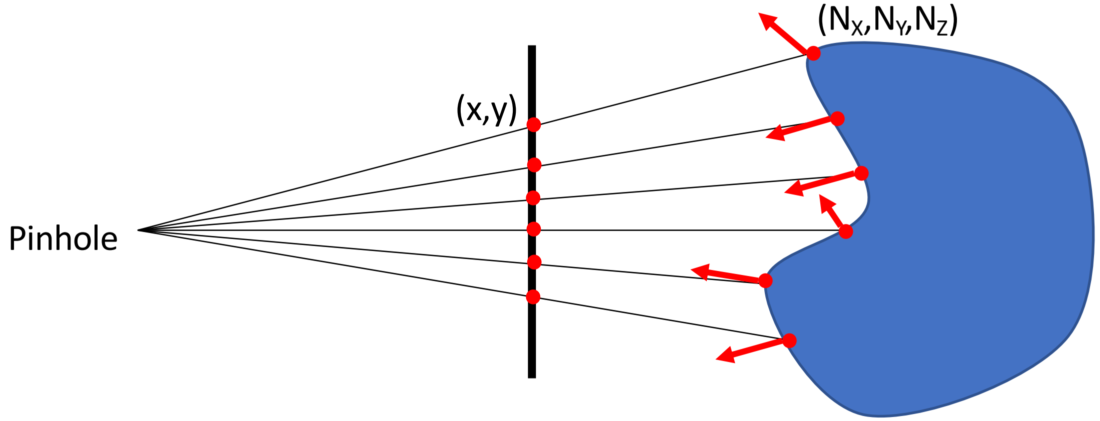
What does 3D structure mean?

- 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*



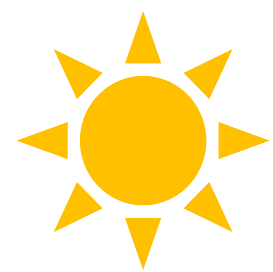
Not enough by itself to reveal absolute locations, but gives enough of a clue to object shape

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



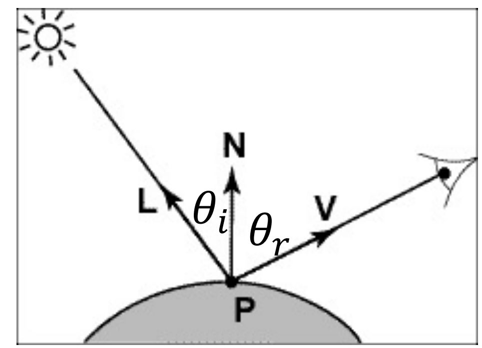
Lambertian surfaces

- For a lambertian surface:

