Radiometry and Reflectance

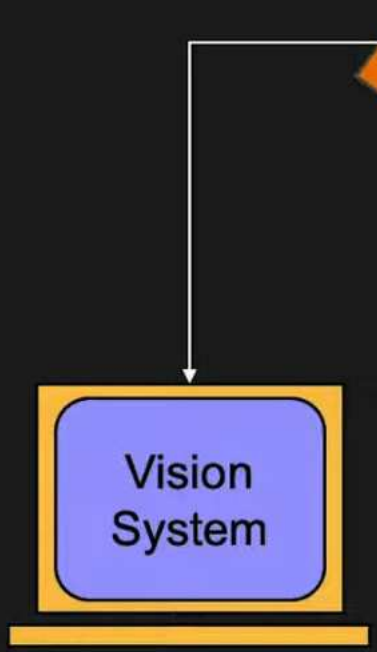
Shree K. Nayar

Columbia University

Topic: Radiometry and Reflectance, Module: Reconstruction I

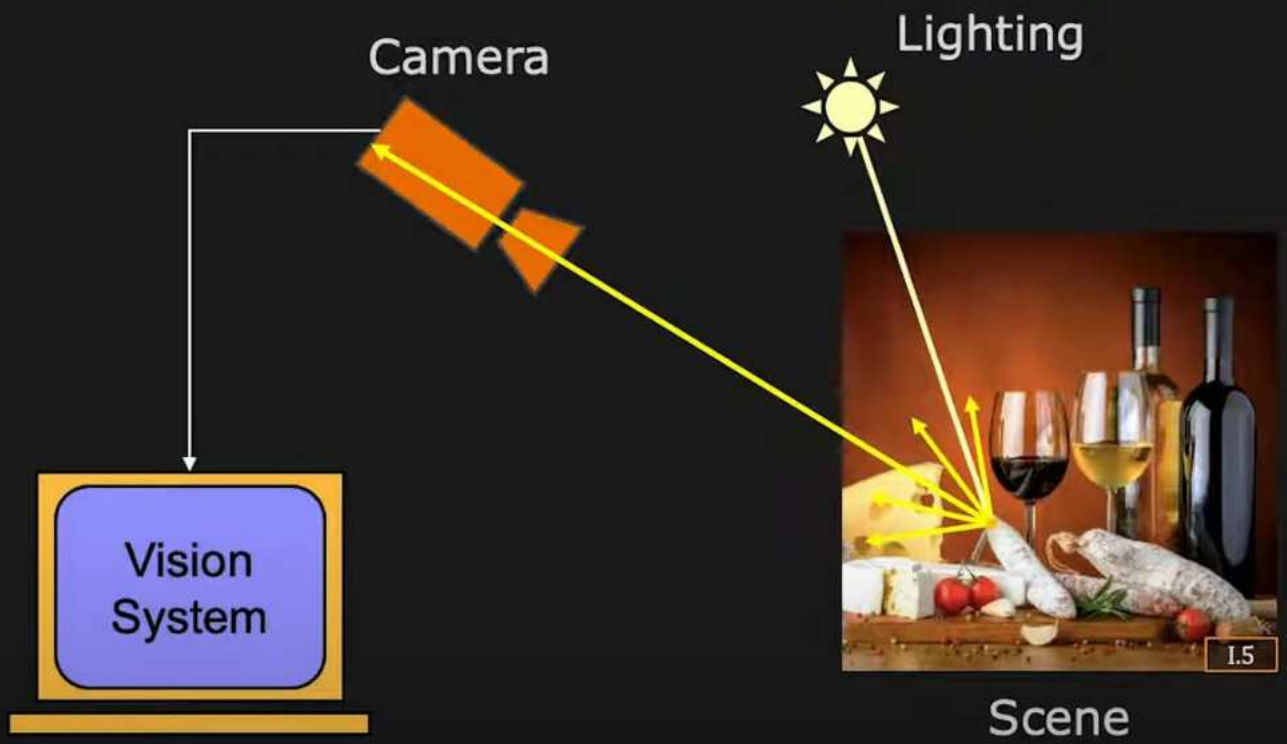
First Principles of Computer Vision

From 2D to 3D



Scene

From 2D to 3D



Radiometry and Reflectance

To interpret image intensities, we need to understand Radiometric Concepts and Reflectance Properties.

Image Intensity

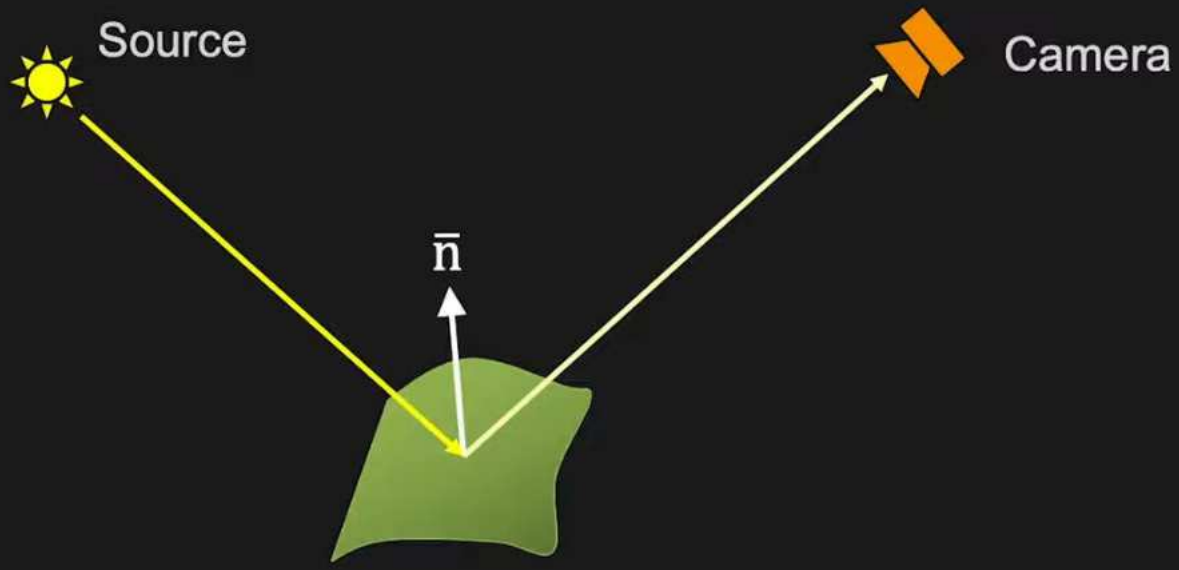


Image Intensity = f (Illumination,
Surface Orientation,
Surface Reflectance)

Image intensity understanding is under-constrained!

Radiometry and Reflectance

To interpret image intensities, we need to understand Radiometric Concepts and Reflectance Properties.

Topics:

(1) Radiometric Concepts

(2) Surface Radiance and Image Irradiance

(3) BRDF: Bidirectional Reflectance Distribution Function

(4) Reflectance Models

Radiometry and Reflectance

To interpret image intensities, we need to understand Radiometric Concepts and Reflectance Properties.

Topics:

(1) Radiometric Concepts

(2) Surface Radiance and Image Irradiance

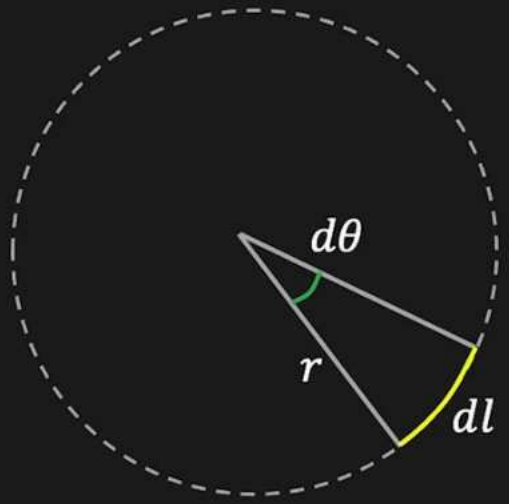
(3) BRDF: Bidirectional Reflectance Distribution Function

(4) Reflectance Models

(5) Dichromatic Model

Concept: Angle (2D)

$$d\theta = \frac{dl}{r}$$



Unit: radian (rad)

$d\theta$ is dimensionless

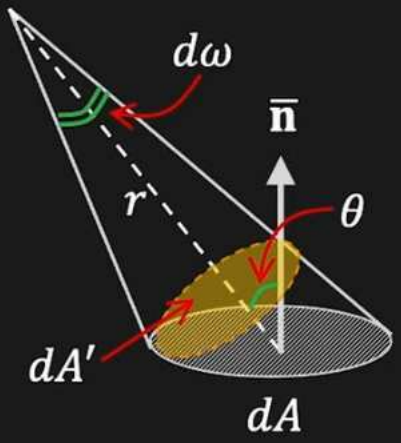
However, radian is used as its unit to distinguish from other dimensionless quantities.

Concept: Solid Angle (3D)

$$d\omega = \frac{dA'}{r^2} = \frac{dA\ \cos\theta}{r^2}$$

Unit: steradian (sr)

$d\omega$ is dimensionless



dA': Foreshortened Area

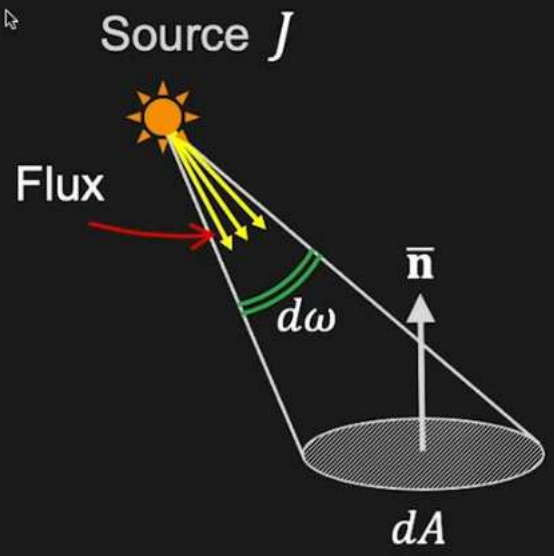
Solid angle subtended by a hemisphere? 2π

Concept: Radiant Intensity

Light flux emitted per unit solid angle

$$J = \frac{d\Phi}{d\omega}$$

Unit: W sr^{-1}

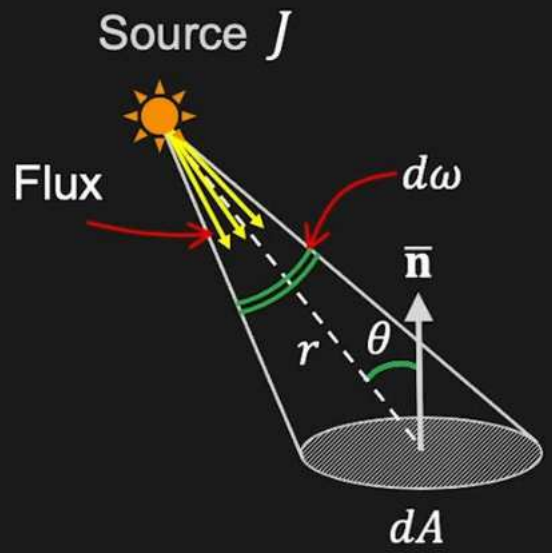


Concept: Surface Irradiance

Light flux incident per unit surface area

$$E = \frac{d\Phi}{dA}$$

Unit: W m^{-2}

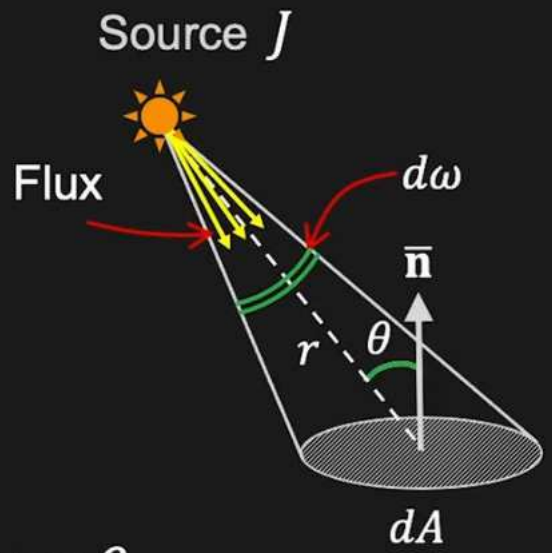


Concept: Surface Irradiance

Light flux incident per unit surface area

$$E = \frac{d\Phi}{dA}$$

Unit: W m^{-2}



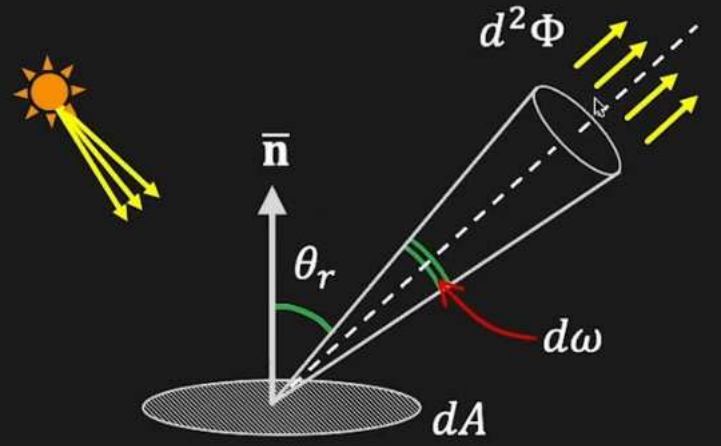
$$E = \frac{J\, d\omega}{dA} = \frac{J \dfrac{dA \cos\theta}{r^2}}{dA} = \frac{J \cos\theta}{r^2}$$

Concept: Surface Radiance

Light flux emitted per unit foreshortened area per unit solid angle

$$L = \frac{d^2\Phi}{(dA\cos\theta_r)d\omega}$$

Unit: W m^{-2} sr^{-1}



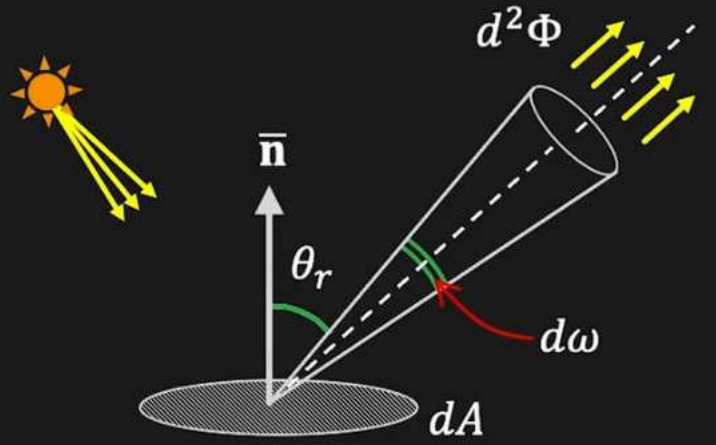
$dA\cos\theta_r$: Foreshortened Area

Concept: Surface Radiance

Light flux emitted per unit foreshortened area per unit solid angle

$$L = \frac{d^2\Phi}{(dA\cos\theta_r)d\omega}$$

Unit: W m^{-2} sr^{-1}

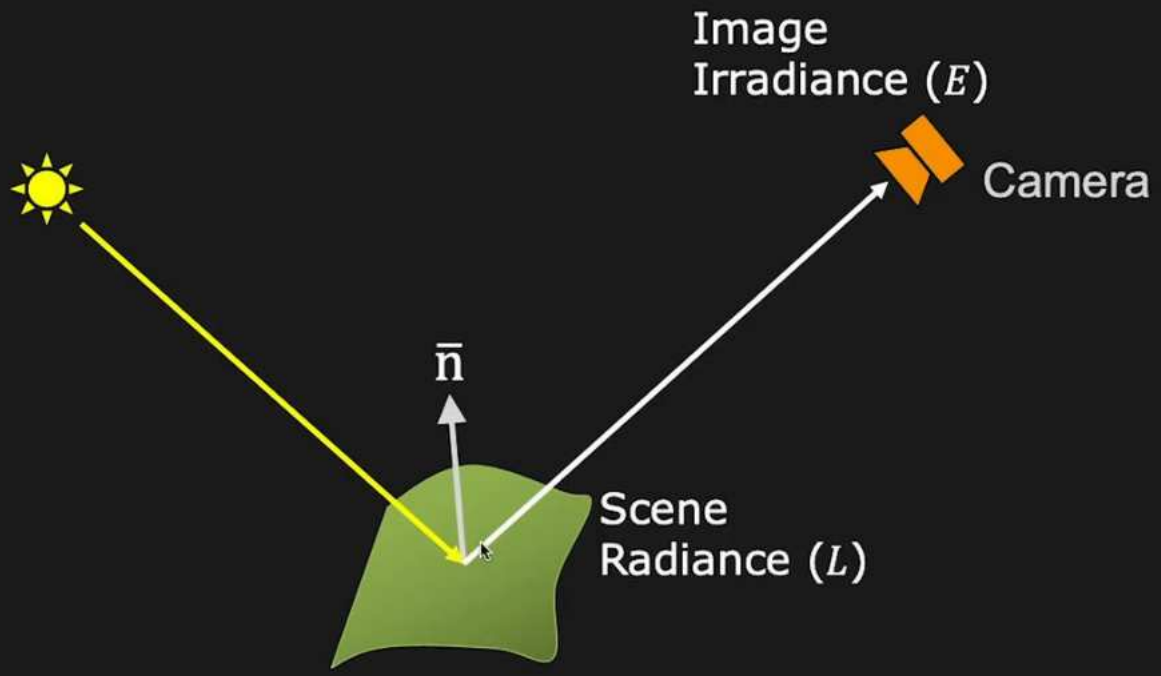


Note:

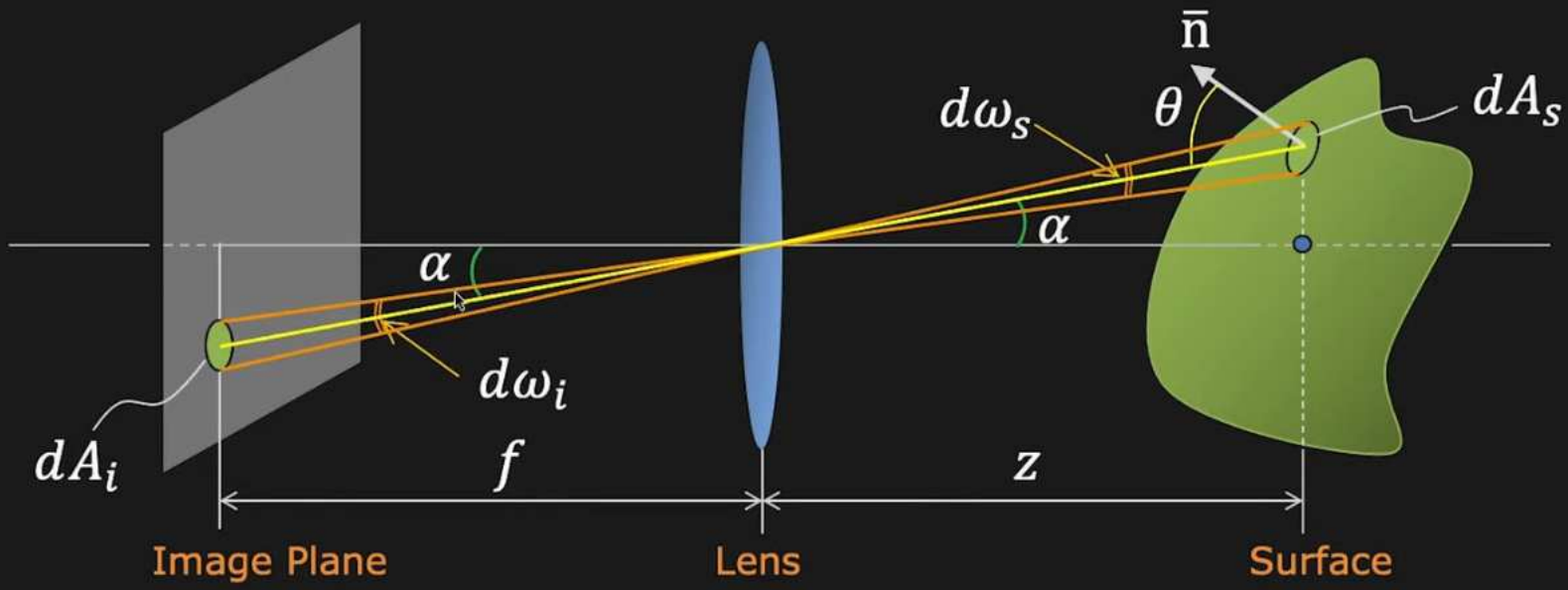
$dA\cos\theta_r$: Foreshortened Area

- L depends on direction θ_r : $L(\theta_r)$
- Surface can radiate into the whole hemisphere.
- L depends on reflectance properties of surface

Scene Radiance and Image Irradiance



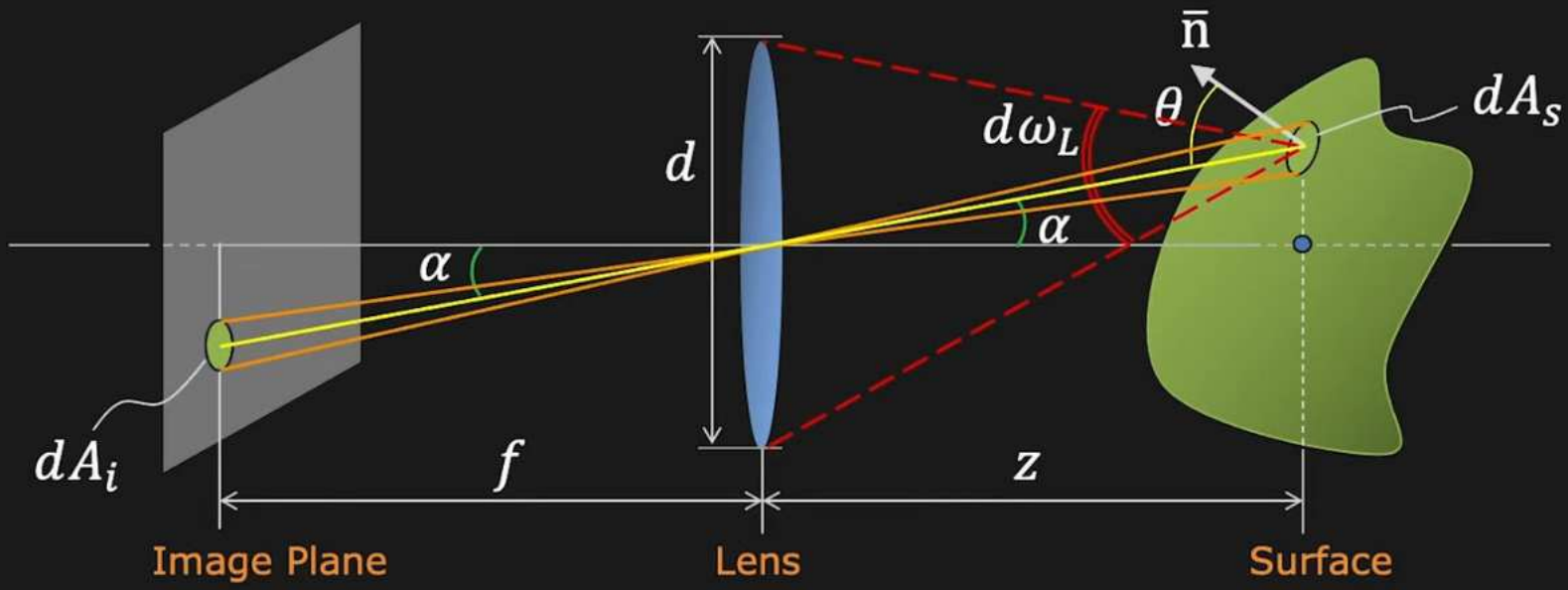
Scene Radiance and Image Irradiance



Solid Angles: $d\omega_i = d\omega_s$

$$\frac{dA_i \cos\alpha}{(f/\cos\alpha)^2} = \frac{dA_s \cos\theta}{(z/\cos\alpha)^2}$$