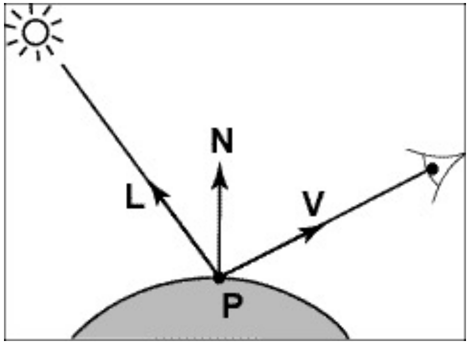
$$L_r = \rho L_i \cos\theta_i$$

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

- **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



Intrinsic Image Decomposition

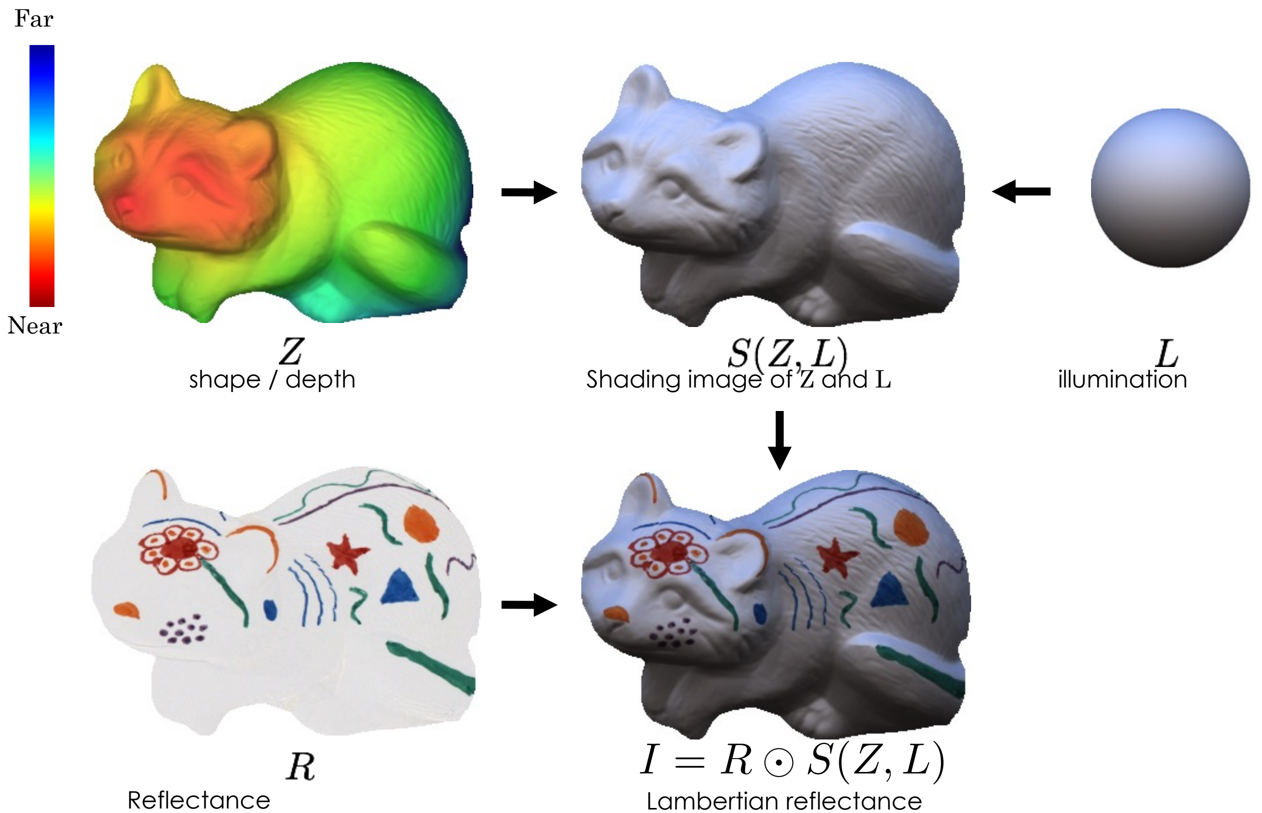
$$I(x,y) = \rho(x,y) L_i \mathbf{L} \cdot \mathbf{N}(x,y)$$