Scene Radiance and Image Irradiance

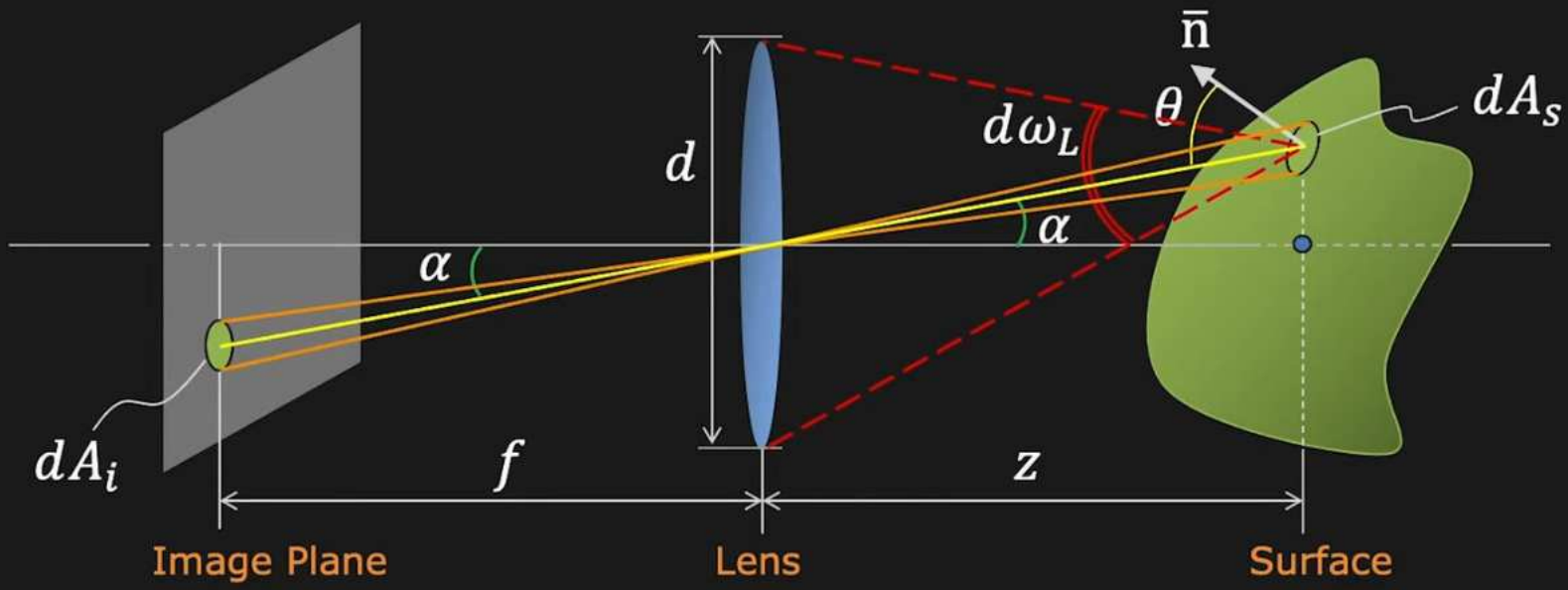


Solid Angle subtended by the lens:

$$d\omega_L = \frac{\pi d^2}{4} \frac{\cos\alpha}{(z/\cos\alpha)^2}$$

Equation (2)

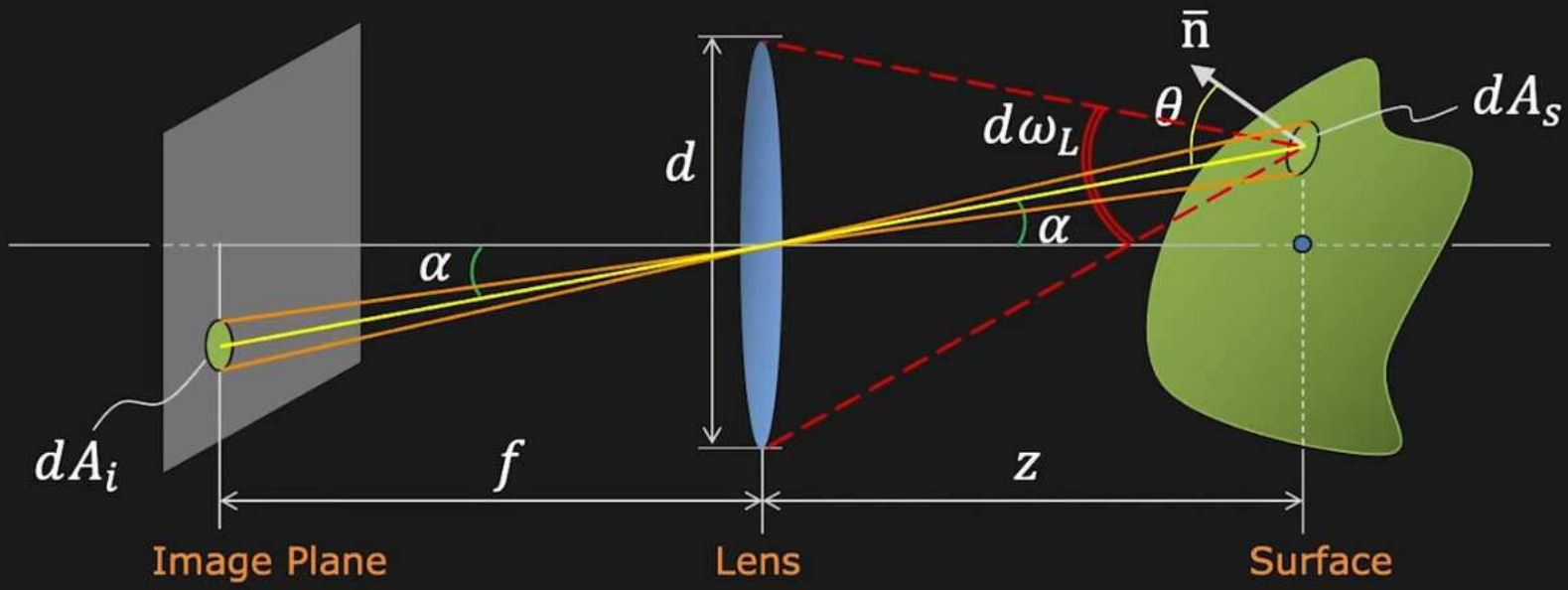
Scene Radiance and Image Irradiance



Energy Conservation:

Flux received by lens from dA_s = Flux projected onto dA_i

Scene Radiance and Image Irradiance



Scene Radiance:

$$L = \frac{d^2\Phi}{(dA_s \cos\theta)d\omega_L}$$

Flux received by lens from dA_s

$$d^2\Phi = L\,(dA_s \cos\theta)d\omega_L$$

Equation (3)

Scene Radiance and Image Irradiance

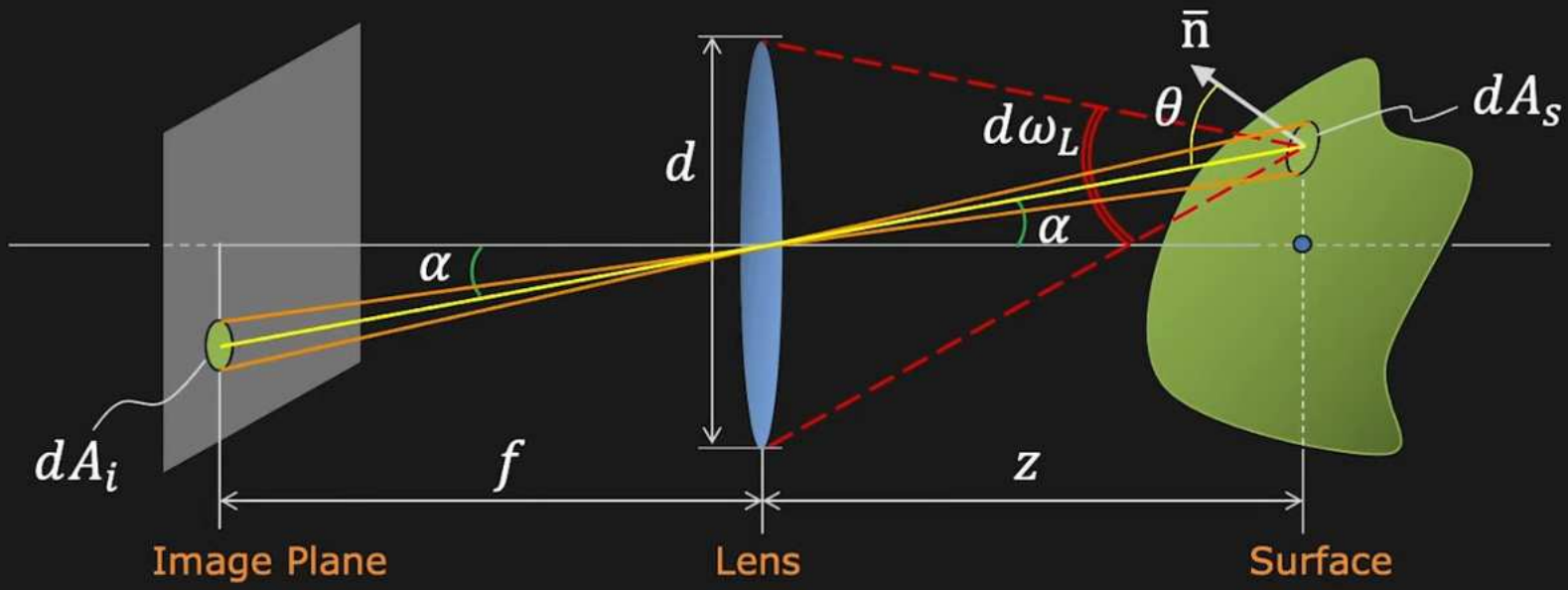


Image Irradiance: Flux projected onto dA_i

$$E = \frac{d\Phi}{dA_i} \quad \Rightarrow \quad \boxed{d\Phi = E\, dA_i}$$

Equation (4)

Scene Radiance and Image Irradiance

Equation (1)

$$\frac{dA_s}{dA_i} = \frac{\cos\alpha}{\cos\theta}\left(\frac{z}{f}\right)^2$$

Equation (2)

$$d\omega_L = \frac{\pi d^2}{4}\frac{\cos\alpha}{(z/\cos\alpha)^2}$$

Equation (3)

$$d^2\Phi = L\,(dA_s\cos\theta)d\omega_L$$

Equation (4)

$$d\Phi = E\,dA_i$$

Scene Radiance and Image Irradiance

Equation (1)

$$\frac{dA_s}{dA_i} = \frac{\cos\alpha}{\cos\theta}\left(\frac{z}{f}\right)^2$$

Equation (2)

$$d\omega_L = \frac{\pi d^2}{4}\frac{\cos\alpha}{(z/\cos\alpha)^2}$$

Equation (3)

$$d^2\Phi = L\,(dA_s\cos\theta)d\omega_L$$

Equation (4)

$$d\Phi = E\,dA_i$$

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

Scene Radiance and Image Irradiance

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

Image Irradiance

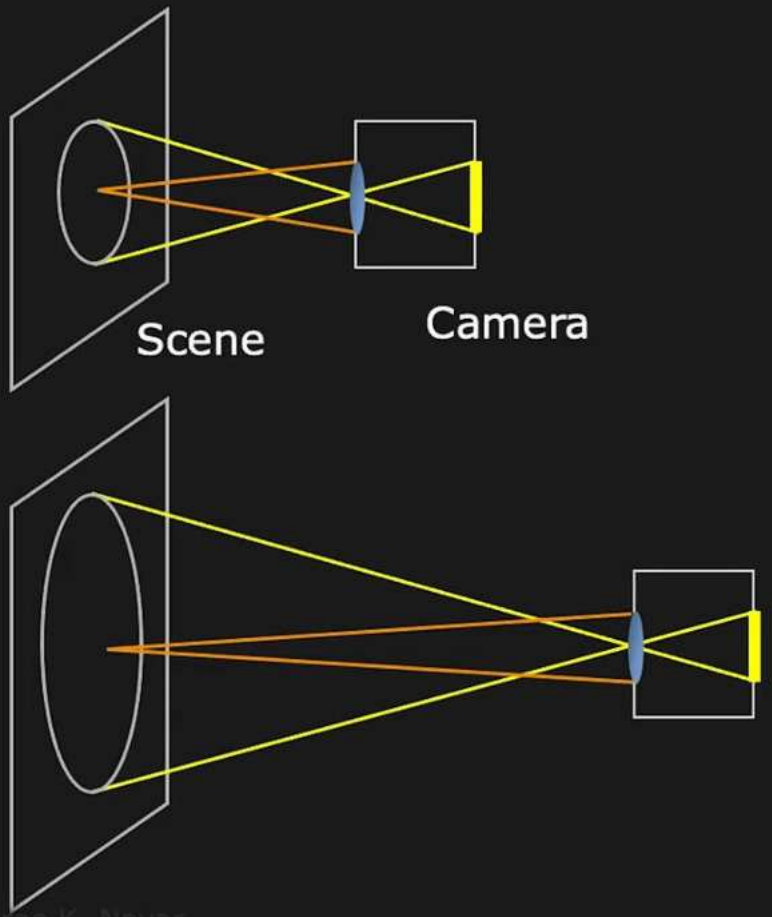
Scene Radiance

1/Effective F-number
(since f is *effective* focal length)

- Image Irradiance is proportional to Scene Radiance
- Image brightness falls off from image center as $\cos^4\alpha$
- For small fields of view, effects of $\cos^4\alpha$ are small

Scene Radiance and Image Irradiance

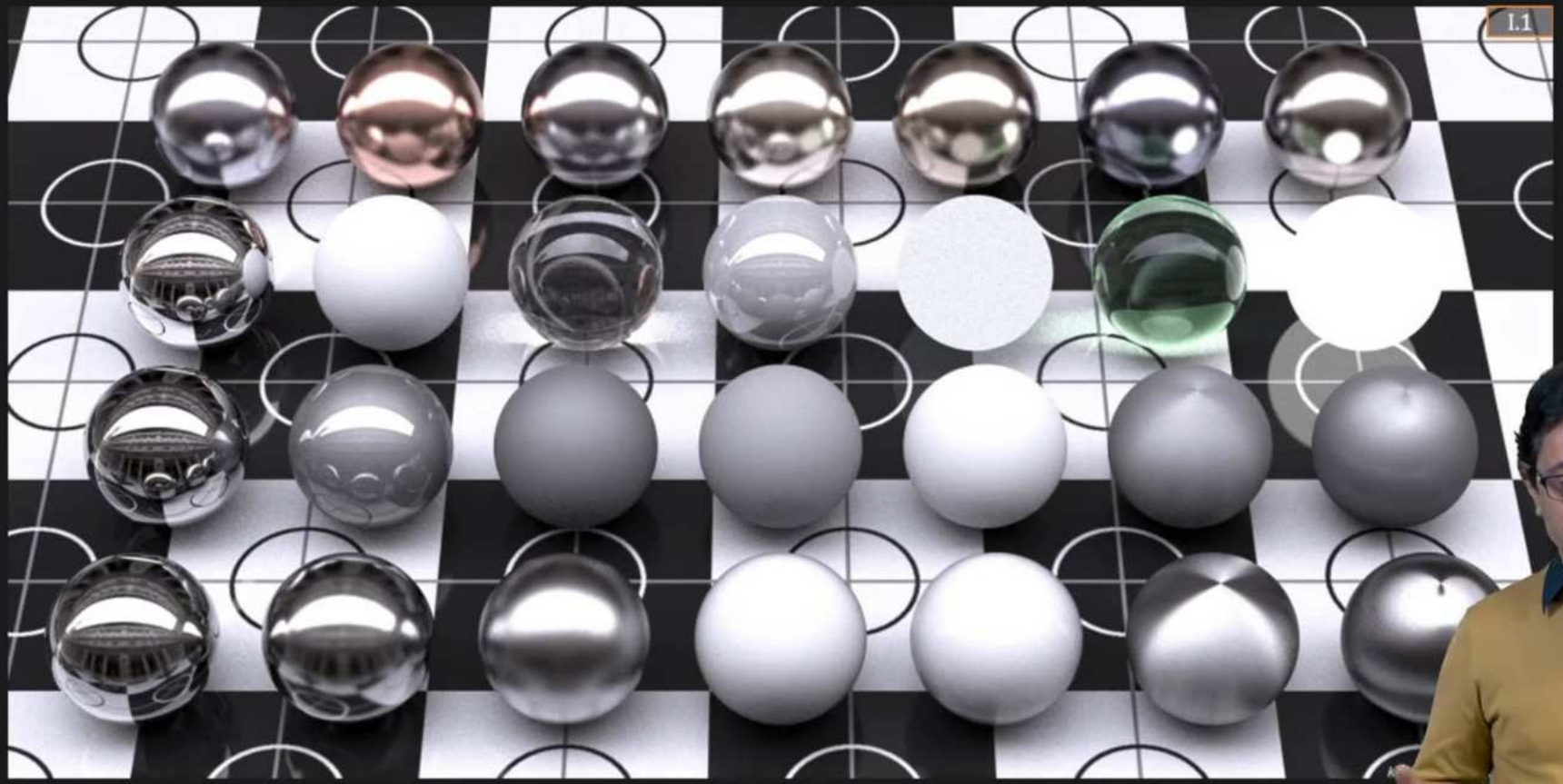
Does image brightness vary with scene depth? NO



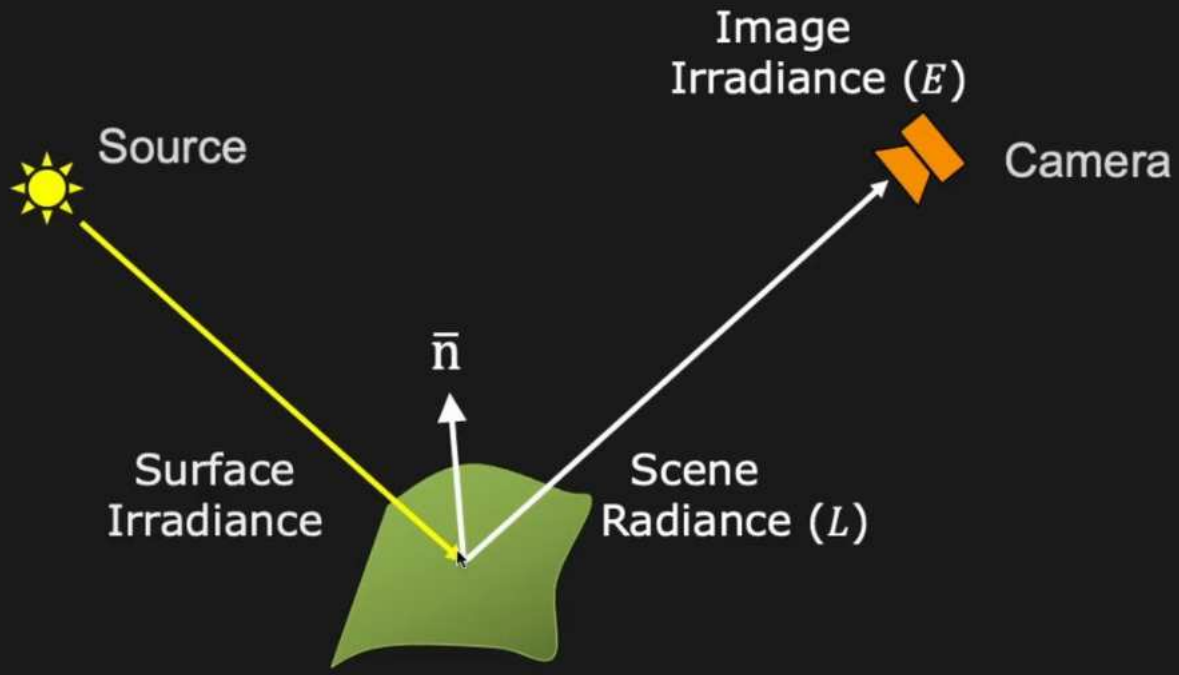
$$E = L\frac{\pi}{4}\left(\frac{d}{f}\right)^2 \cos^4\alpha$$

- Larger the scene depth, larger the area of light accumulation.
- Larger the scene depth, smaller the solid angle subtended by each point onto the lens, and hence less light from each point.

Surface Appearance



Surface Reflection

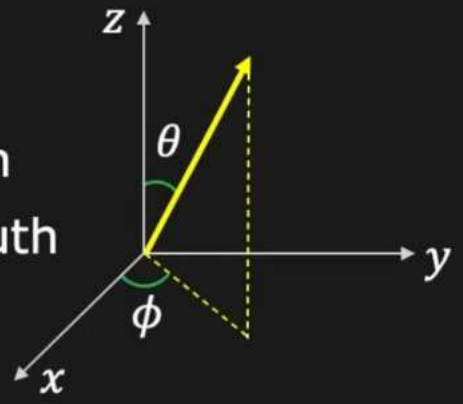


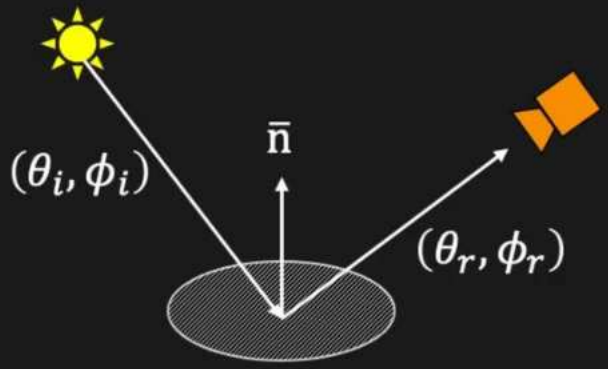
Surface reflection depends on both the viewing and illumination directions.

Bidirectional Reflectance Distribution Function

θ: zenith

ϕ: azimuth





$E(\theta_i, \phi_i)$: Irradiance due to source in direction (θ_i, ϕ_i)

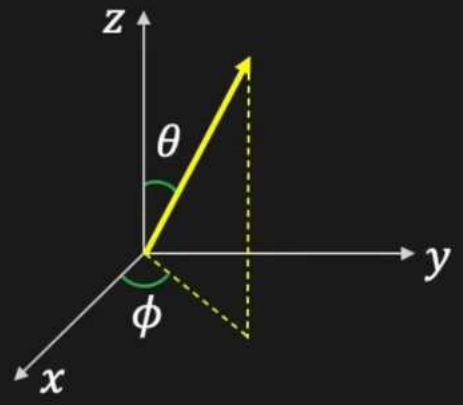
$L(\theta_r, \phi_r)$: Radiance of surface in direction (θ_r, ϕ_r)

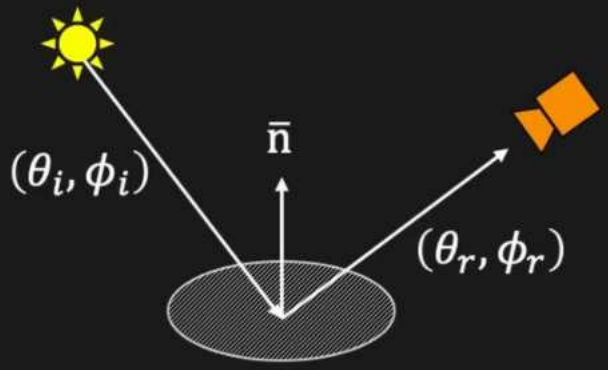
BRDF: $$f(\theta_i, \phi_i, \theta_r, \phi_r) = \frac{L(\theta_r, \phi_r)}{E(\theta_i, \phi_i)}$$

Unit: 1/sr

[Nicodemus 1977]

Properties of BRDF





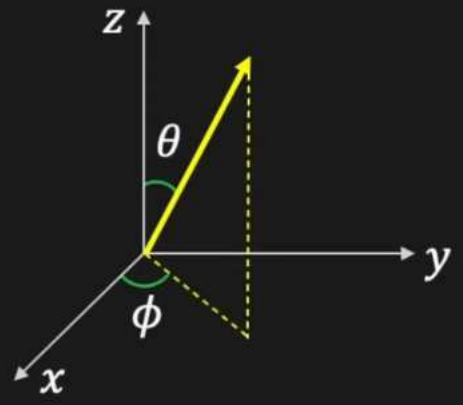
Non-Negative:

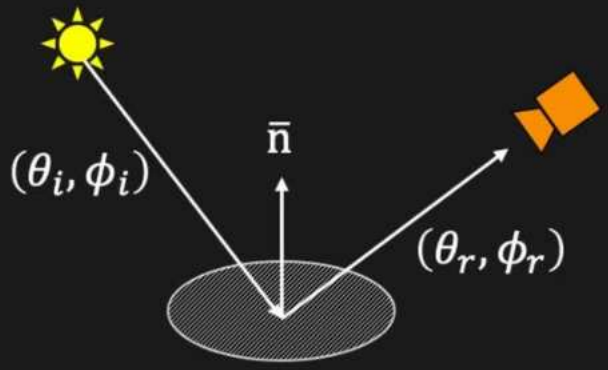
$$f(\theta_i, \phi_i, \theta_r, \phi_r) > 0$$

Helmholtz Reciprocity:

$$f(\theta_i, \phi_i, \theta_r, \phi_r) = f(\theta_r, \phi_r, \theta_i, \phi_i)$$

BRDF of Isotropic Surfaces

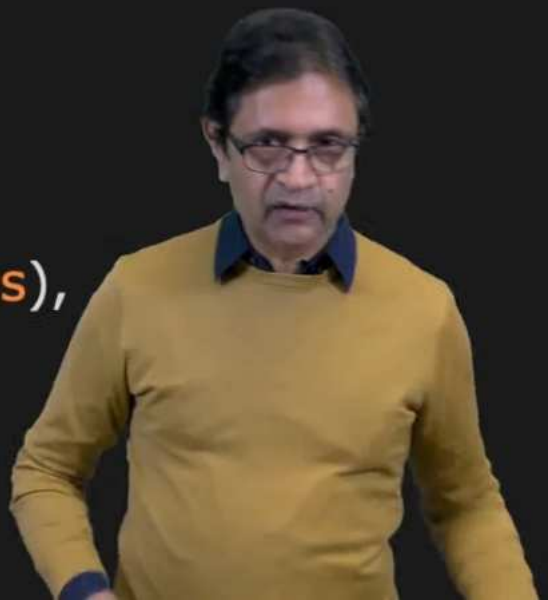




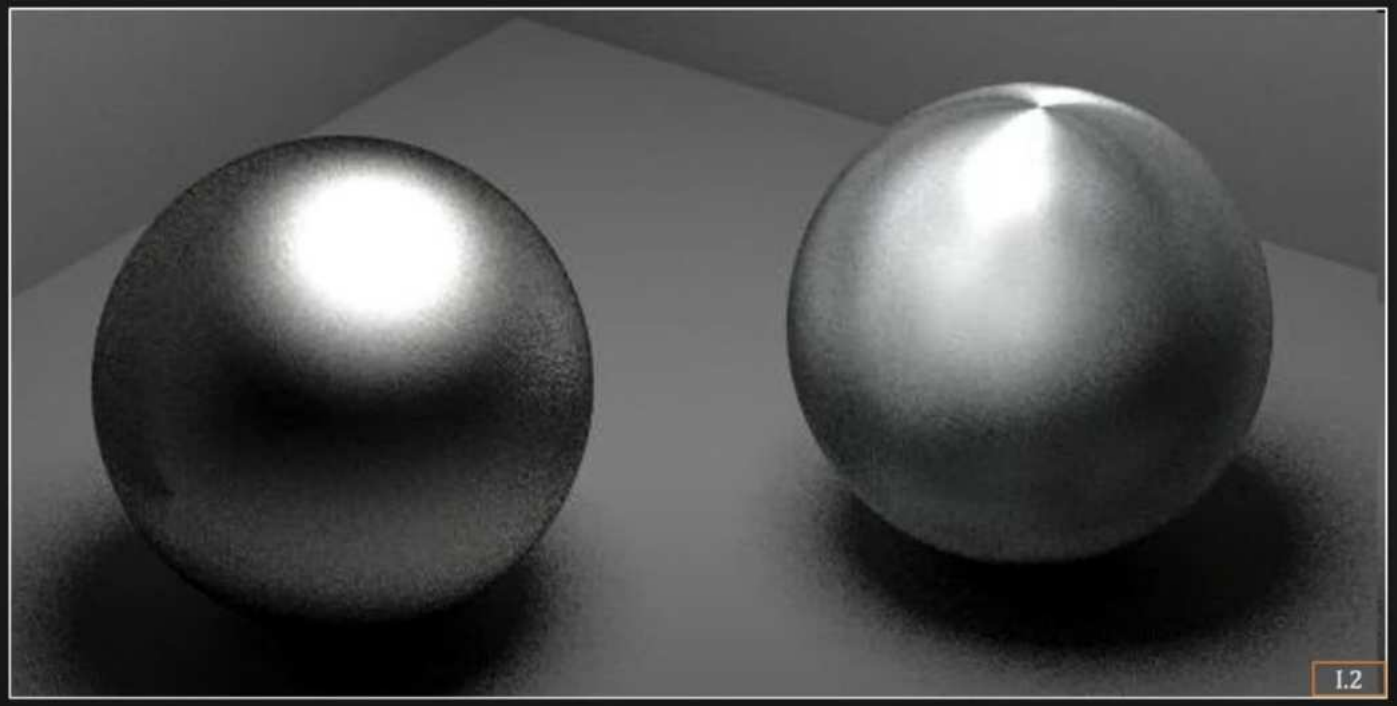
In general, BRDF is a 4-D function: $f(\theta_i, \phi_i, \theta_r, \phi_r)$

For rotationally symmetric reflectance (Isotropic Surfaces), BRDF is a 3-D function:

$$f(\theta_i, \theta_r, \phi_i - \phi_r)$$



Isotropic BRDF and Anisotropic BRDF



Isotropic BRDF

Anisotropic BRDF

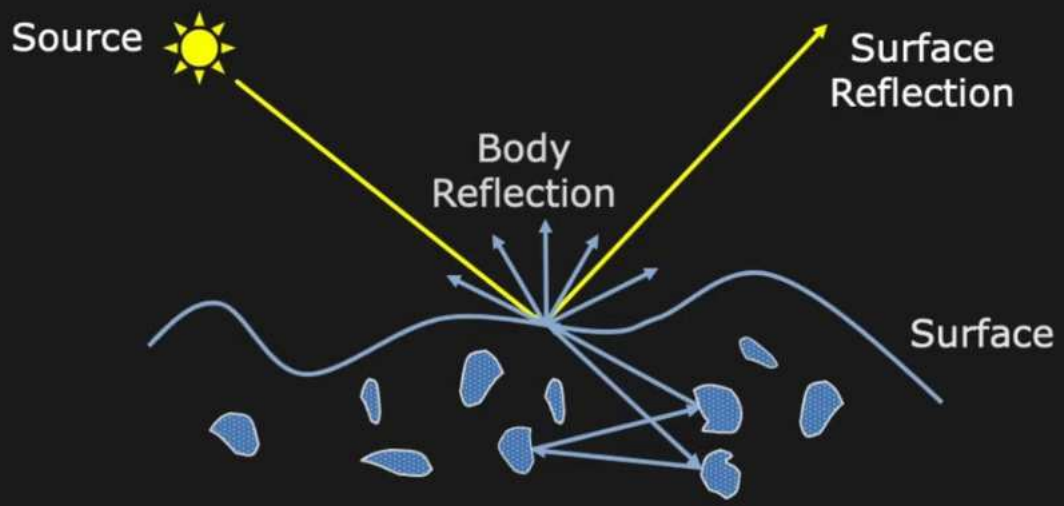
[Westin 1992]

Anisotropic BRDFs in the Real World

Butterfly wings

Peacock feathers

Reflection Mechanisms



Surface Reflection

- Specular Reflection
- Glossy Appearance
- Smooth Surfaces (e.g., mirror, glass)

Body Reflection

- Diffuse Reflection
- Matte Appearance
- Non-Homogeneous Medium (e.g., clay, paper)

Examples

Body Reflection:

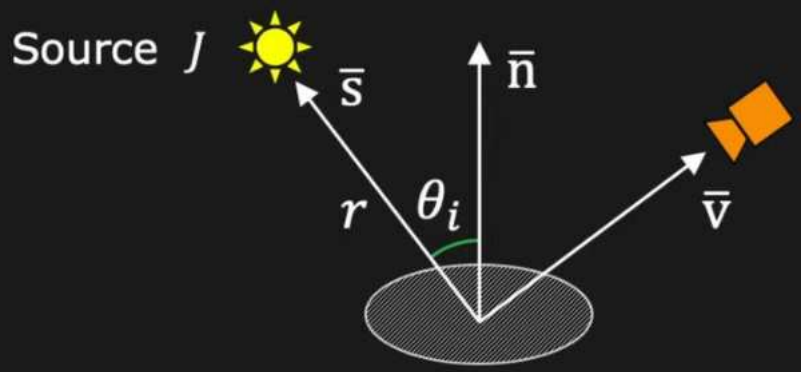
Surface Reflection:

Hybrid Reflection
(Body + Surface):

[Nayar 1991]

Lambertian Model (Body)

Surface appears equally bright from ALL directions



Lambertian BRDF:

$$f(\theta_i, \phi_i, \theta_r, \phi_r) = \frac{\rho_d}{\pi}$$

ρ_d : Albedo

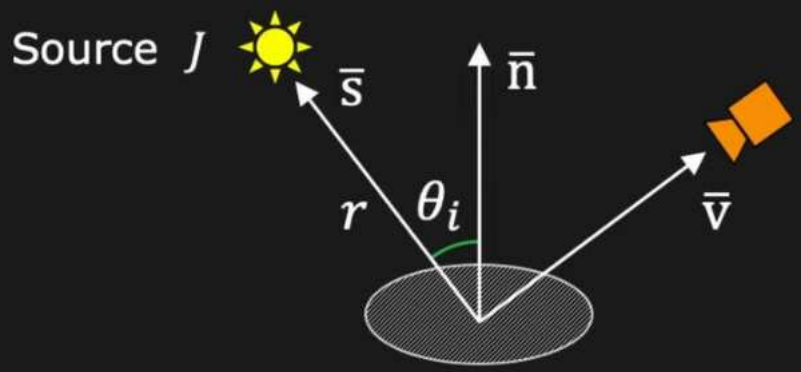
$(0 \leq \rho_d \leq 1)$

Commonly used in Vision and Graphics

[Lambert 1760]

Lambertian Model (Body)

Surface appears equally bright from ALL directions



Radiance is proportional to Irradiance:

$$L = \frac{\rho_d}{\pi} E \qquad E = \frac{J \cos\theta_i}{r^2} = \frac{J}{r^2}(\bar{n} \cdot \bar{s})$$

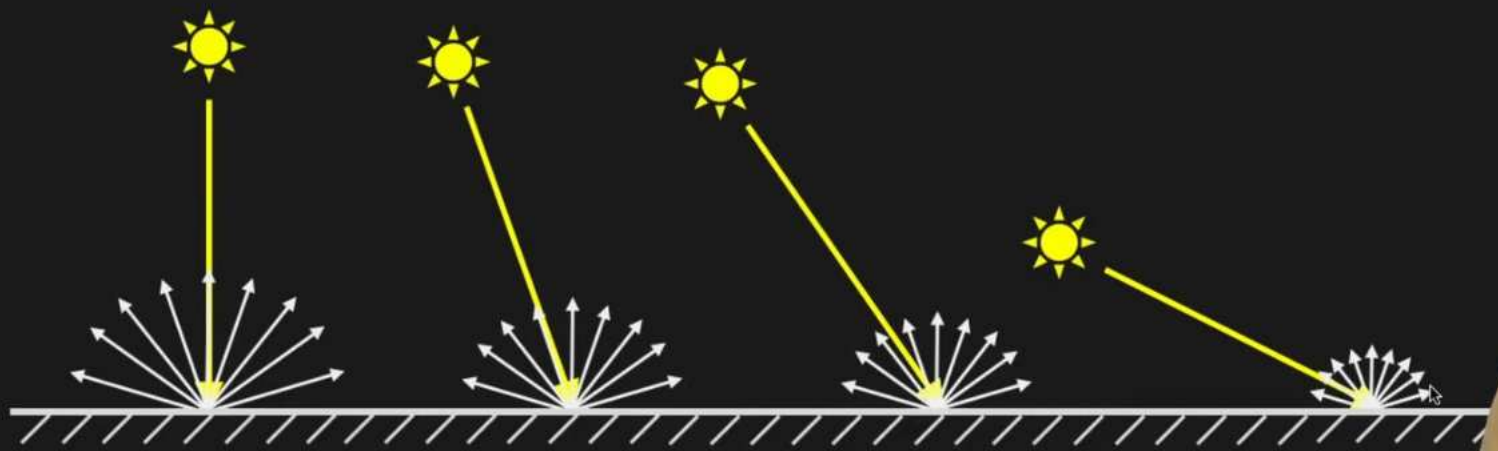
$$\boxed{L = \frac{\rho_d}{\pi} \frac{J}{r^2}(\bar{n} \cdot \bar{s})}$$

[Lambert 1760]

Lambertian Model

L is independent of viewing direction

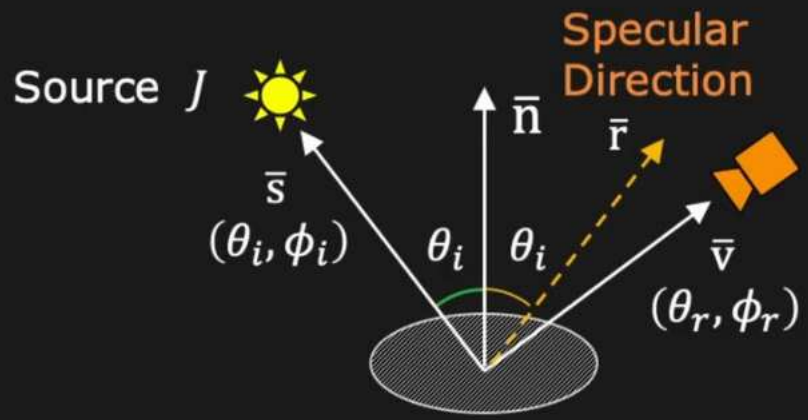
$$L = \frac{\rho_d}{\pi} \frac{J}{r^2} (\bar{n} \cdot \bar{s})$$



Lambertian Surface

Ideal Specular Model (Surface)

Perfect Mirrors: All incident energy is reflected in a single direction.



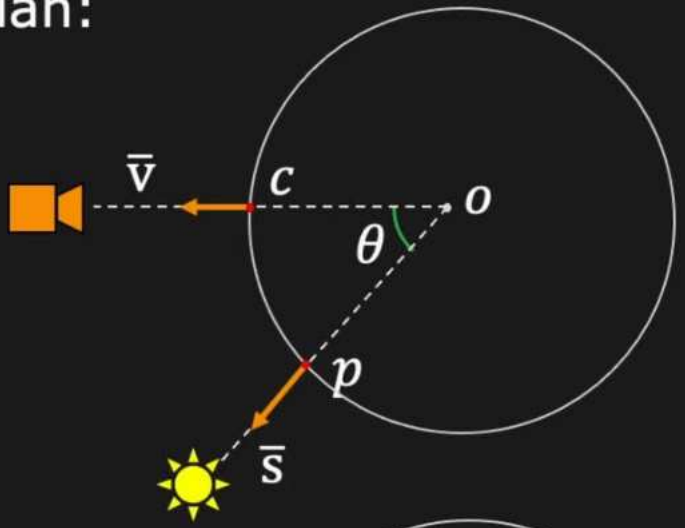
Mirror BRDF:

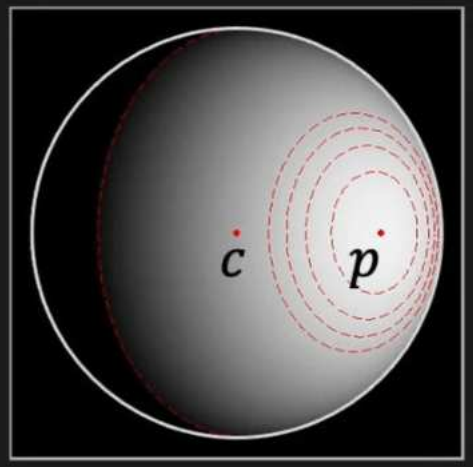
$$f(\theta_i, \phi_i, \theta_r, \phi_r) = \frac{\delta(\theta_i - \theta_r)\,\delta(\phi_i + \pi - \phi_r)}{\cos\theta_i \sin\theta_i}$$

Viewer receives light only when $\bar{v} = \bar{r}$

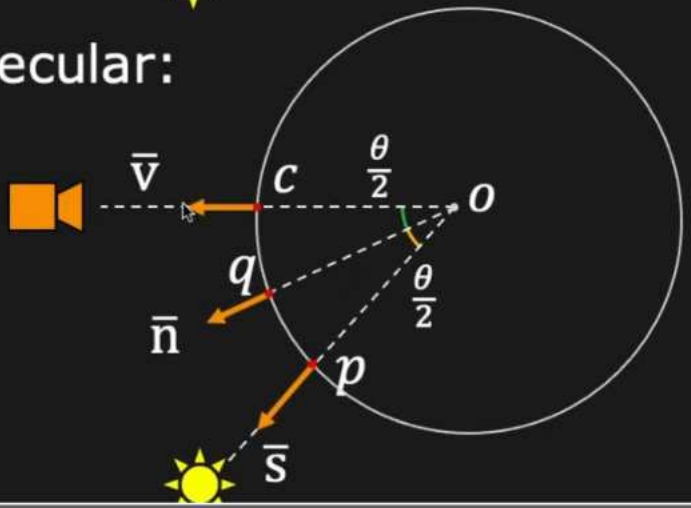
Examples

Lambertian:

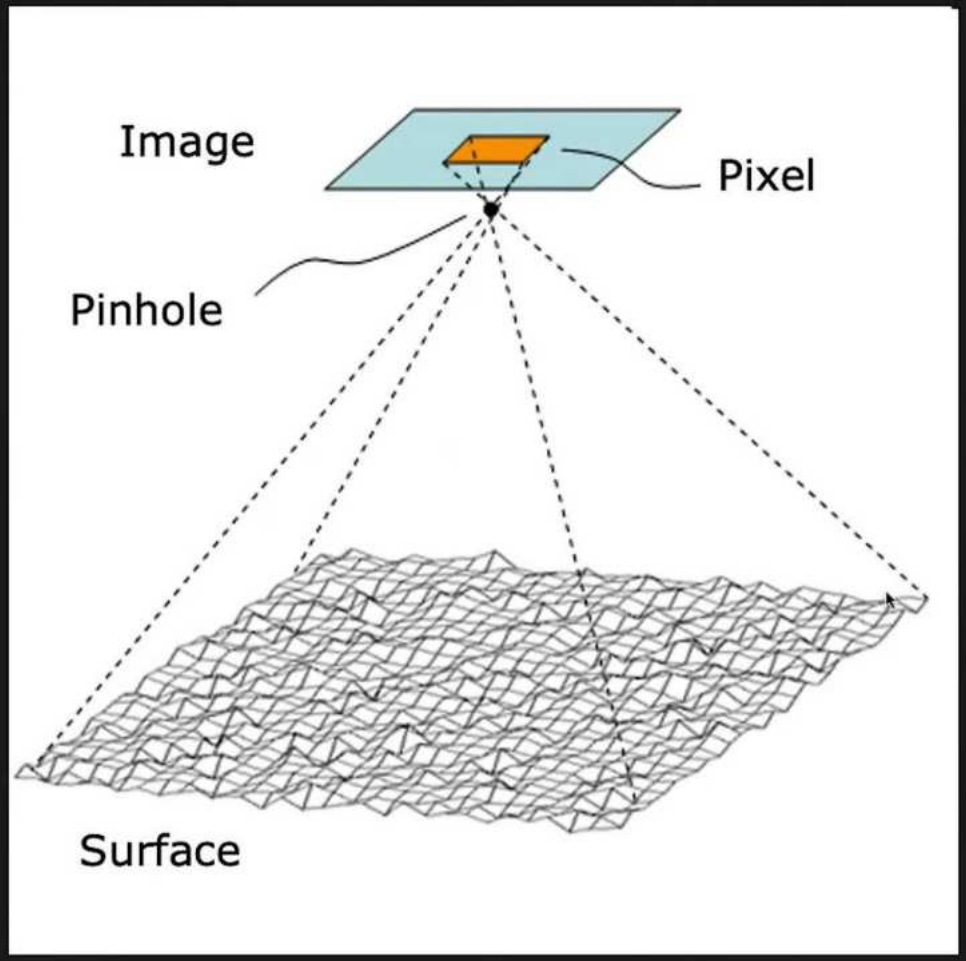




Ideal Specular:



Surface Roughness



Modeling Surface Roughness

Micro-Facet Structure for Rough Surfaces:

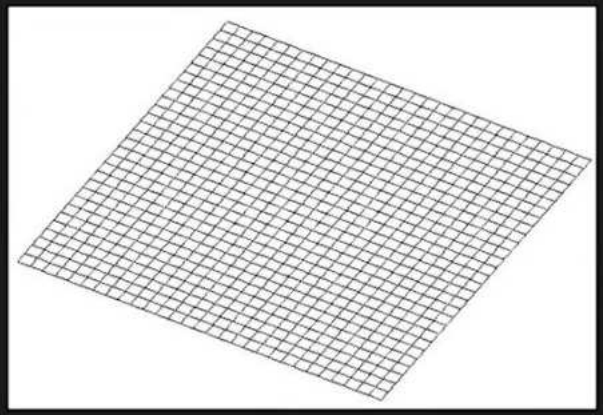
$\bar{n}$: Mean Orientation

Gaussian Micro-Facet Model:

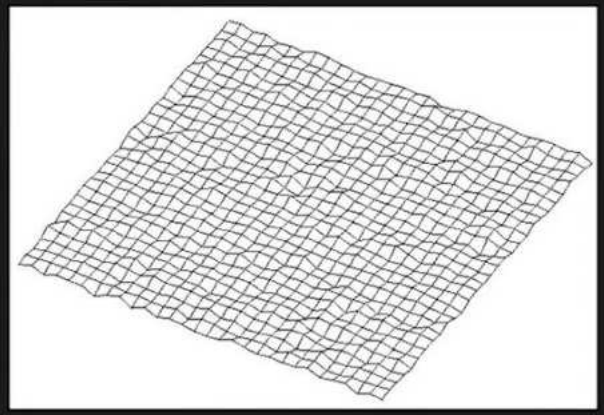
$$p(\alpha, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\alpha^2}{2\sigma^2}}$$

where σ: Roughness Parameter

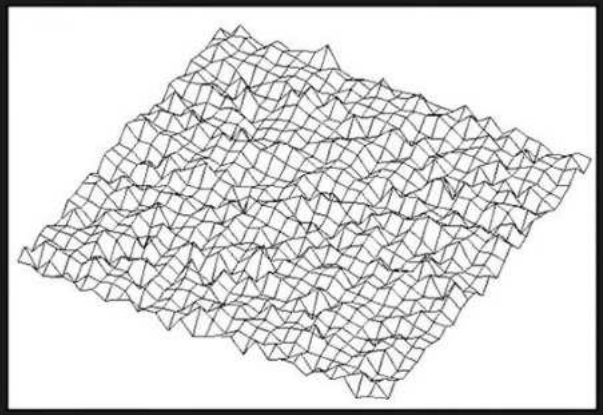
Rough Surfaces



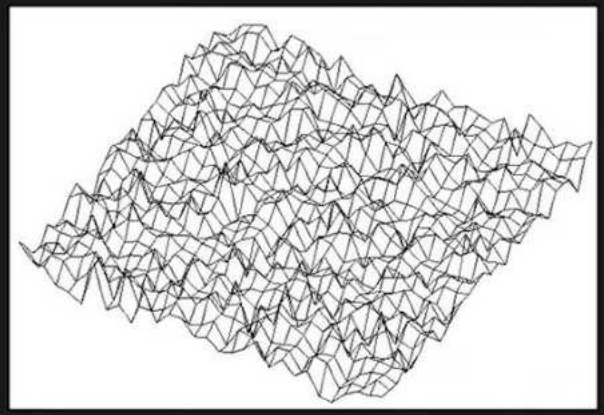
$\sigma = 0$



$\sigma = 0.1$



$\sigma = 0.3$



$\sigma = 0.6$

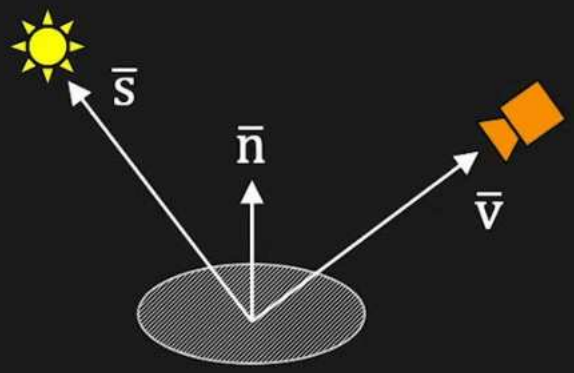
Specular Reflection from Rough Surface

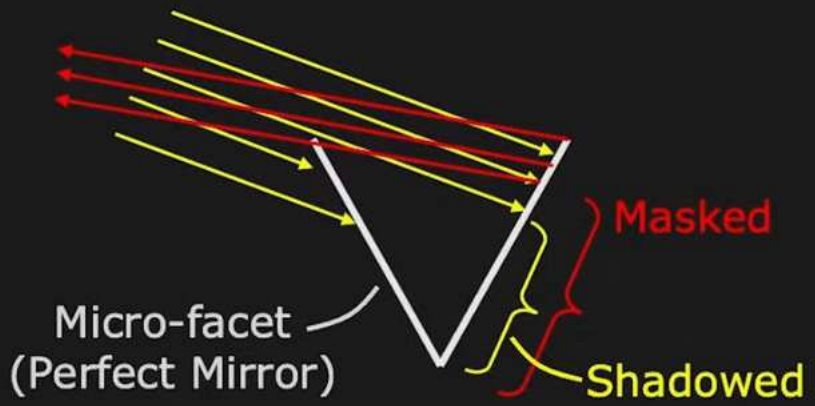
Torrance-Sparrow BRDF Model: (Each Facet is a Perfect Mirror)

$$f(\bar{s}, \bar{v}) = \frac{\rho_s}{(\bar{n} \cdot \bar{s})(\bar{n} \cdot \bar{v})} \, p(\alpha, \sigma) \, G(\bar{s}, \bar{n}, \bar{v})$$

where $p(\alpha, \sigma)$: surface roughness distribution

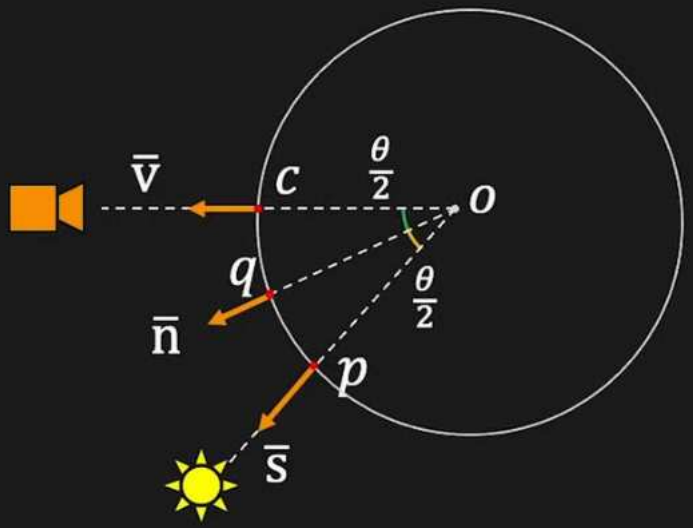
$G(\bar{s}, \bar{n}, \bar{v})$: geometric factor (masking, shadowing)