Reflectance image: albedo of surface corresponding to each pixel

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

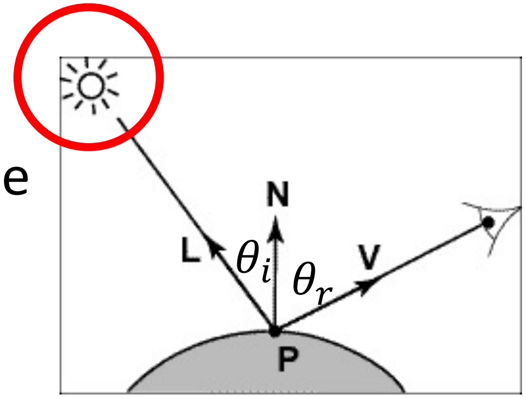
Lambertian surfaces

Lambertian surfaces



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



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*

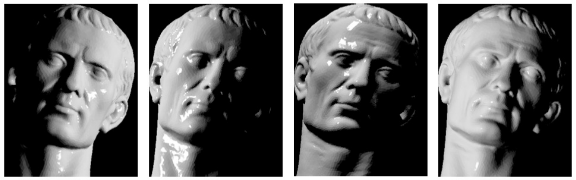


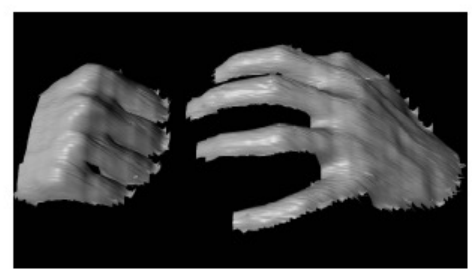
Image credit: Wikipedia

Reconstruction with active sensors

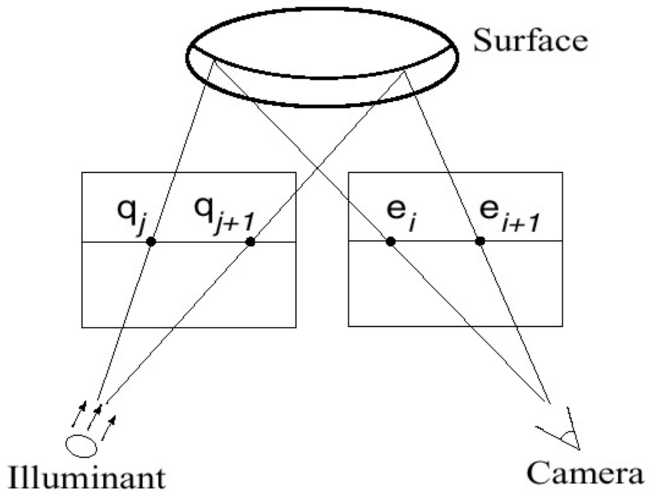
Active stereo

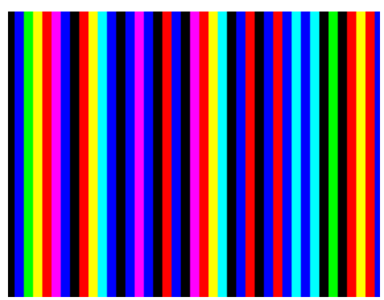
- Stereo system where one camera is replaced with a projector.
- Projector outputs infrared pattern that makes correspondence easier.
  - Image 1 is known pattern of stripes (output of projector)
  - Image 2 is captured (see below)