[Torrance 1967]

Specular Reflection from Rough Surface



$\sigma = 0$

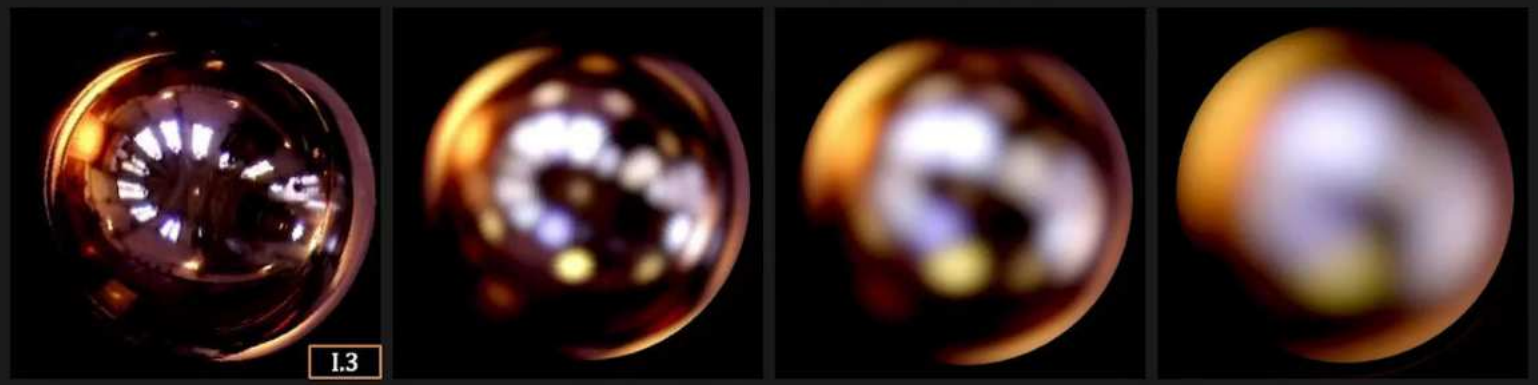
(Mirror)

$\sigma = 0.1$

$\sigma = 0.3$

$\sigma = 0.6$

Specular Reflection from Rough Surface

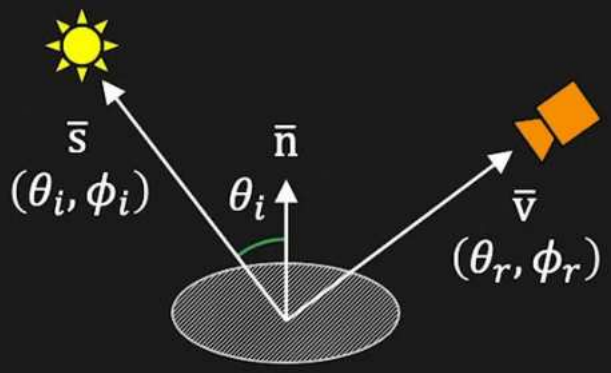


Roughness

Body Reflection from Rough Surfaces

Oren-Nayar BRDF Model: (Each Facet is Lambertian)

$$f(\theta_i, \phi_i, \theta_r, \phi_r) = \frac{\rho_d}{\pi}(A + B \cdot \max(0, \cos(\phi_r - \phi_i)) \cdot \sin\alpha \cdot \tan\beta)$$



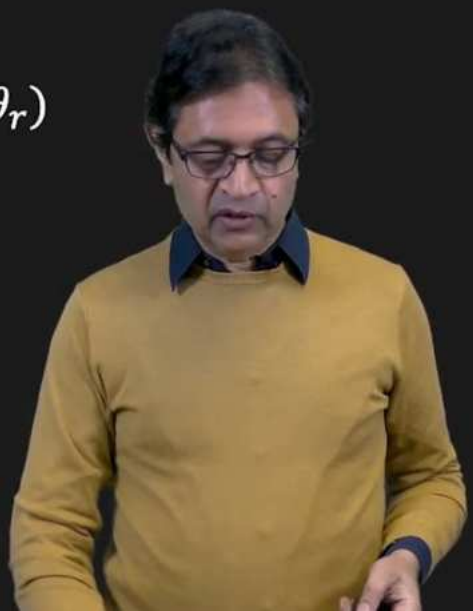
$$A = 1 - \frac{\sigma^2}{2(\sigma^2 + 0.33)} \qquad \alpha = \max(\theta_i, \theta_r)$$

$$B = \frac{0.45\sigma^2}{(\sigma^2 + 0.09)} \qquad \beta = \min(\theta_i, \theta_r)$$

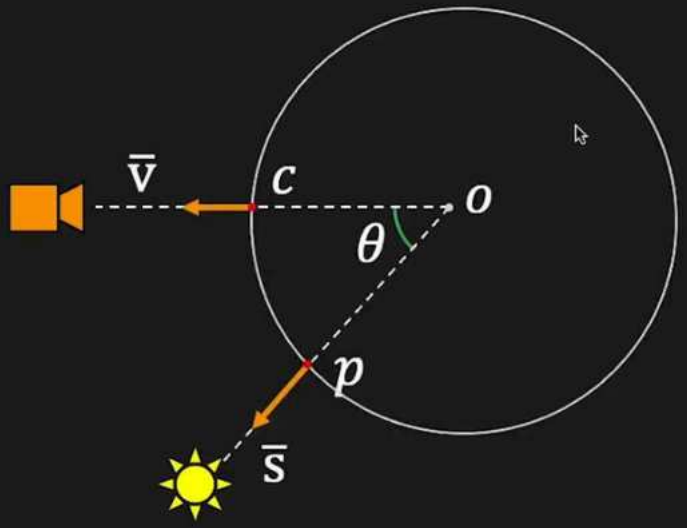
σ - surface roughness

When $\sigma = 0$, it is the Lambertian model

[Oren 1993]



Body Reflection from Rough Surfaces



$\sigma = 0$

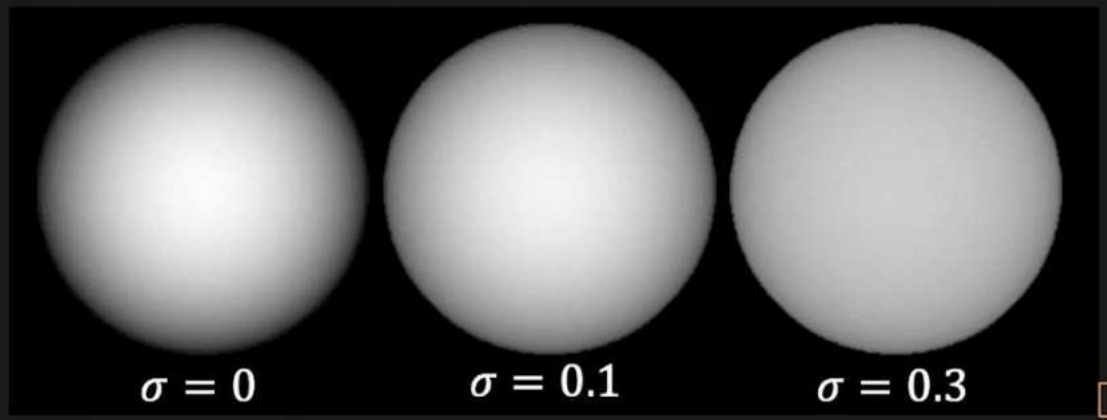
(Lambertian)

$\sigma = 0.1$

$\sigma = 0.3$

$\sigma = 0.6$

Body Reflection from Rough Surfaces



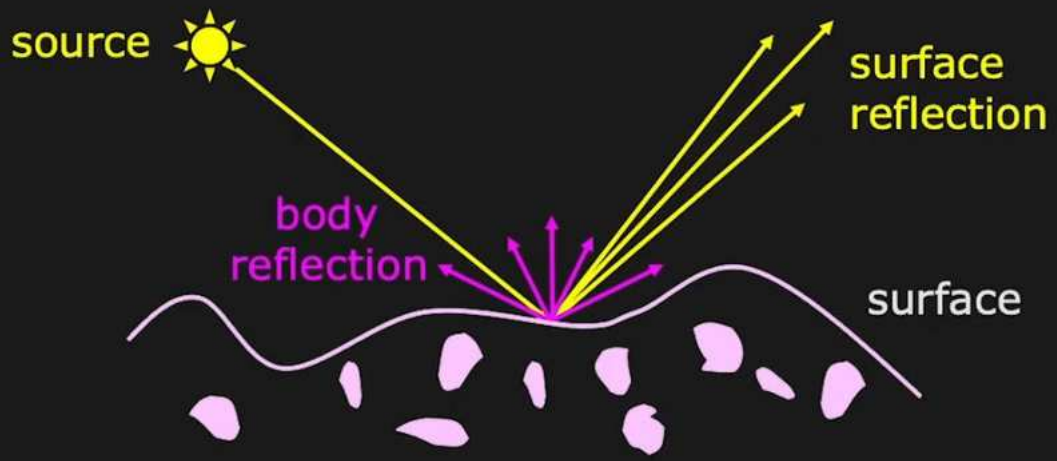
I.6

Full Moon

Roughness

[Oren 1993]

Color Reflectance Model



- Color of body (diffuse) reflection
  = color of object x color of illumination

- Color of surface (specular) reflection
  = color of illumination

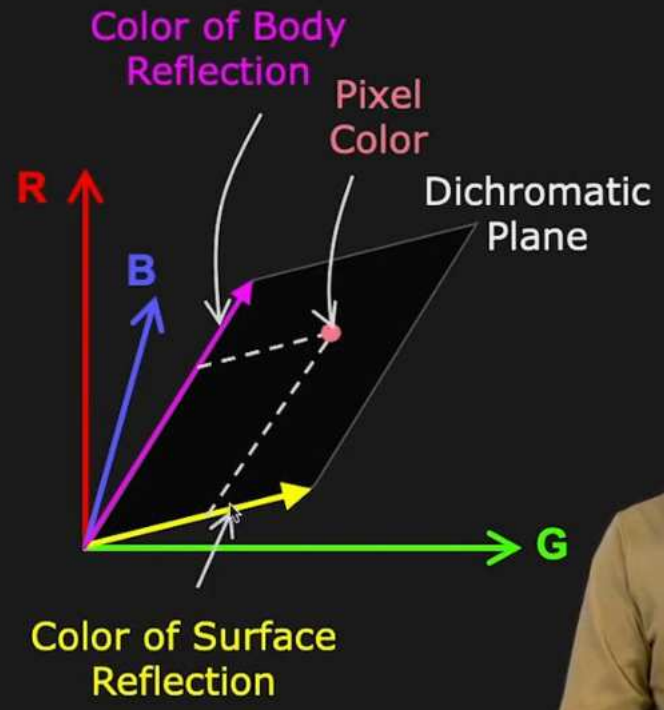
Color Reflectance: Dichromatic Model

Pixel color is a linear combination of the color of body reflection and the color of surface reflection.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = m_b \begin{bmatrix} R_b \\ G_b \\ B_b \end{bmatrix} + m_s \begin{bmatrix} R_s \\ G_s \\ B_s \end{bmatrix}$$

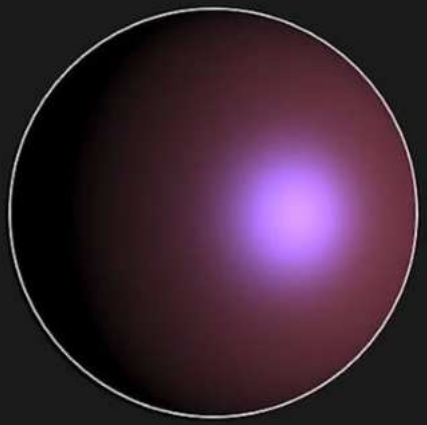
Color of Body Reflection

Color of Surface Reflection

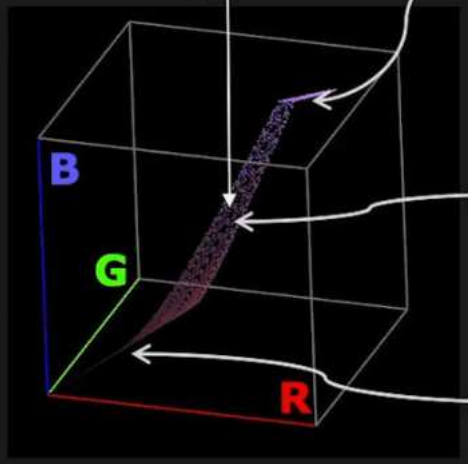


[Shafer 1985]

Color Reflectance: Dichromatic Model



Illumination Color:



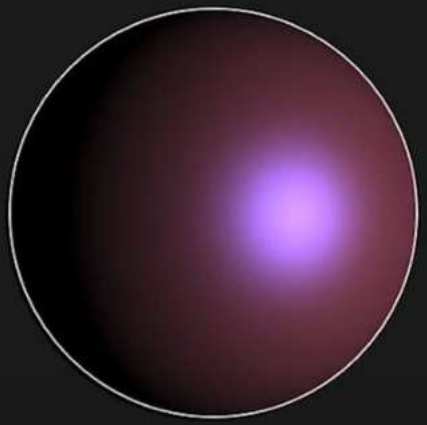
Clipping due to Saturation

Surface Reflection

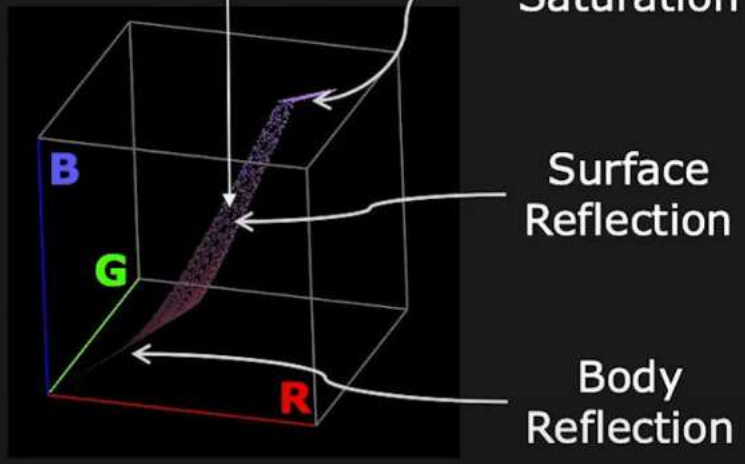
Body Reflection

Color Histogram

Color Reflectance: Dichromatic Model

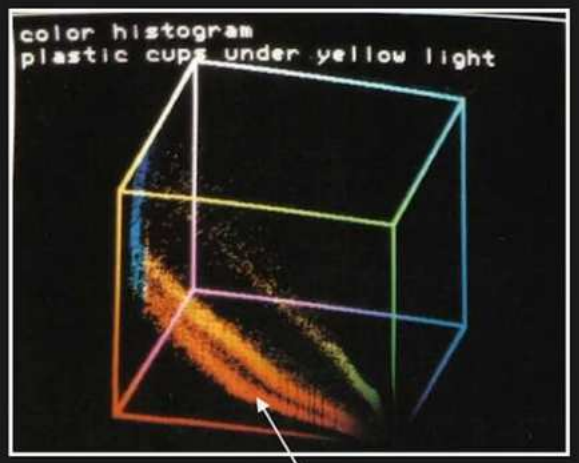


Illumination Color:

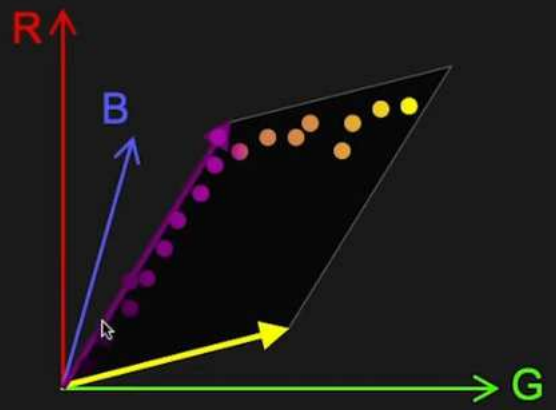


Color Histogram

Separating Body and Surface Reflection



"Skewed-T"



[Klinker 1990]

Separating Body and Surface Reflection

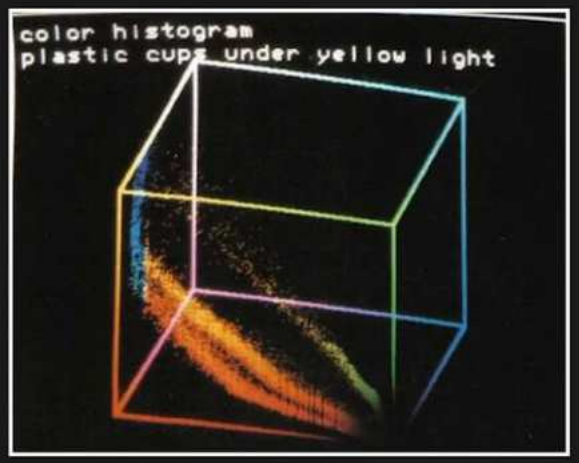


Image Intensity

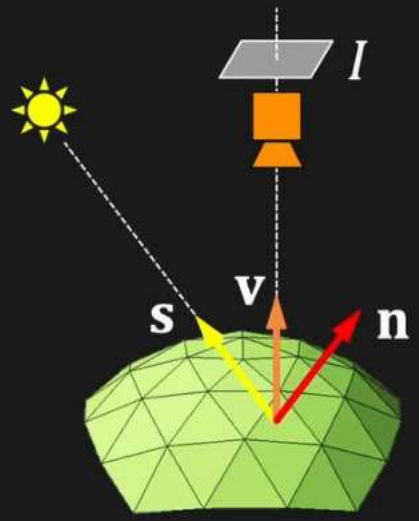


Image Intensity $I = \mathcal{F}$(Source,
Surface Normal $\mathbf{n}$,
Surface Reflectance)

Photometric Stereo

Method for recovering 3D shape information from image intensities measured using multiple sources.

Topics:

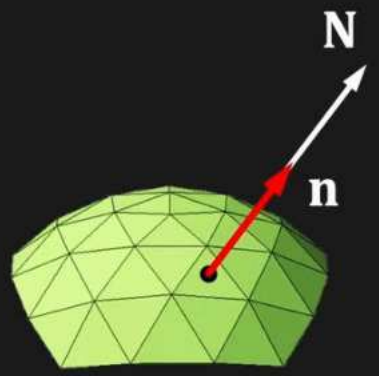
(1) Gradient Space and Reflectance Map

(2) Photometric Stereo

(3) Calibration Based Photometric Stereo

Surface Gradient and Normal

Let $z = f(x, y)$ represent a 3D surface.



$z = f(x, y)$

Surface gradient:

$$\left(-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}\right) = (p, q)$$

Surface normal:

$$\mathbf{N} = \left(-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1\right) = (p, q, 1)$$

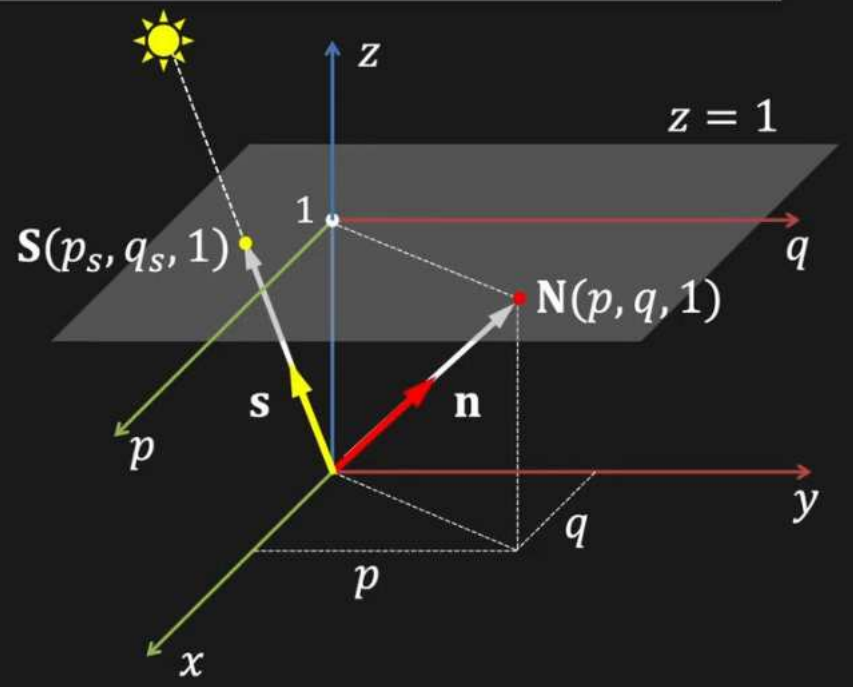
Unit surface normal:

$$\mathbf{n} = \frac{\mathbf{N}}{\|\mathbf{N}\|} = \frac{(p, q, 1)}{\sqrt{p^2 + q^2 + 1}}$$

Surface Normal represented with only two parameters $(p,$

Gradient Space

Plane $z = 1$ is called the Gradient Space or pq Plane



Surface normal:

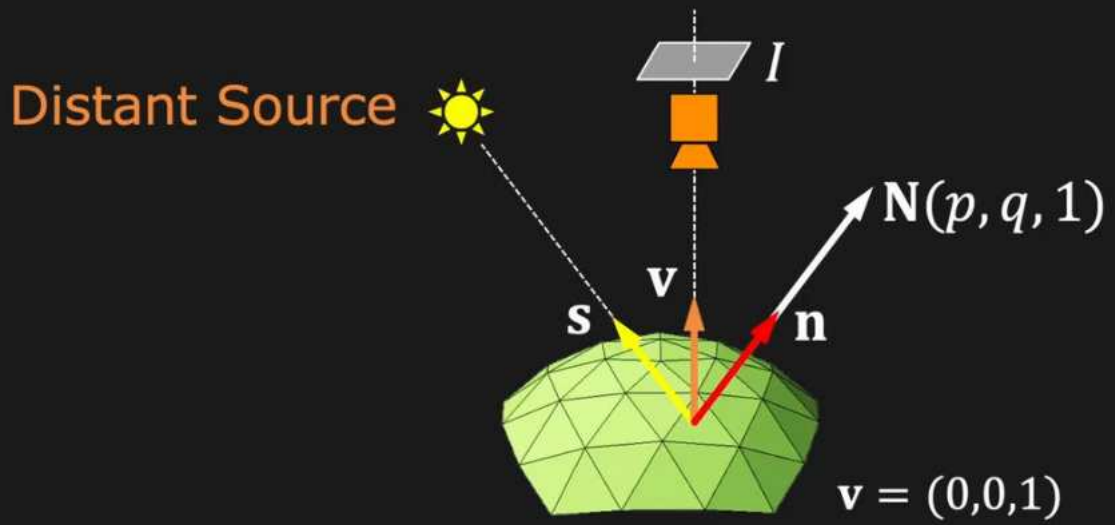
$$\mathbf{n} = \frac{\mathbf{N}}{\|\mathbf{N}\|} = \frac{(p, q, 1)}{\sqrt{p^2 + q^2 + 1}}$$

Source direction:

$$\mathbf{s} = \frac{\mathbf{S}}{\|\mathbf{S}\|} = \frac{(p_s, q_s, 1)}{\sqrt{{p_s}^2 + {q_s}^2 + 1}}$$

Every point (p, q) in the Gradient Space corresponds to a unique orientation.

Reflectance Map $R(p, q)$



For a given source direction **s** and surface reflectance, image intensity at a point (x, y):

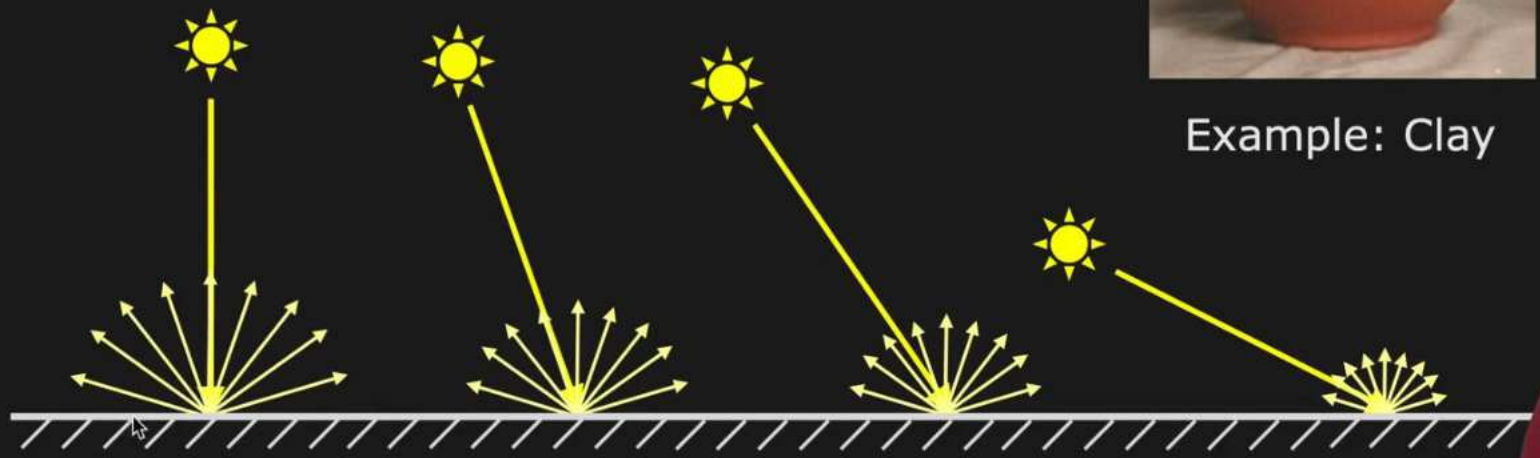
$$I = R(p, q)$$

Reflectance Map

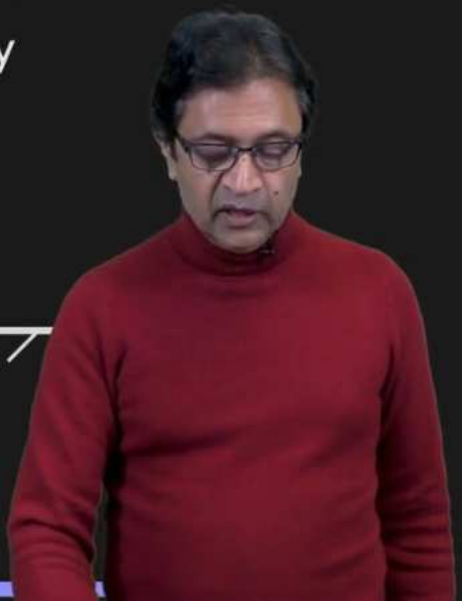
Review: Lambertian Surface

Image intensity I is independent of viewing direction.

Example: Clay



Lambertian (Diffuse or Matte) Surface



Reflectance Map: Lambertian Surface

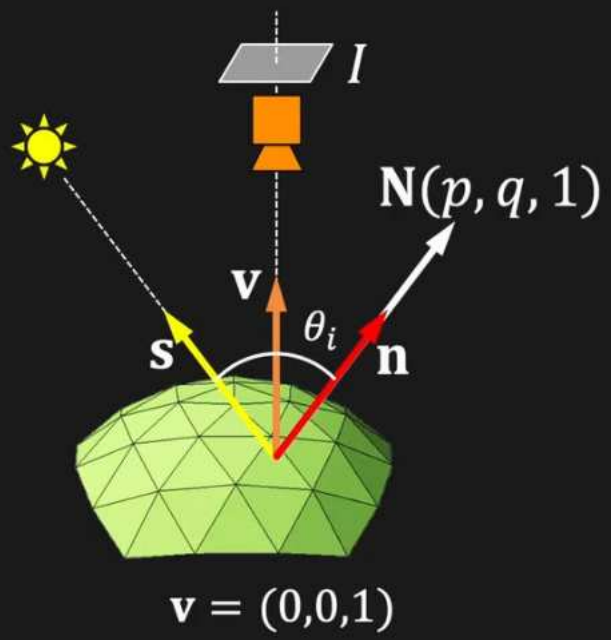
Image Intensity:

$$I = c\frac{\rho}{\pi}\frac{J}{r^2}\cos\theta_i = c\frac{\rho}{\pi}k(\mathbf{n}\cdot\mathbf{s})$$

where k: Source "Brightness"

ρ: Surface Albedo (Reflectance)

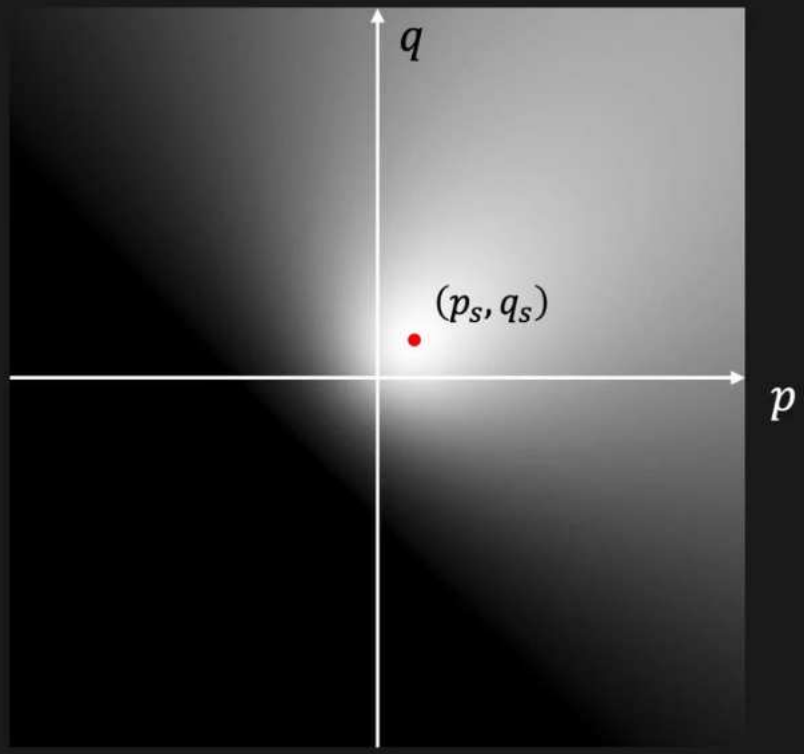
c: Constant (Camera Gain)



Let $c\frac{\rho}{\pi}k = 1$ then, $\boxed{I = \cos\theta_i = \mathbf{n}\cdot\mathbf{s}}$

Reflectance Map: Lambertian Surface

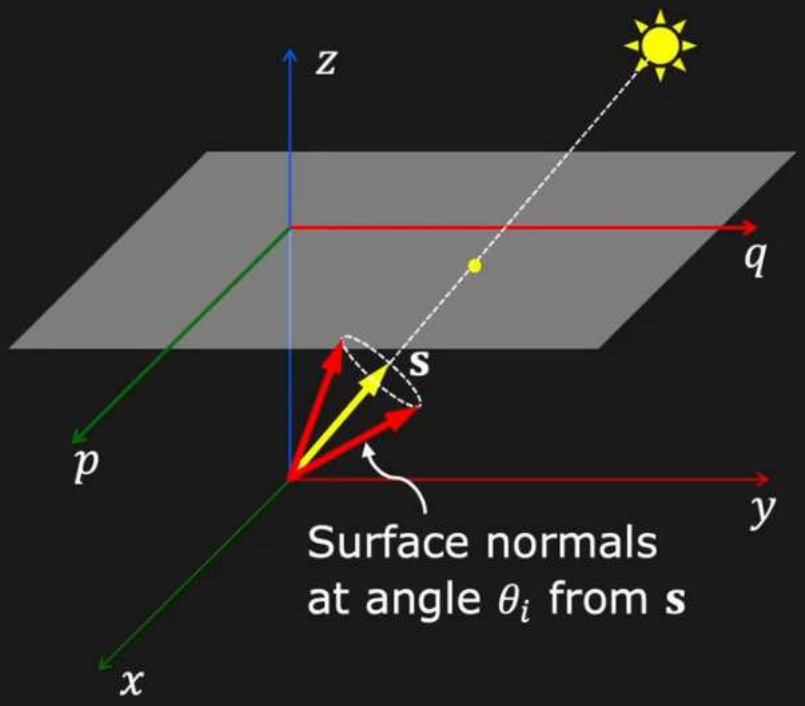
$$I = \mathbf{n} \cdot \mathbf{s} = \frac{pp_s + qq_s + 1}{\sqrt{p^2 + q^2 + 1}\sqrt{{p_s}^2 + {q_s}^2 + 1}} = R(p, q)$$



Reflectance Map

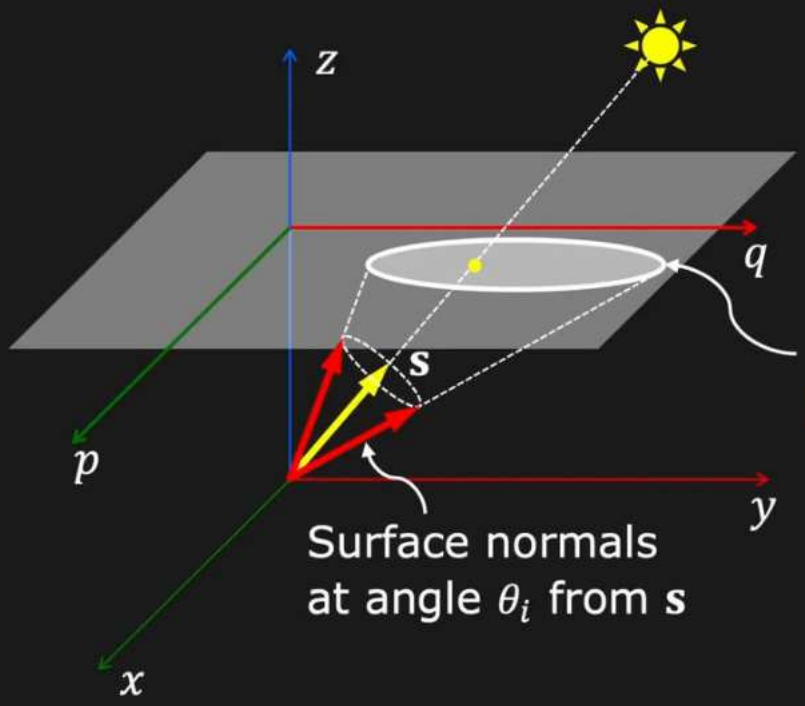
Reflectance Map: Iso-Brightness Contours

$$I = \mathbf{n} \cdot \mathbf{s} = \frac{pp_s + qq_s + 1}{\sqrt{p^2 + q^2 + 1}\sqrt{{p_s}^2 + {q_s}^2 + 1}} = R(p, q)$$



Reflectance Map: Iso-Brightness Contours

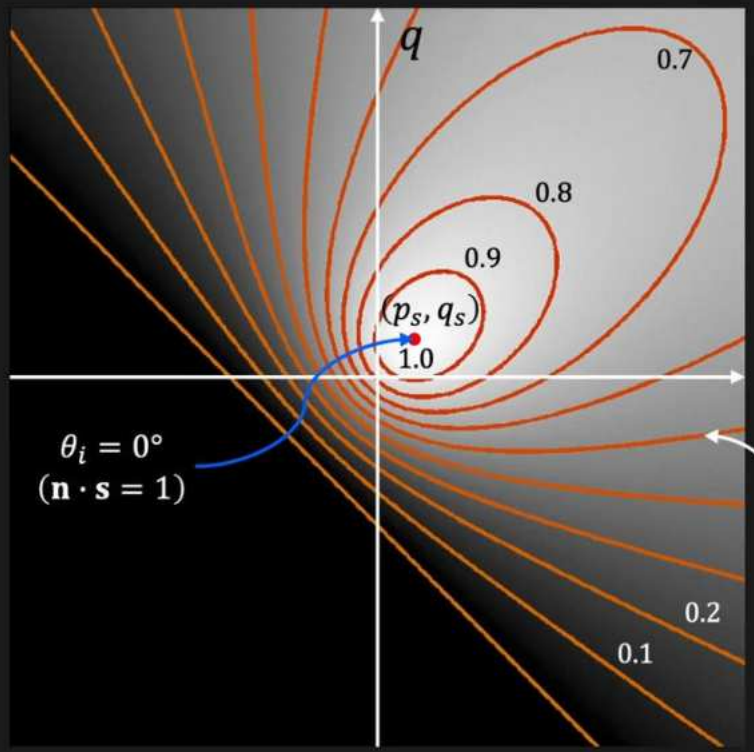
$$I = \mathbf{n} \cdot \mathbf{s} = \frac{pp_s + qq_s + 1}{\sqrt{p^2 + q^2 + 1}\sqrt{{p_s}^2 + {q_s}^2 + 1}} = R(p, q)$$



Iso-Brightness Contour
($\mathbf{n} \cdot \mathbf{s}$ is constant) is a
Conic Section

Reflectance Map: Lambertian Surface

$$I = \mathbf{n} \cdot \mathbf{s} = \frac{pp_s + qq_s + 1}{\sqrt{p^2 + q^2 + 1}\sqrt{{p_s}^2 + {q_s}^2 + 1}} = R(p, q)$$



Iso-Brightness Contour

Reflectance Map

Shape from a Single Image?

Given image I, source direction $\mathbf{s}$ and surface reflectance

Reflectance Map $R(p, q)$

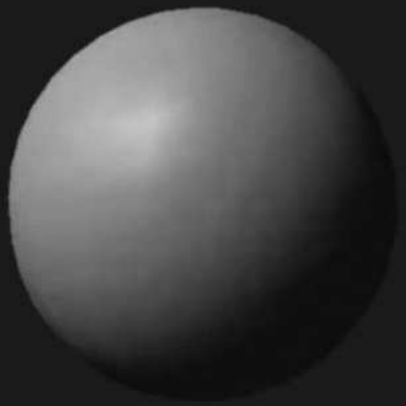
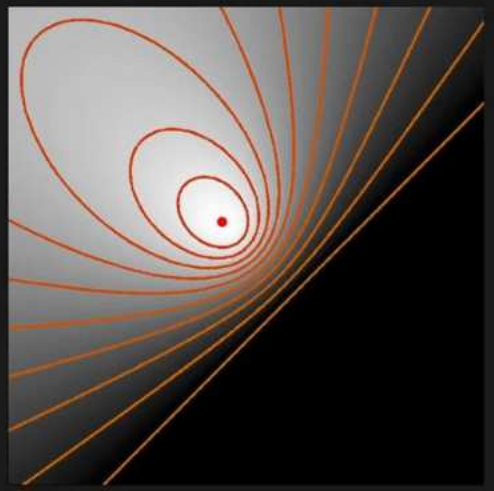
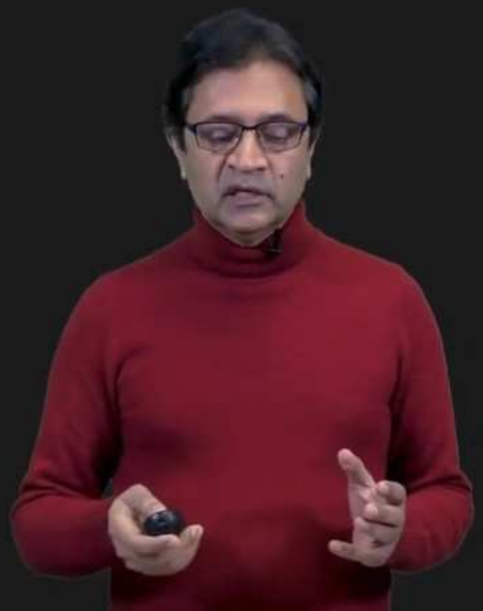


Image I



Reflectance Map $R(p, q)$



Shape from a Single Image?

Given image I, source direction s and surface reflectance

Reflectance Map $R(p, q)$

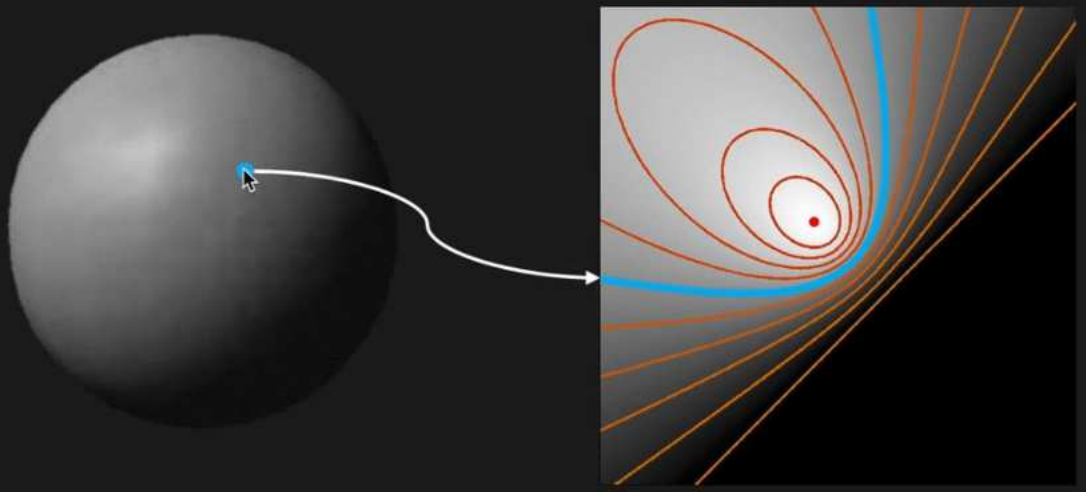


Image I Reflectance Map $R(p, q)$

Can we estimate surface gradients (p, q) at each pixel? NO

Intensity at each pixel maps to infinite (p, q) values along the corresponding iso-brightness contour.

Photometric Stereo

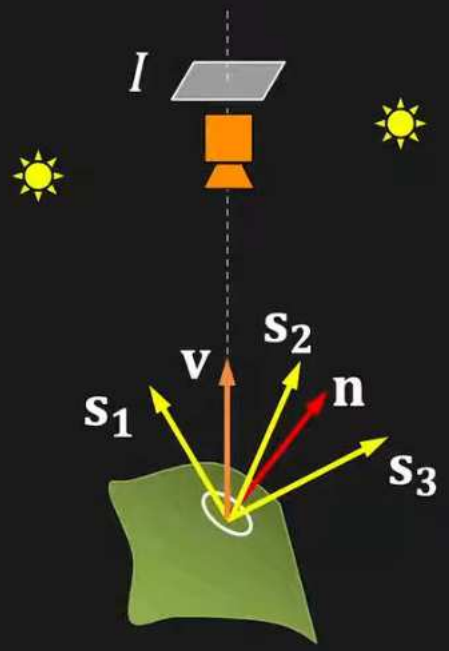
Idea: Use multiple images under different lighting to resolve the ambiguity in surface orientation.

Notation:

Direction of source i: $\mathbf{s_i} \equiv (p_{s_i}, q_{s_i})$

Reflectance map for source i: $R_i(p, q)$

Image intensity produced by source i: $I_i(x, y)$



[Woodham 1980]

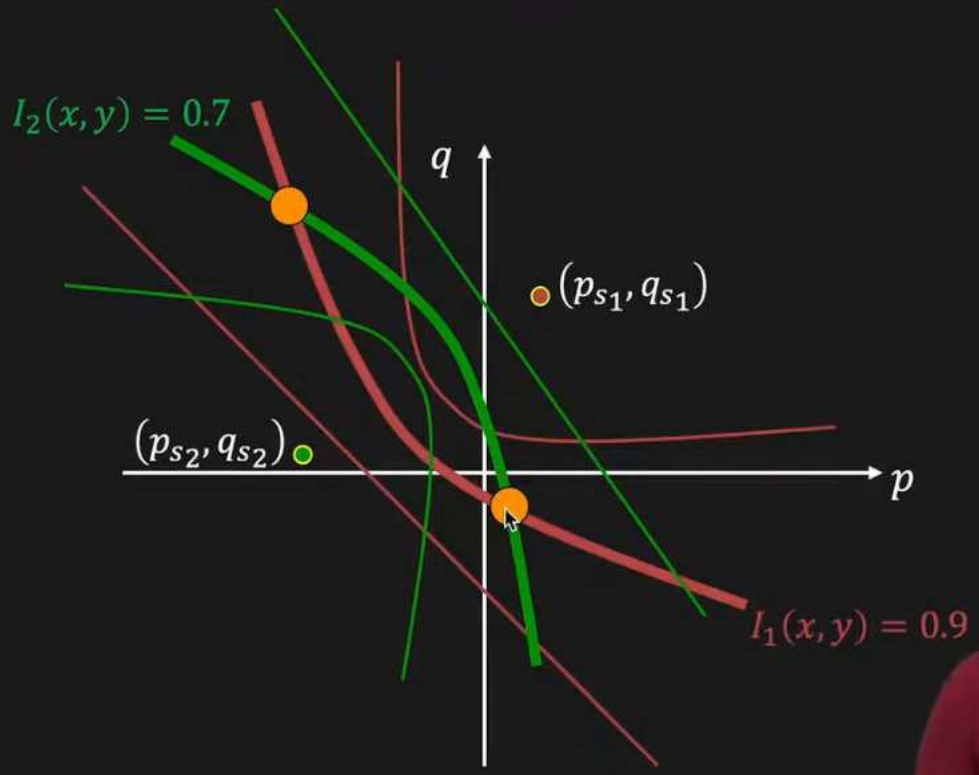
Photometric Stereo: Basic Idea

Capture Image I_2 under Light Source $\mathbf{s}_2(p_{s_2}, q_{s_2})$.

For Example:

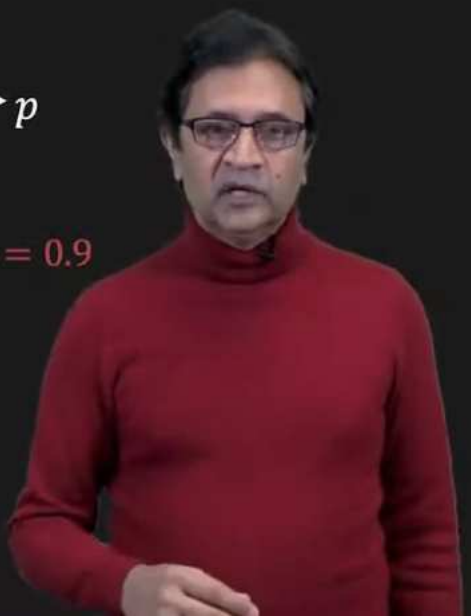
Let $I_1(x, y) = 0.9$

$I_2(x, y) = 0.7$



Two solutions exist for (p, q)

Reflectance Maps $R_1(p, q)$, $R_2(p, q)$



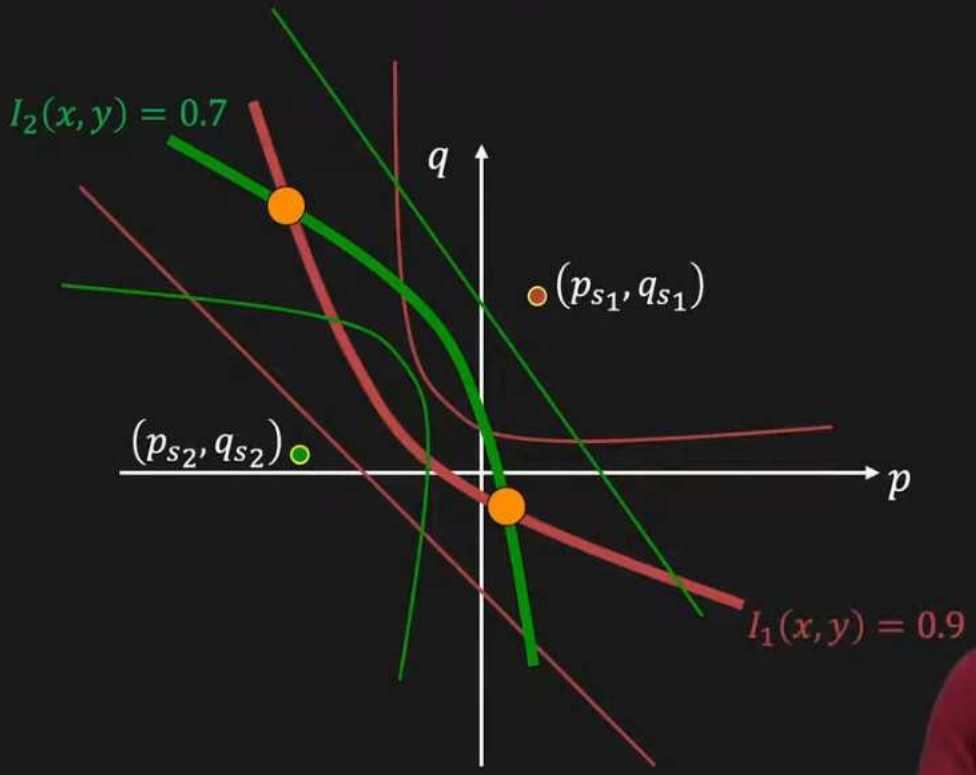
Photometric Stereo: Basic Idea

Capture Image I_2 under Light Source $\mathbf{s}_2(p_{s_2}, q_{s_2})$.