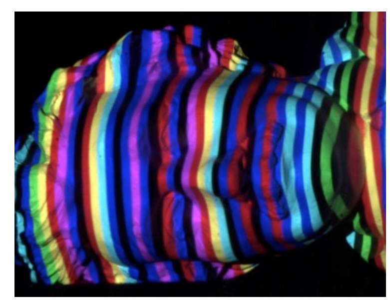




Active stereo with structured light

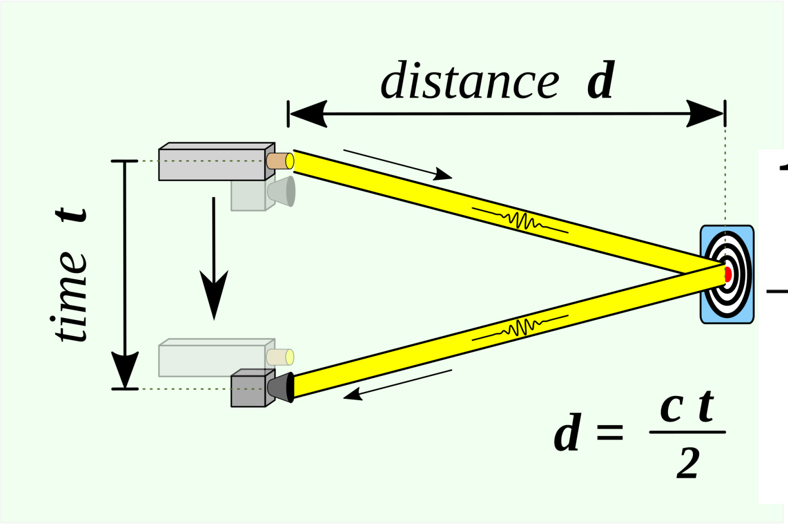


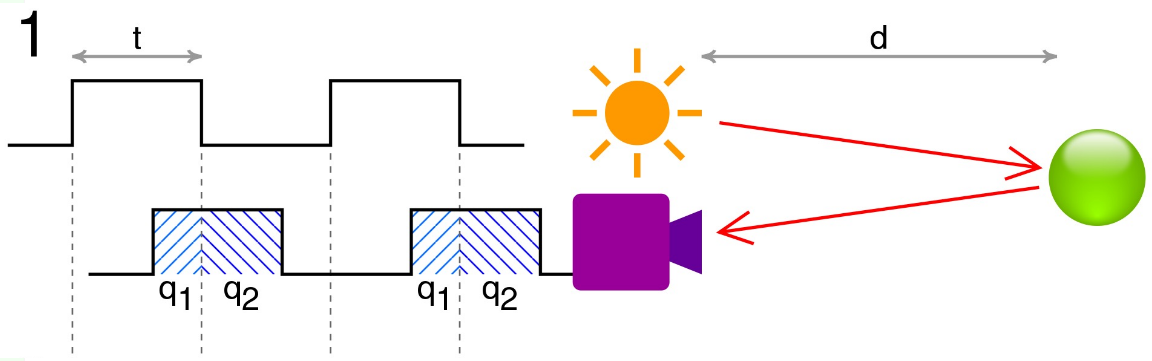




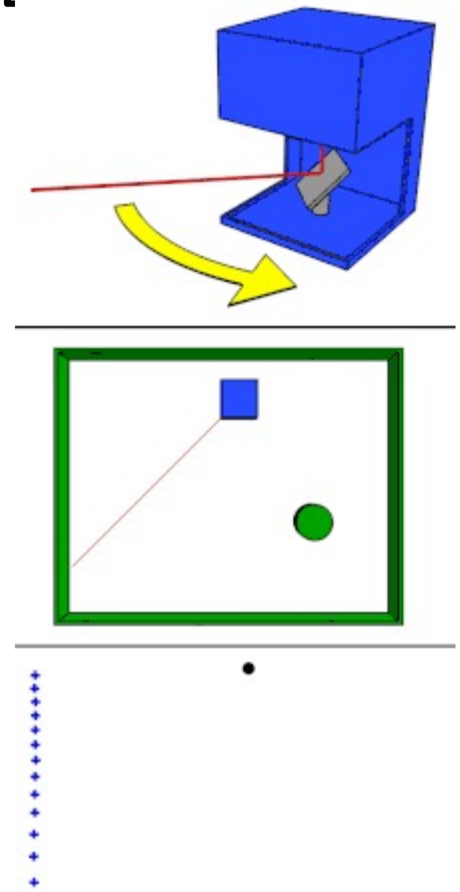
L. Zhang, B. Curless, and S. M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. *3DPVT* 2002

Time-of-flight sensors





Time-of-flight sensor - LiDAR



Active photometric stereo

(a) (b) (c)

Figure 1. (a) A cookie is pressed against the skin of an elastomer block. (b) The skin is distorted, as shown in this view from beneath. (c) The cookie's shape can be measured using photometric stereo and rendered at a novel viewpoint.

Active photometric sstereo

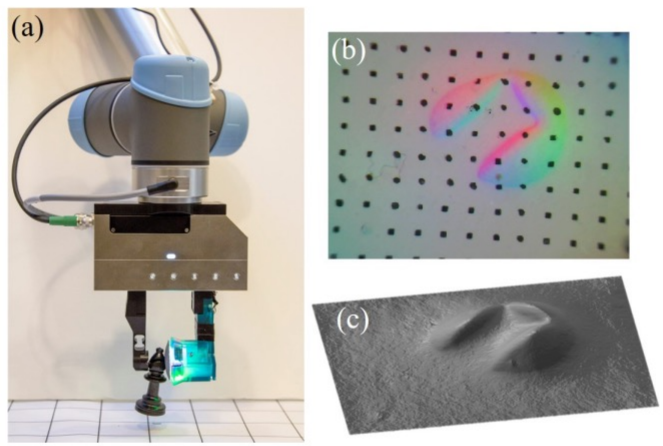


Fig. 1. (a) A parallel gripper WSG-50 with the new GelSight sensor gripping a chess Bishop.(b) Image captured by GelSight sensor during the grasp (c) The reconstructed 3D geometry from (b)