For Example:

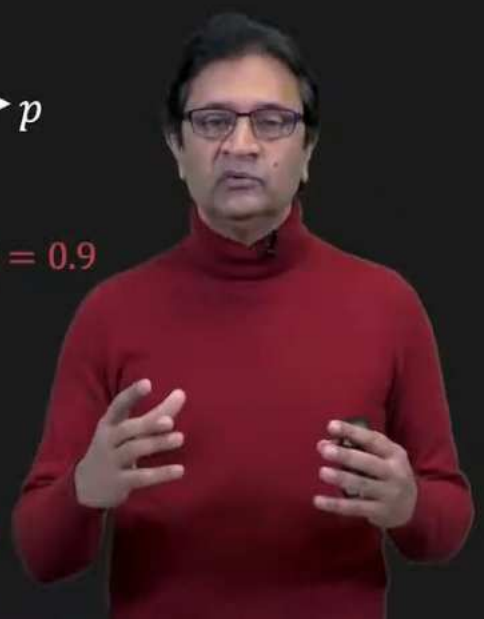
Let $I_1(x, y) = 0.9$

$I_2(x, y) = 0.7$



Two solutions exist for (p, q)

Reflectance Maps $R_1(p, q)$, $R_2(p, q)$



Photometric Stereo: Basic Idea

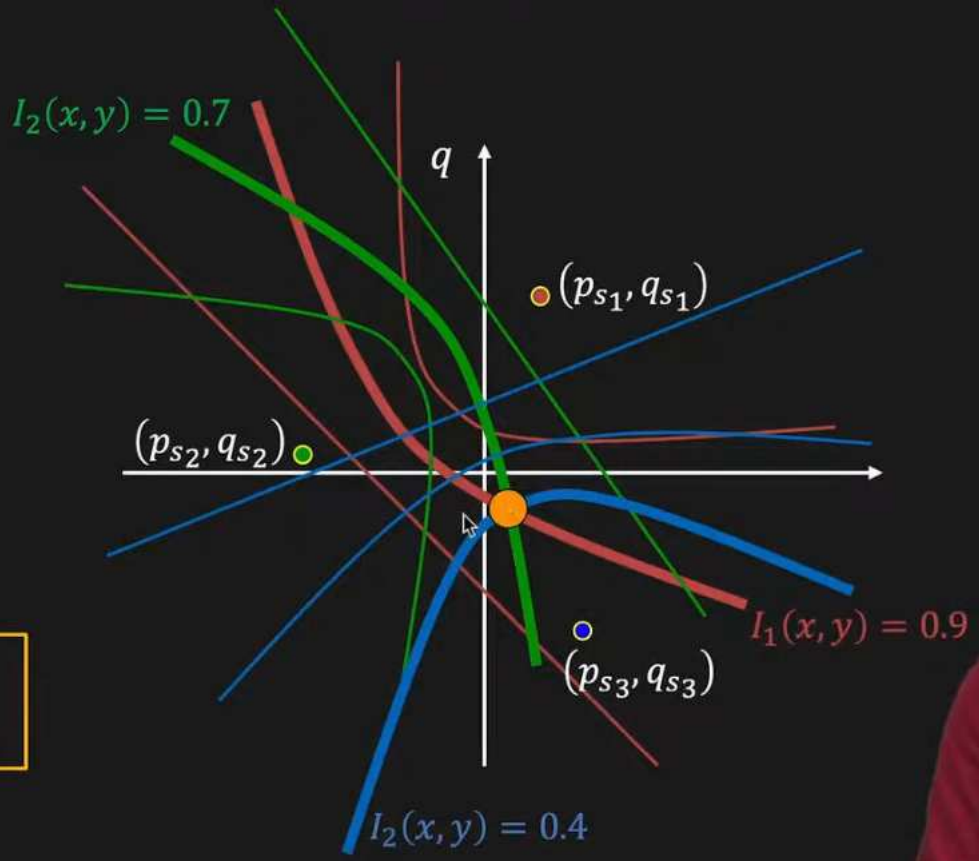
Capture Image I_3 under light source $\mathbf{s}_3(p_{s_3}, q_{s_3})$.

For Example:

Let $I_1(x, y) = 0.9$

$I_2(x, y) = 0.7$

$I_3(x, y) = 0.4$



Unique solution for surface orientation: (p', q')

Reflectance Maps $R_1(p, q)$, $R_2(p, q)$, $R_3(p,$

Photometric Stereo: Basic Idea

Step 1: Acquire K images with K known light sources.

Step 2: Using known source direction and BRDF, construct reflectance map for each source direction.

Step 3: For each pixel location (x, y), find (p, q) as the intersection of K curves. This (p, q) gives the surface normal at pixel (x, y).

Smallest K needed depends on the material properties.

Photometric Stereo: Lambertian Case

Image intensities measured at point (x, y) under each of the three light sources:

$$I_1 = \frac{\rho}{\pi}\mathbf{n} \cdot \mathbf{s_1} \qquad I_2 = \frac{\rho}{\pi}\mathbf{n} \cdot \mathbf{s_2} \qquad I_3 = \frac{\rho}{\pi}\mathbf{n} \cdot \mathbf{s_3}$$

where: $\mathbf{n} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}$ and $\mathbf{s_i} = \begin{bmatrix} s_{x_i} \\ s_{y_i} \\ s_{z_i} \end{bmatrix}$

We can write this in matrix form:

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \frac{\rho}{\pi} \begin{bmatrix} s_{x_1} & s_{y_1} & s_{z_1} \\ s_{x_2} & s_{y_2} & s_{z_2} \\ s_{x_3} & s_{y_3} & s_{z_3} \end{bmatrix} \mathbf{n}$$

Measured $\qquad$ $S_{3\times3}$ (Known)

Photometric Stereo: Lambertian Case

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \frac{\rho}{\pi} \underbrace{\begin{bmatrix} s_{x_1} & s_{y_1} & s_{z_1} \\ s_{x_2} & s_{y_2} & s_{z_2} \\ s_{x_3} & s_{y_3} & s_{z_3} \end{bmatrix}}_{S_{3\times 3} \text{ (Known)}} \mathbf{n} \quad \Rightarrow \quad \begin{cases} I = S\mathbf{N} \\ \text{where: } \mathbf{N} = \frac{\rho}{\pi}\mathbf{n} \end{cases}$$

Solution: $\mathbf{N} = (S)^{-1} I$

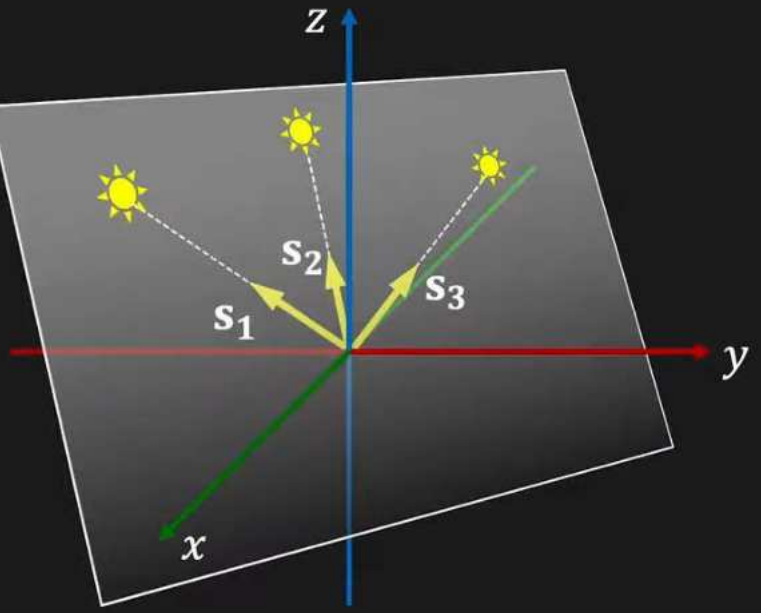
Surface Normal: $\mathbf{n} = \frac{\mathbf{N}}{\|\mathbf{N}\|}$ Albedo: $\frac{\rho}{\pi} = \|\mathbf{N}\|$

When Does It Not Work?

When $S_{3\times3}$ is not invertible.

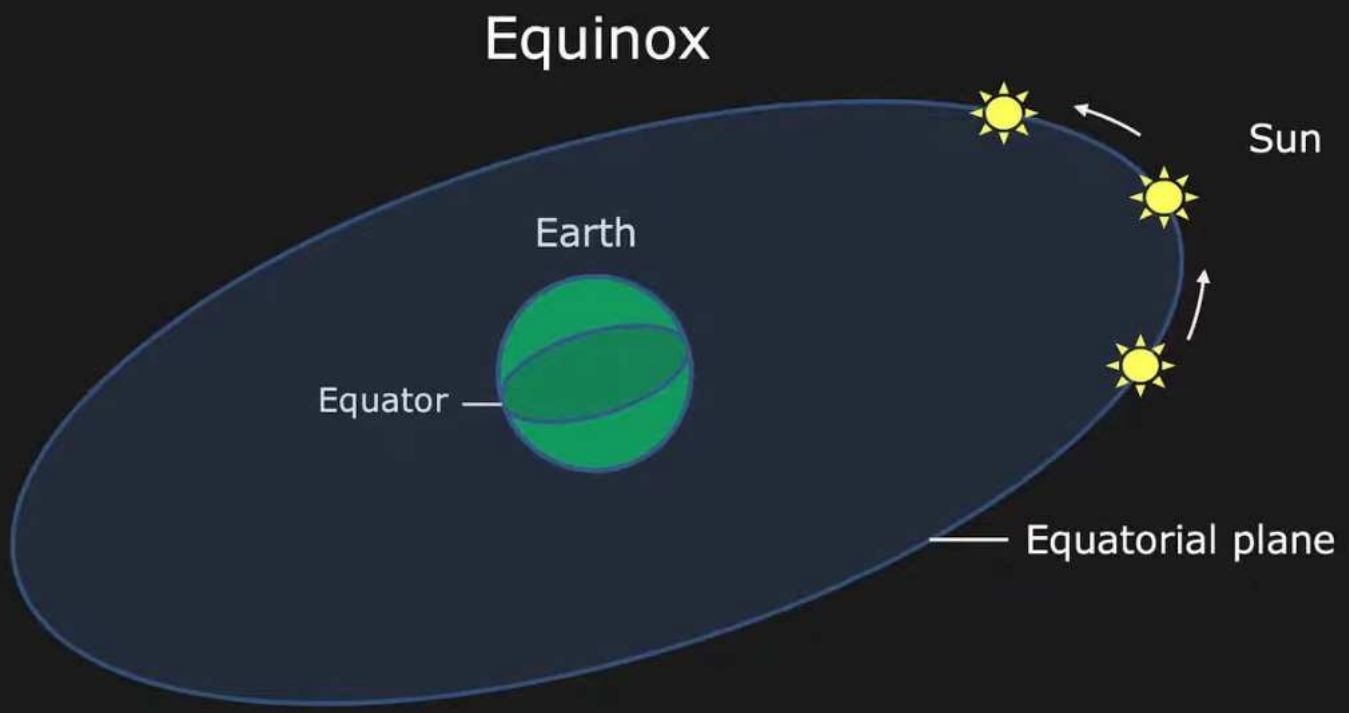
That is, when one source direction can be represented as a linear combination of the other two.

$$\mathbf{s_3} = \alpha\mathbf{s_1} + \beta\mathbf{s_2}$$



All sources and the origin lie on a plane

Bad Days for Outdoor Photometric Stereo



More Sources than Minimum Needed

Better results by using more ($K > 3$) light sources

$$\underbrace{\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_K \end{bmatrix}}_{I_{K\times 1}} = \frac{\rho}{\pi} \underbrace{\begin{bmatrix} s_{x_1} & s_{y_1} & s_{z_1} \\ s_{x_2} & s_{y_2} & s_{z_2} \\ \vdots & \vdots & \vdots \\ s_{x_K} & s_{y_K} & s_{z_K} \end{bmatrix}}_{S_{K\times 3}} \mathbf{n} \Rightarrow \begin{cases} I = S\mathbf{N} \\ \text{where: } \mathbf{N} = \frac{\rho}{\pi}\mathbf{n} \end{cases}$$

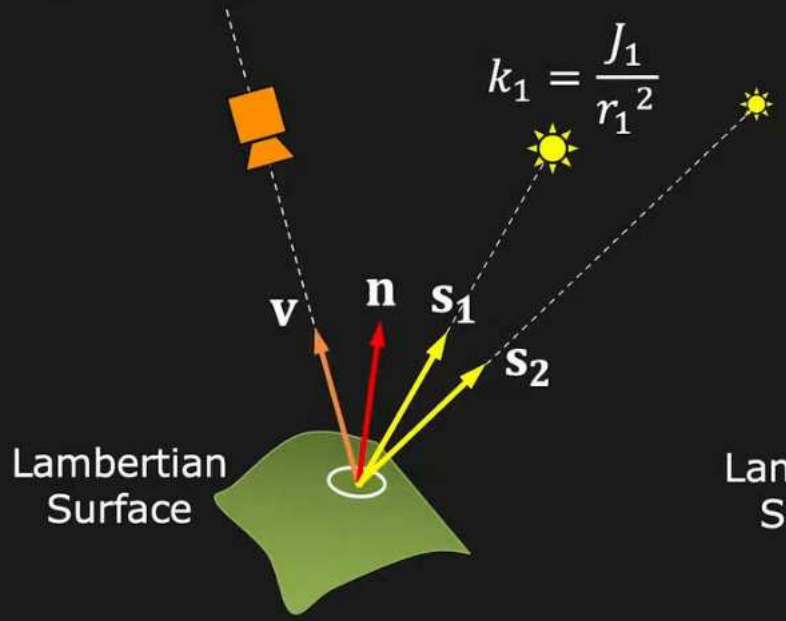
$S_{K\times 3}$ is not a square matrix and hence not invertible.

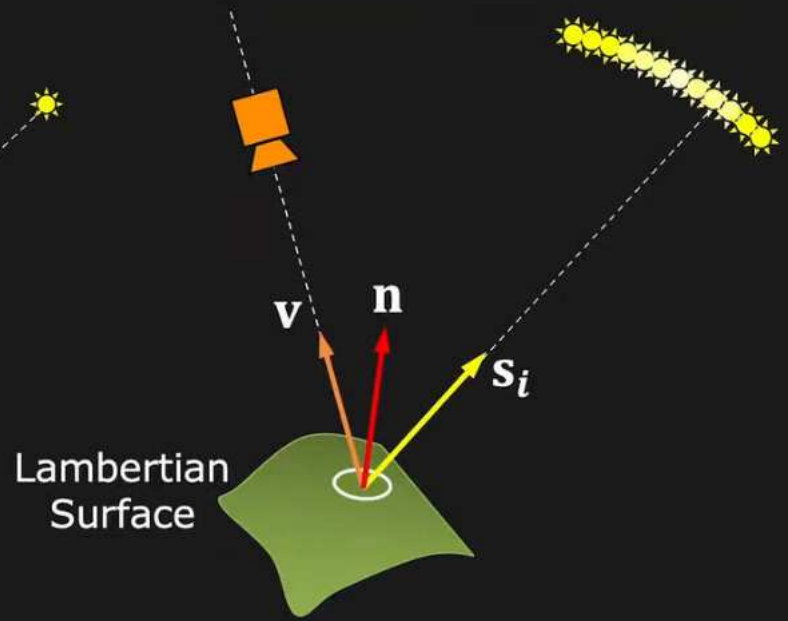
Solution: Use Least Squares Estimation

$$S^T I = \underbrace{S^T S}_{3\times 3}\mathbf{N}$$

$$\mathbf{N} = \underbrace{(S^T S)^{-1} S^T}_{\text{Pseudo Inverse}} I$$

"Effective" Point Source for Multiple Sources





$$I_1 = \frac{\rho}{\pi} k_1 \, \mathbf{n} \cdot \mathbf{s_1} \quad I_2 = \frac{\rho}{\pi} k_2 \, \mathbf{n} \cdot \mathbf{s_2}$$

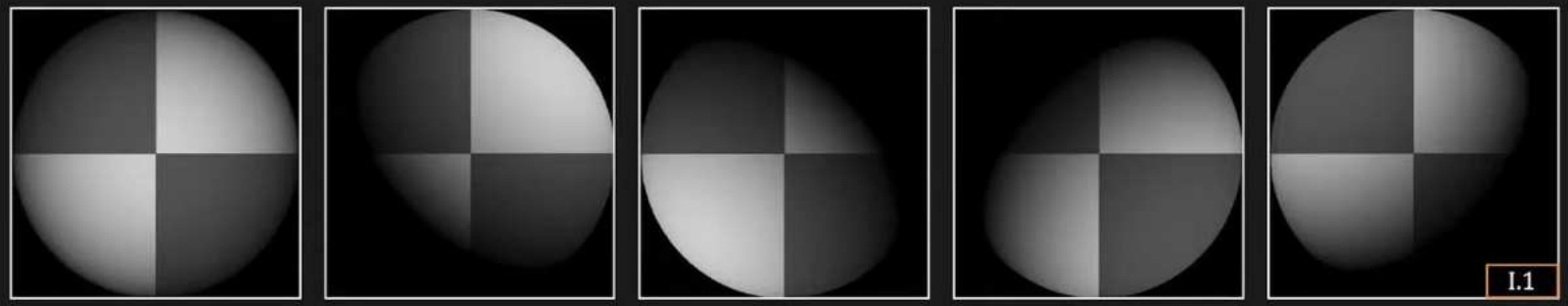
$$I = \frac{\rho}{\pi} \mathbf{n} \cdot (k_1 \mathbf{s_1} + k_2 \mathbf{s_2}) = \frac{\rho}{\pi} \hat{k} \, \mathbf{n} \cdot \hat{\mathbf{s}}$$

$$I = \frac{\rho}{\pi} \mathbf{n} \cdot \sum_i k_i \mathbf{s_i}$$

$$I = \frac{\rho}{\pi} \hat{k} \, \mathbf{n} \cdot \hat{\mathbf{s}}$$

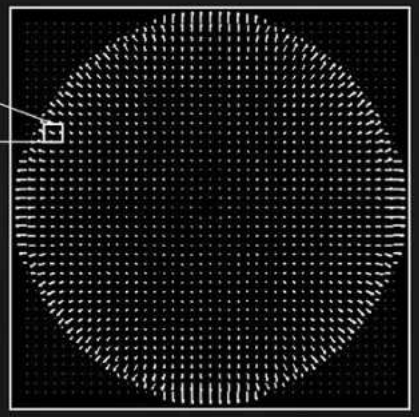
$(\hat{k}, \hat{\mathbf{s}})$: Effective Source is in direction of "Centroid" of Sources

Results: Lambertian Sphere

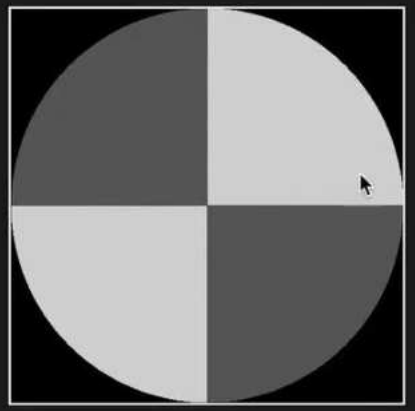


Input images

Needles are projections of surface normals on image plane



Estimated surface normals

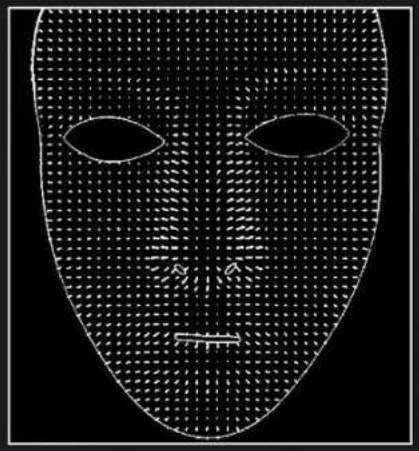


Estimated albedo

Results: Lambertian Mask



Input images



Estimated surface normals

Estimated albedo

Results: Toy

Input images

Estimated surface normals

Estimated albedo

Calibration Based Photometric Stereo

Use a calibration object of known size, shape (e.g. sphere) and same reflectance as the scene objects.

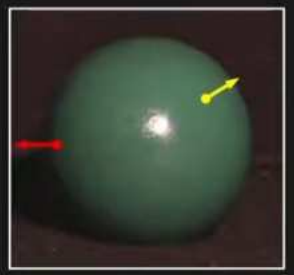
Calibration Sphere

Scene

[Silver 1980]

Calibration Based Photometric Stereo

Use a calibration object (ex: sphere) of known size, shape and same reflectance as the scene objects.



Calibration Sphere

Scene

Orientation Consistency: Points with the same surface norma
produce the same set of intensities under different lighting.

[Silver 1980]

Calibration Procedure

Image 1 Image 2 Image K

Step 1: Capture $K \geq 3$ images under K different light sources.

Each point on the sphere produces K image intensities $(I_1, I_2, \ldots, I_K)$ corresponding to the K light sources.

Calibration Procedure

Image 1

Image 2

...

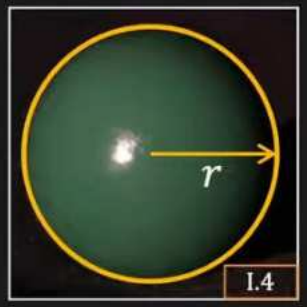
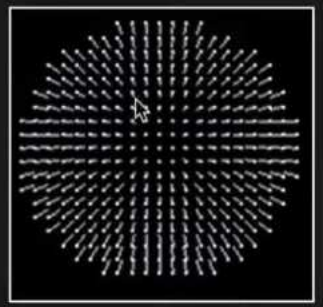


Image K



Surface Normals $(p, q, 1)$

Step 1: Capture $K \geq 3$ images under K different light sources.

Each point on the sphere produces K image intensities $(I_1, I_2, \ldots, I_K)$ corresponding to the K light sources.

Step 2: Using the known size of the sphere, calculate the surface normal $(p, q, 1)$ for each point on the sphere.

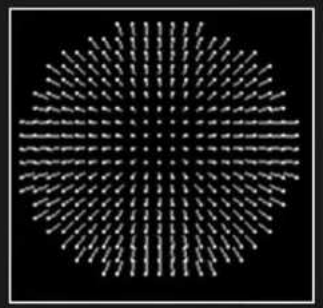
Calibration Procedure

Image 1

Image 2

...

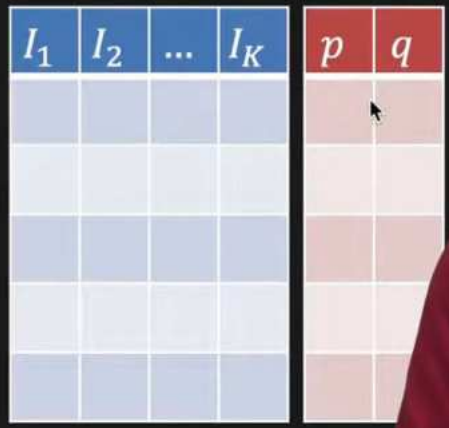
Image K

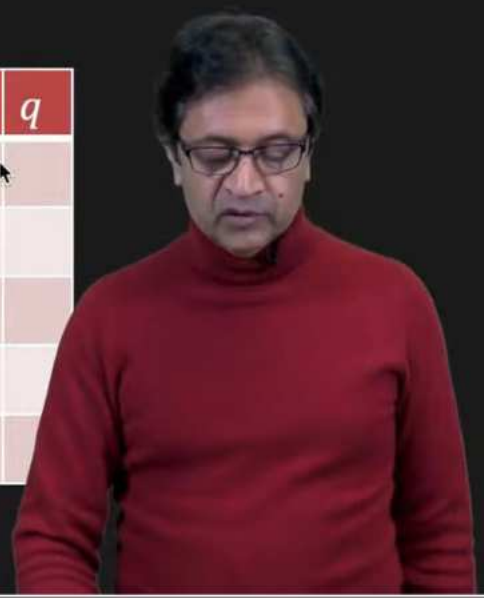


Surface Normals $(p, q, 1)$

Step 3: Create a lookup table for the K-tuple: $(I_1, I_2, \ldots, I_K) \rightarrow (p, q)$

Populate the lookup table with $(I_1, I_2, \ldots, I_K)$ and (p, q) of all pixels on the sphere.





Looking Up Surface Normal

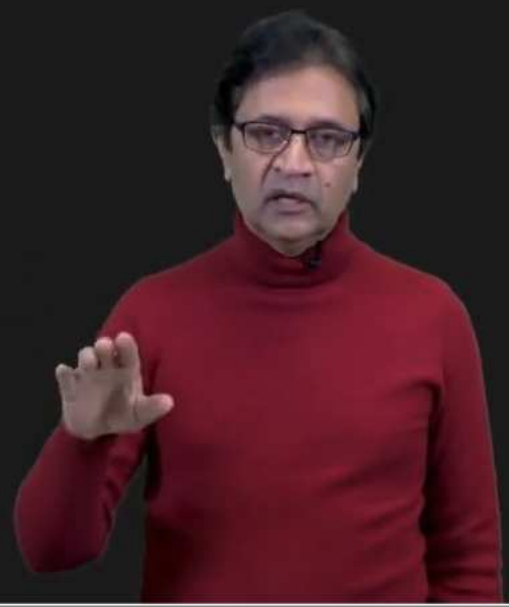
Step 4: Capture K images of the scene object under the same K light sources.

Image 1

Image 2

...

Image K



Looking Up Surface Normal

Step 4: Capture K images of the scene object under the same K light sources.

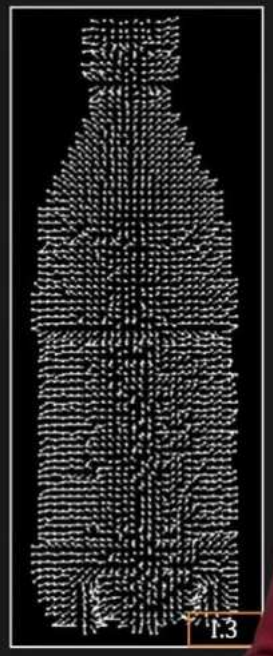
Step 5: For each pixel in the scene, use lookup table to map $(I_1, I_2, \ldots, I_K) \rightarrow (p, q)$

Image 1

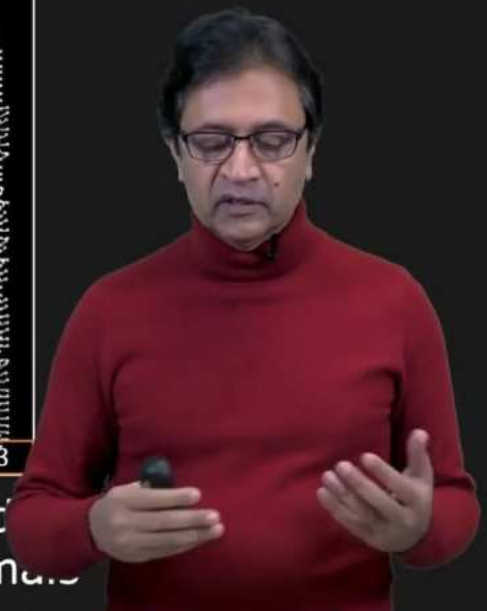
Image 2

...

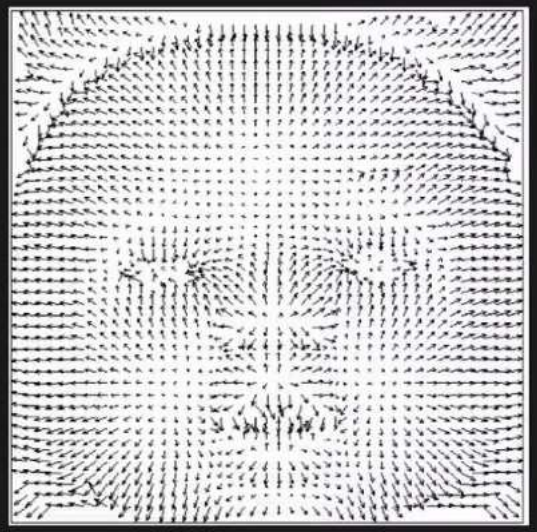
Image K



Estimated surface normals

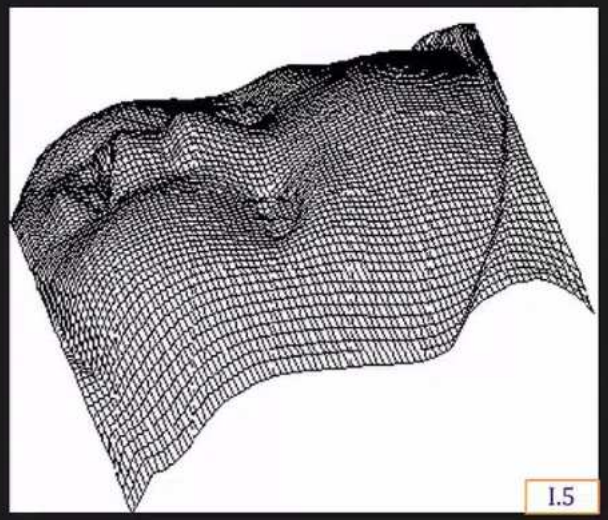


Shape From Surface Normals



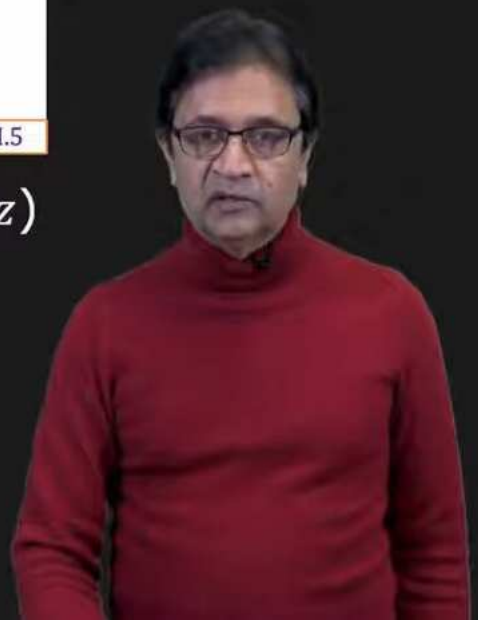
Gradient/normal map

$[p(x,y), q(x,y), 1]$

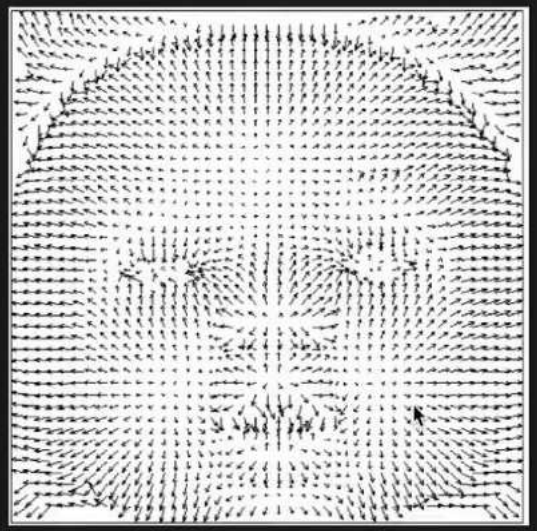


Shape or depth map (z)

$$(p,q) = \left(-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}\right)$$

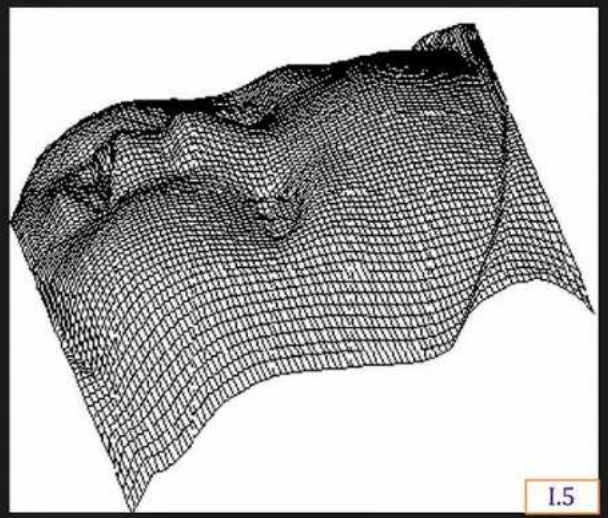


Shape From Surface Normals



Differentiation

Integration

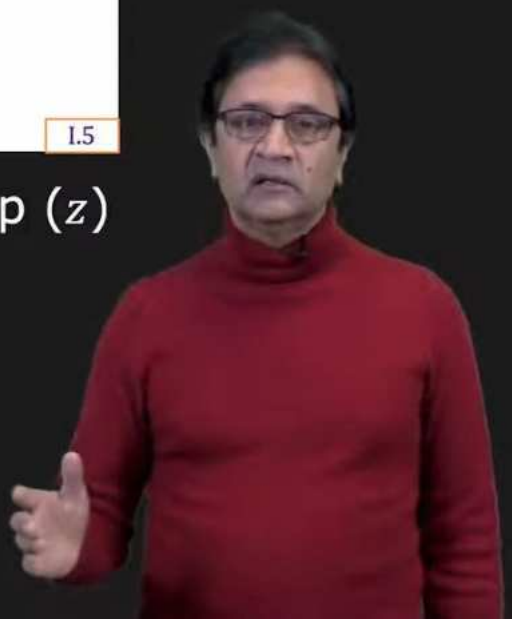


Gradient/normal map

$[p(x,y), q(x,y), 1]$

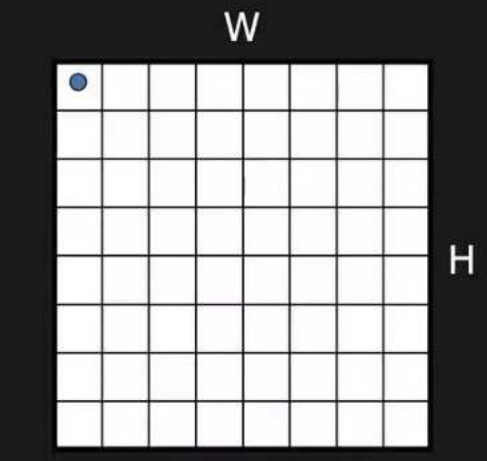
Shape or depth map (z)

$$(p,q) = \left(-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}\right)$$



Naïve Algorithm for Estimating Shape

1. Initialize reference depth
   $z(0,0) = 0$



Computed Depth Map

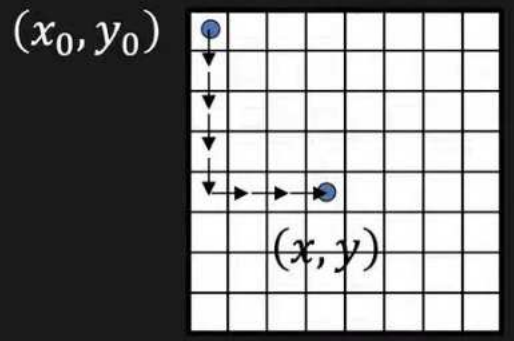
Shape From Surface Normals

Estimate surface by integrating surface gradient:

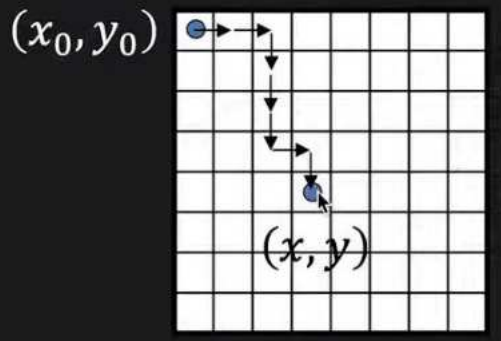
$$z(x,y) = z(x_0,y_0) + \int_{(x_0,y_0)}^{(x,y)} -(pdx + qdy)$$

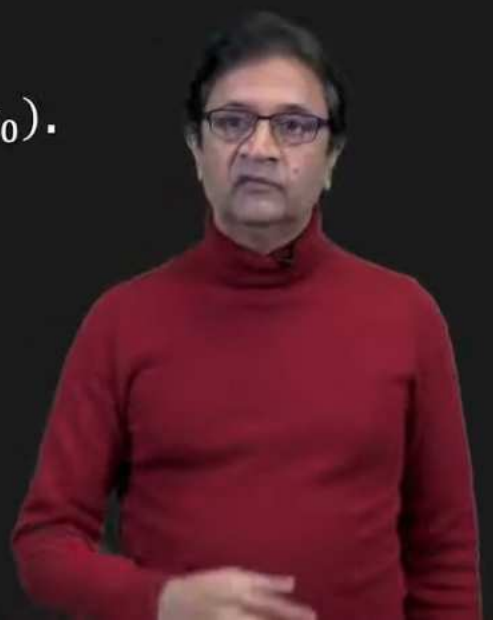
where (x_0, y_0) is a reference point and $z(x_0, y_0) = 0$.

$z(x,y)$ obtained by integration along any path from (x_0, y_0).



or



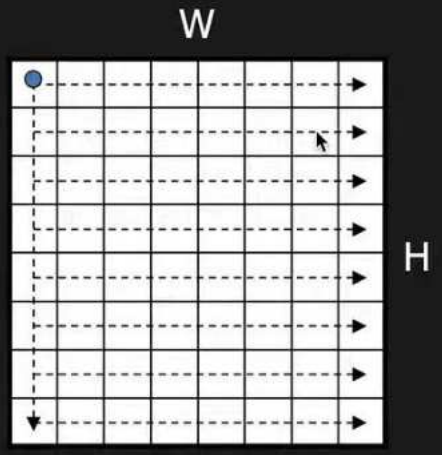


Naïve Algorithm for Estimating Shape

1. Initialize reference depth
   $z(0,0) = 0$

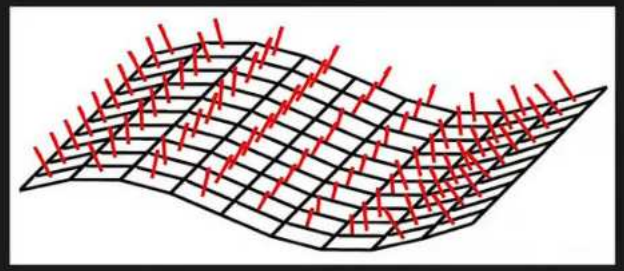
2. Compute depth for first column
   for $y = 1$ to $(H-1)$
   $z(0,y) = z(0,y-1) - q(0,y)$

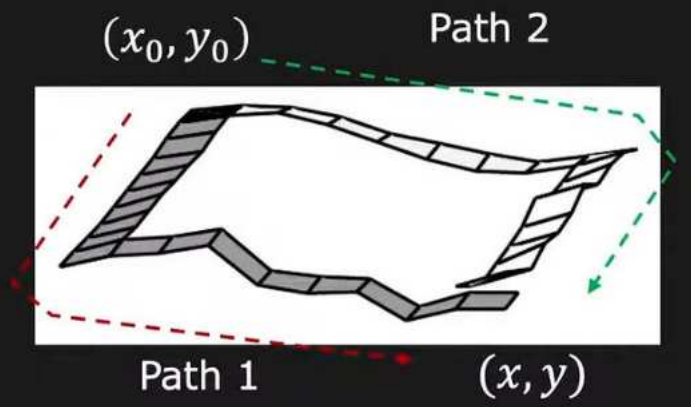
3. Compute depth for each row
   for $y = 0$ to $(H-1)$
   for $x = 1$ to $(W-1)$
   $z(x,y) = z(x-1,y) - p(x,y)$



Computed Depth Map

Noise Sensitivity of Computed Shape





Actual surface shown with noisy estimates of surface gradient

Depth computed from noisy gradients depends on the integration path

One Solution: Compute depth maps using different paths. Then, find average of computed depth maps to reduce error.

Estimating Shape Using Least Squares

Minimize the errors between measured surface gradients (p, q) and surface gradients of estimated surface $z(x, y)$.

Error Measure:

$$D = \iint_{Image} \left(\frac{\partial z}{\partial x} + p\right)^2 + \left(\frac{\partial z}{\partial y} + q\right)^2 dxdy$$

where $\frac{\partial z}{\partial x}$ and $\frac{\partial z}{\partial y}$ are gradients of the estimated surface.

[Horn 1990]

Frankot-Chellappa Algorithm

Minimize objective function D in Fourier Domain.

Let $Z(u,v)$, $P(u,v)$ and $Q(u,v)$ be the Fourier Transforms of $z(x,y)$, $p(x,y)$ and $q(x,y)$, respectively. Then:

$$z(x,y) = \iint_{-\infty}^{\infty} Z(u,v) e^{i2\pi(ux+vy)} du dv$$

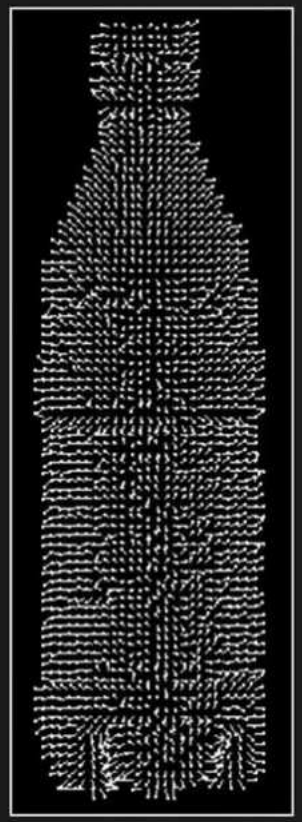
$$p(x,y) = \iint_{-\infty}^{\infty} P(u,v) e^{i2\pi(ux+vy)} du dv$$

$$q(x,y) = \iint_{-\infty}^{\infty} Q(u,v) e^{i2\pi(ux+vy)} du dv$$

Substitute for $z(x,y)$, $p(x,y)$ and $q(x,y)$ in equation for D.

[Frankot 1988]

Results: 3D Surface Reconstruction



Surface Normals

Estimated Depth Map

$z = f(x, y)$

Estimated Surface
(Rendered)

Frankot-Chellappa Algorithm

Find $Z(u,v)$ that minimizes D using $\frac{\partial D}{\partial Z} = 0$.

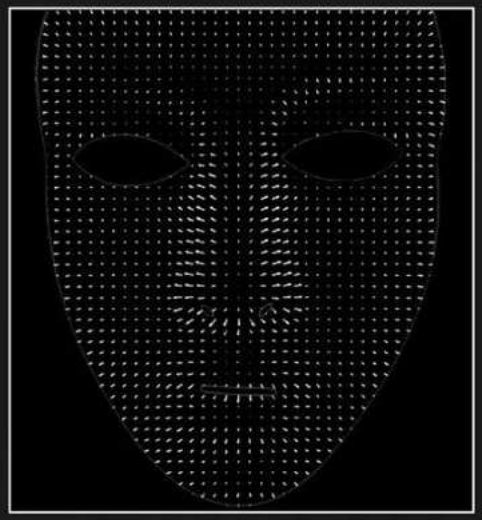
Solution: $\tilde{Z}(u,v) = \frac{iuP(u,v) + ivQ(u,v)}{u^2 + v^2}$

This is the Fourier Transform of the best fit surface.

Compute Inverse Fourier Transform to obtain $\tilde{z}(x,y)$.

[Frankot 1988]

Results: 3D Surface Reconstruction

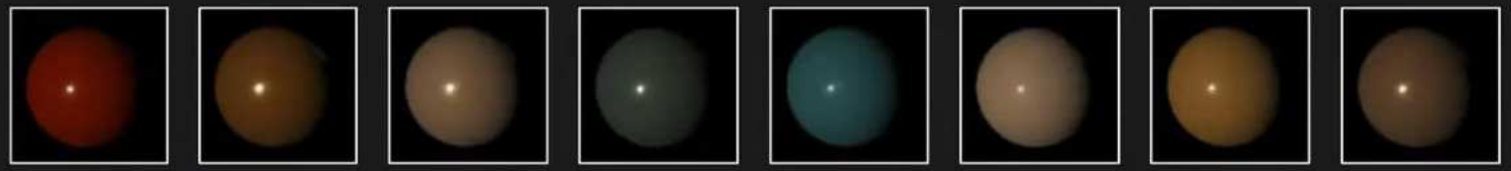


Surface Normals

Estimated Depth Map
$z = f(x, y)$

Estimated Surface
(Rendered)

Results: Fish Toy



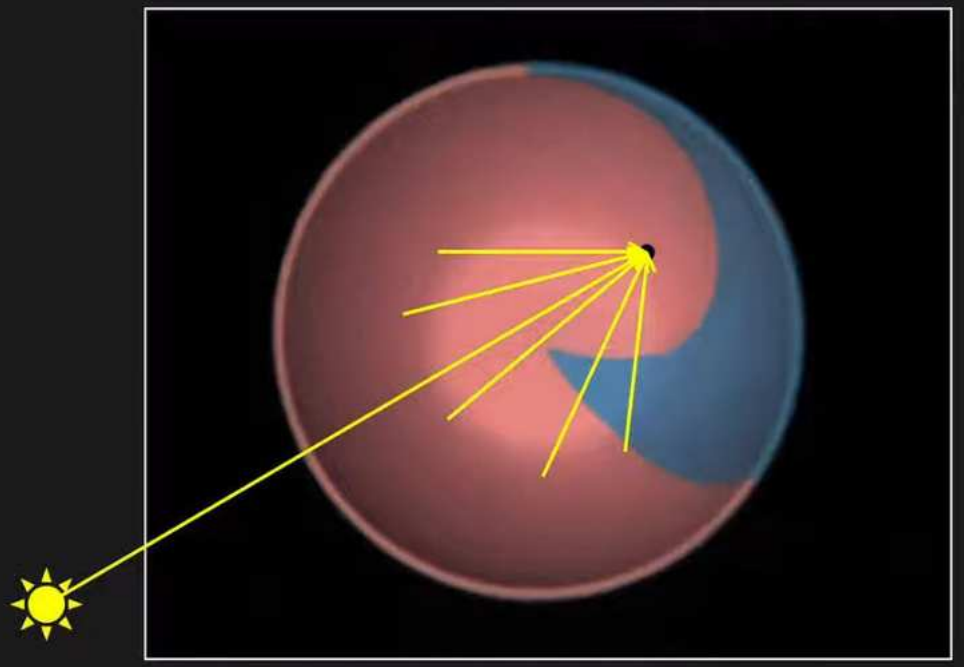
Calibration Spheres (Images under one light source)

Scene
(Image under one light source of 14)

Computed Surface
(Rendered)

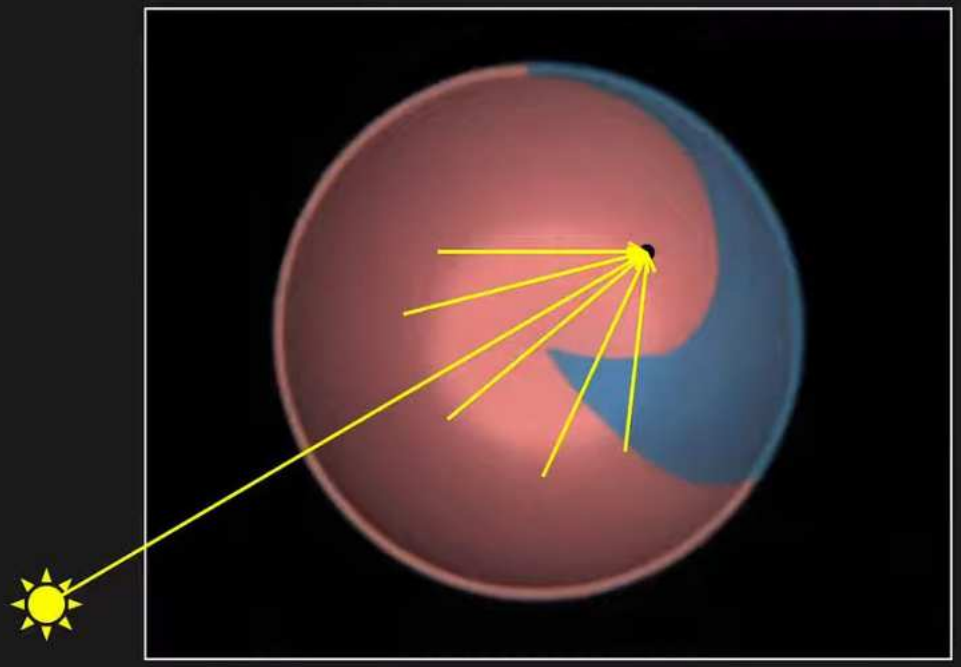
[Hertzmann 2005]

Interreflection Problem



Brightness of a scene point is not only due to the light source but also due to light from other scene points.

Interreflection Problem



Brightness of a scene point is not only due to the light source but also due to light from other scene points.

Photometric stereo overestimates albedo and underestimates surface tilt.

Shape from Interreflection

Iteratively estimate the correct shape taking into account the interreflections within the bowl.

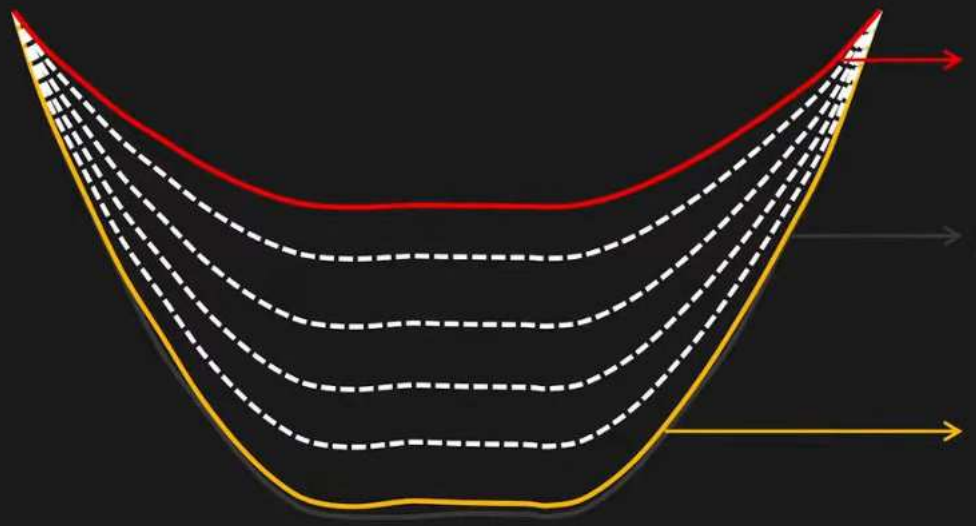
Incorrect profile obtained using photometric stereo

Actual shape

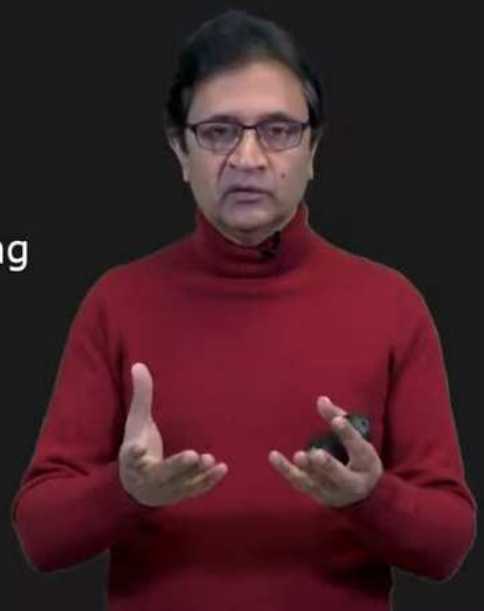
[Nayar 1991]

Shape from Interreflection

Iteratively estimate the correct shape taking into account the interreflections within the bowl.



[Nayar 1991]



Shape from a Single Image?

Given image I, source direction $\mathbf{s}$ and surface reflectance

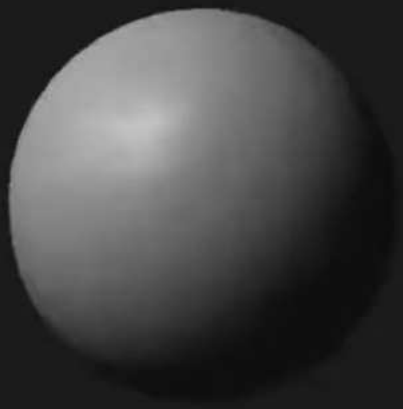
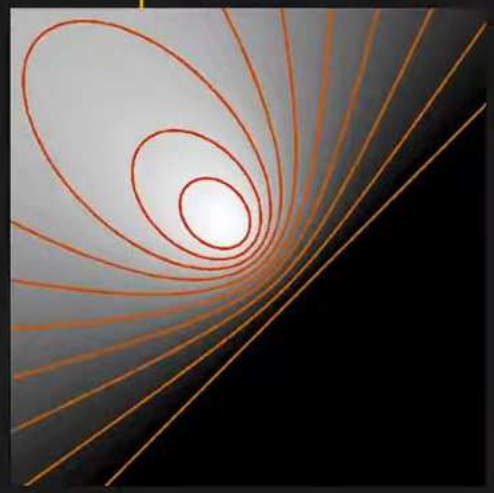


Image I



Reflectance Map $R(p, q)$

REVIEW

Shape from a Single Image?

Given image I, source direction $\mathbf{s}$ and surface reflectance

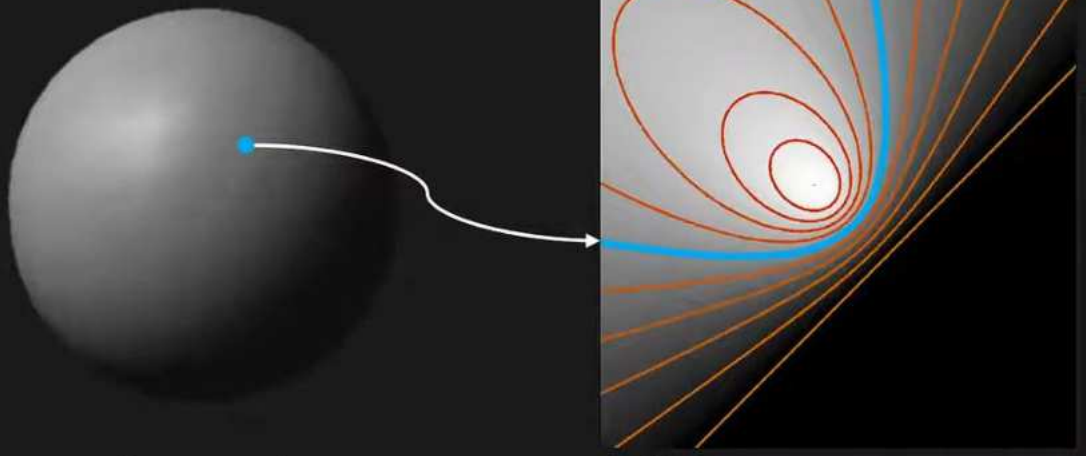
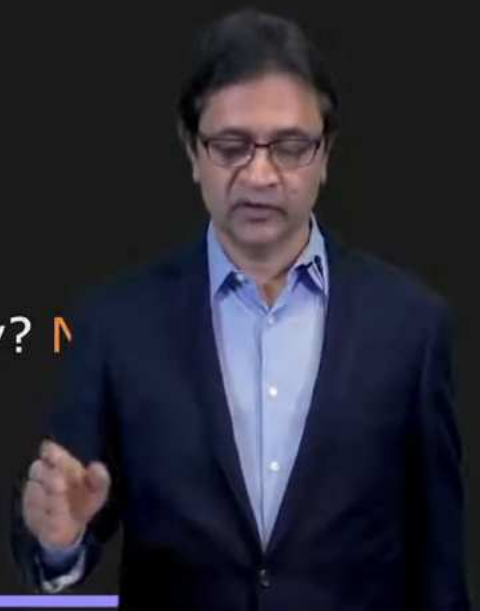


Image I

Reflectance Map $R(p, q)$

Can we estimate surface gradient (p, q) from single intensity? N



Shape From Shading

Method for recovering 3D shape information from a single image using shading.

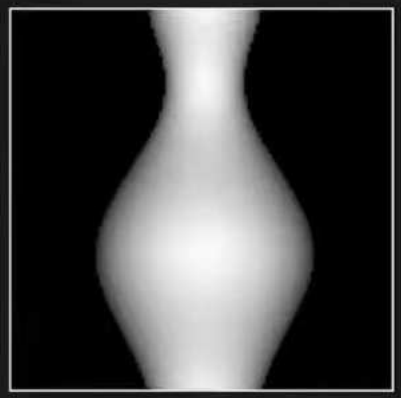
Topics:

(1) Human Perception of Shading

(2) Shape From Shading Algorithm

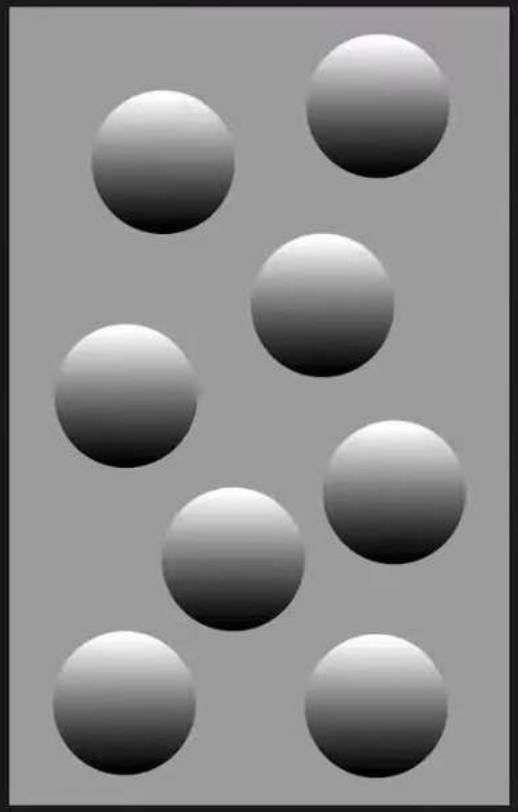
Shape From Shading in Humans

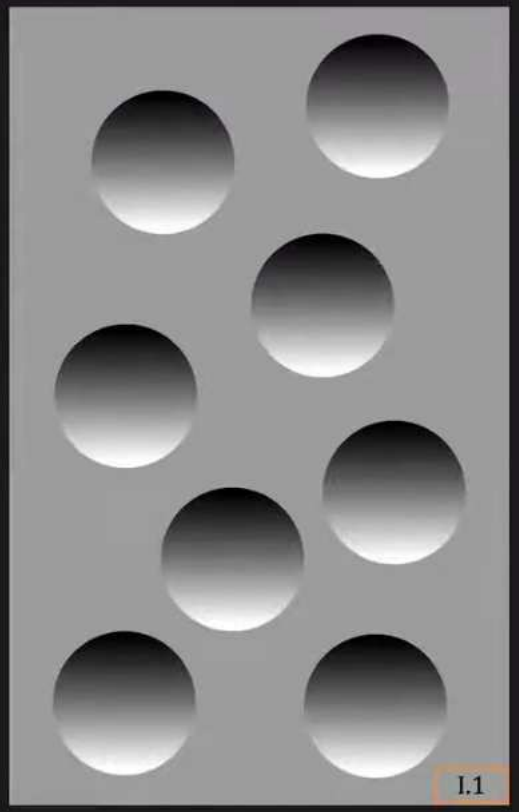
We seem to perceive shape from a single shaded image



We make many assumptions in doing so

We Assume Light Source is Above Us!

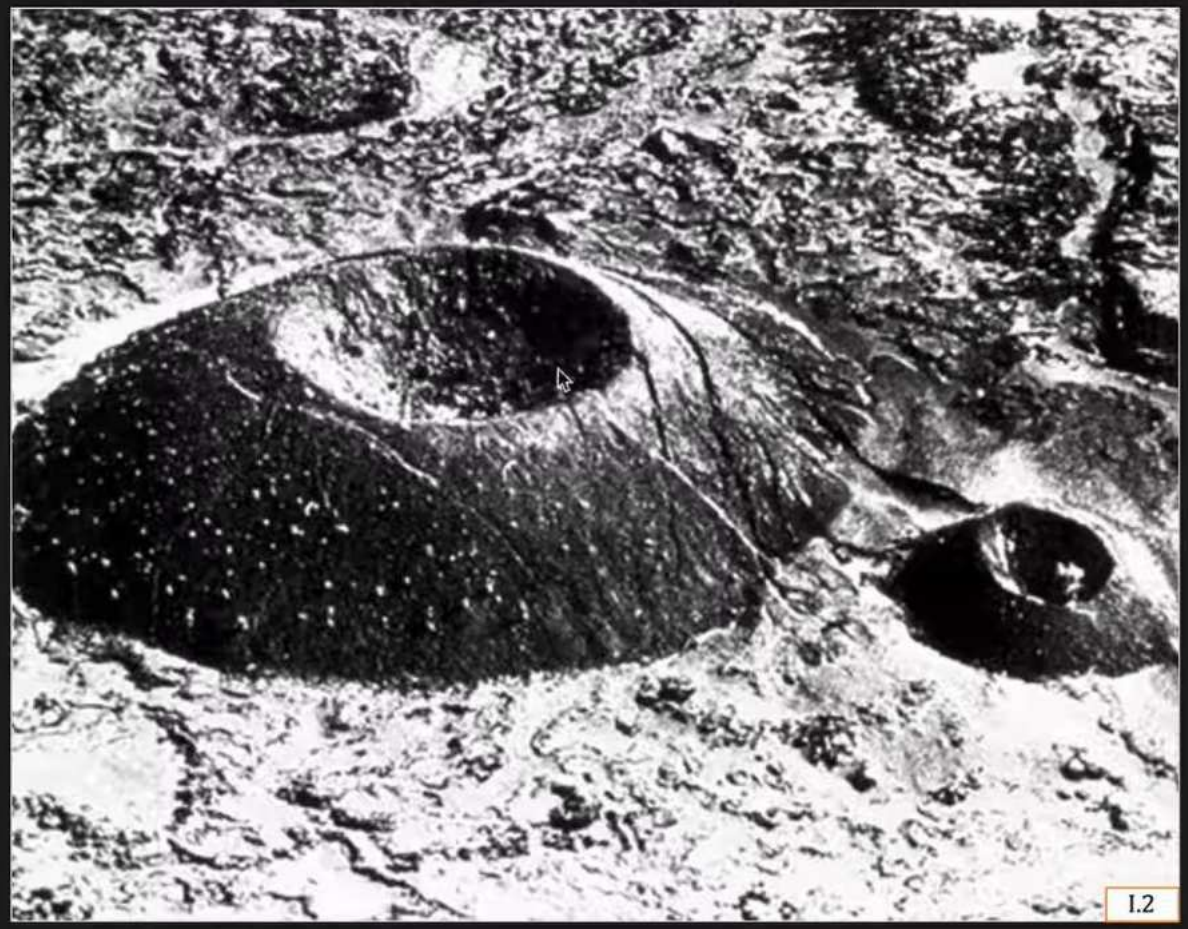




The shaded objects in the left panel are usually seen as convex, whereas those in the right panel are usually seen as concave.

[Ramachandran 1990]

We Assume Light Source is Above Us!



Crater on a Mound

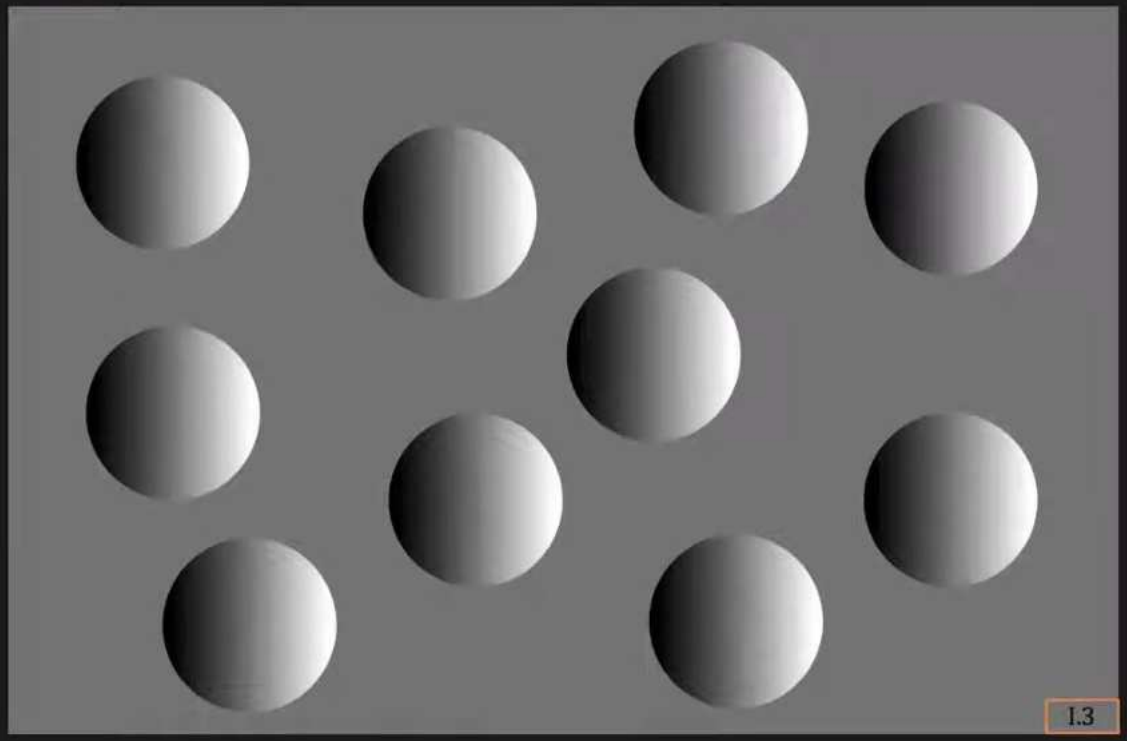
[Rittenhouse 1786]

We Assume Light Source is Above Us!

Mound in a Crater

[Rittenhouse 1786]

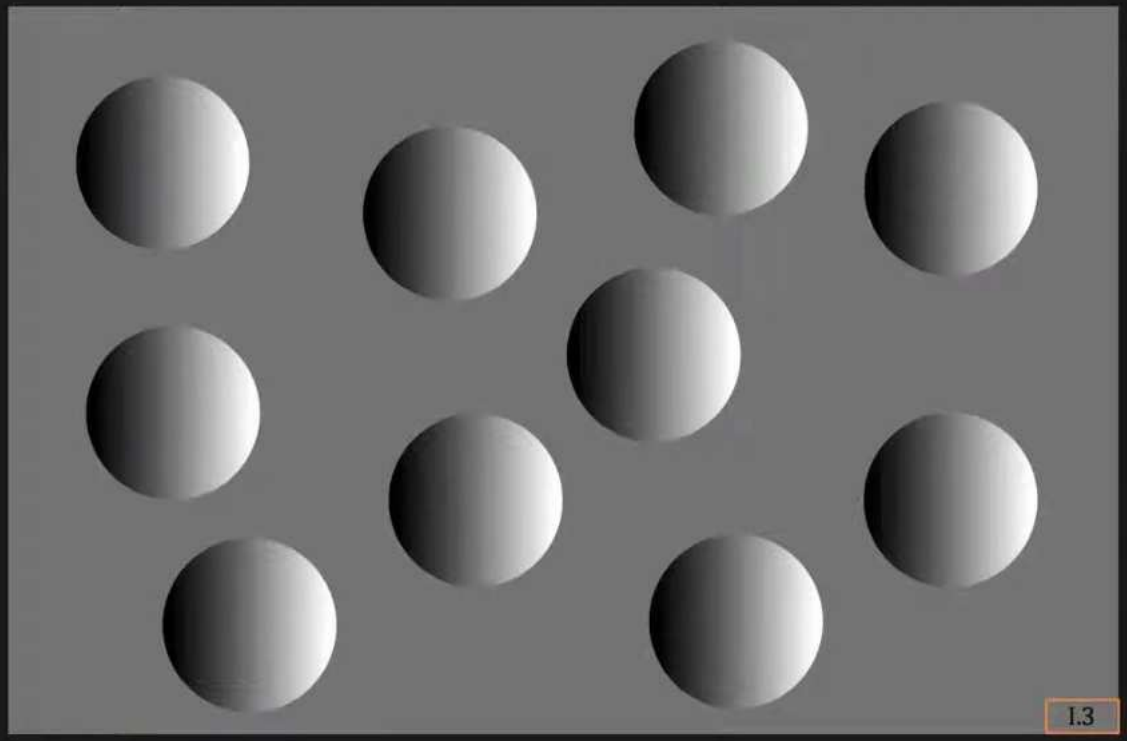
What If Illumination is Sideways?



Bumps or Cavities? It depends on where you think the light source is
You can reverse the depth of the objects by mentally shifting the lig
source from left to right.

[Ramachandran 1990]

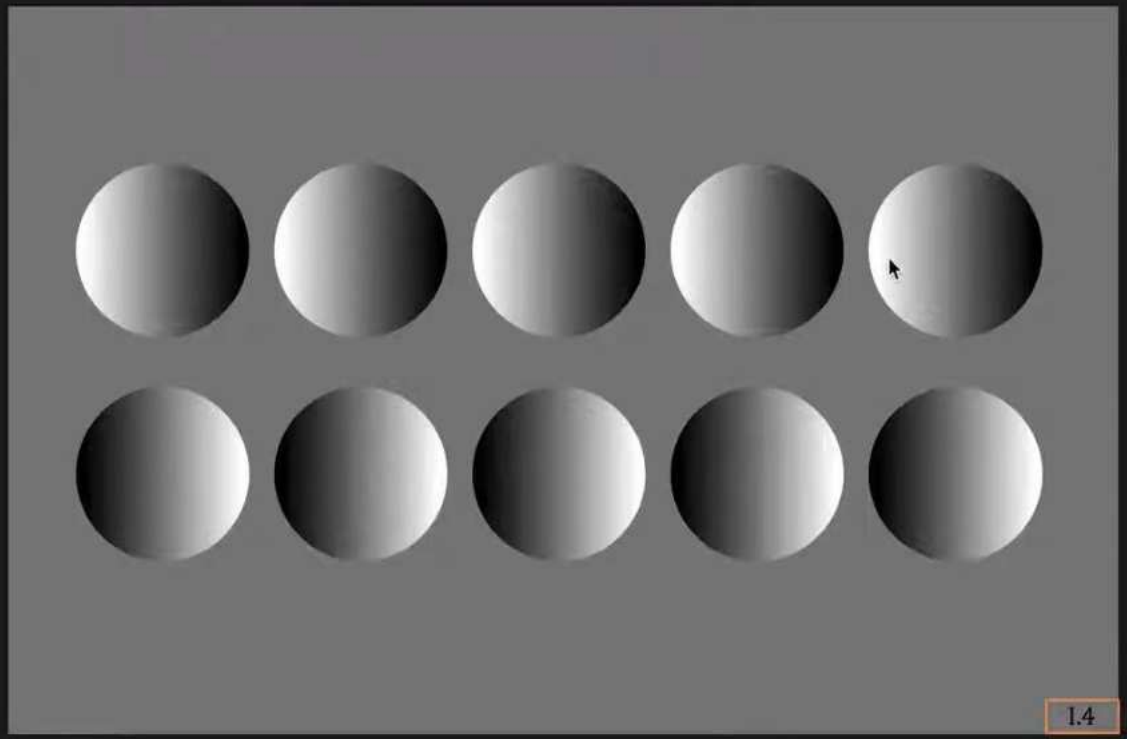
What If Illumination is Sideways?



Bumps or Cavities? It depends on where you think the light source is
You can reverse the depth of the objects by mentally shifting the ligh
source from left to right.

[Ramachandran 1990]

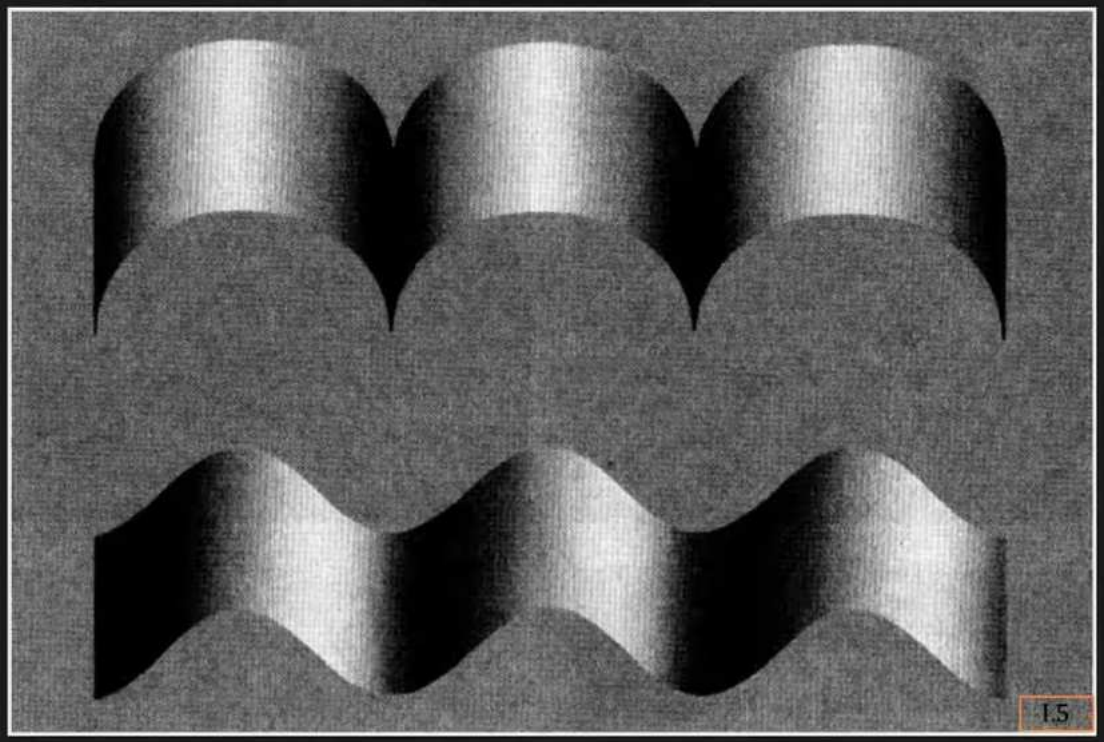
We Assume Uniform Global Illumination



Objects in one row can be seen as either convex or concave if the oth
row is excluded; but when both rows are viewed simultaneously, seei
one row as convex forces the other to be perceived as concave.

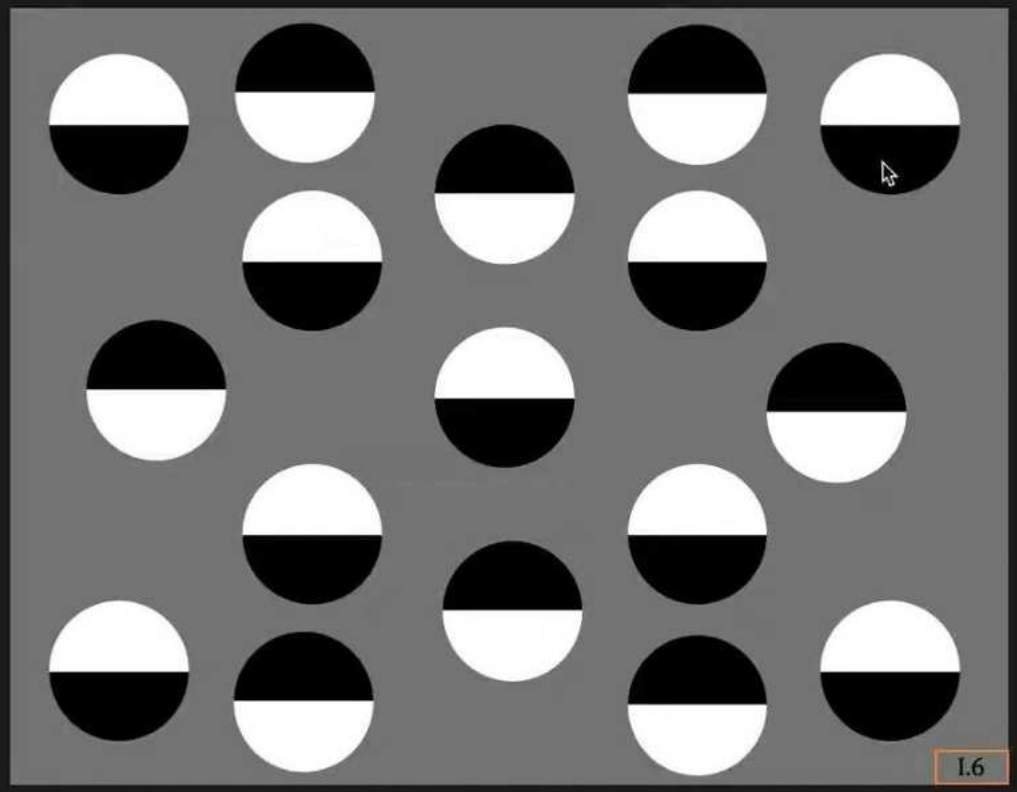
[Ramachandran 1990]

Boundaries Influence Perceived Shape



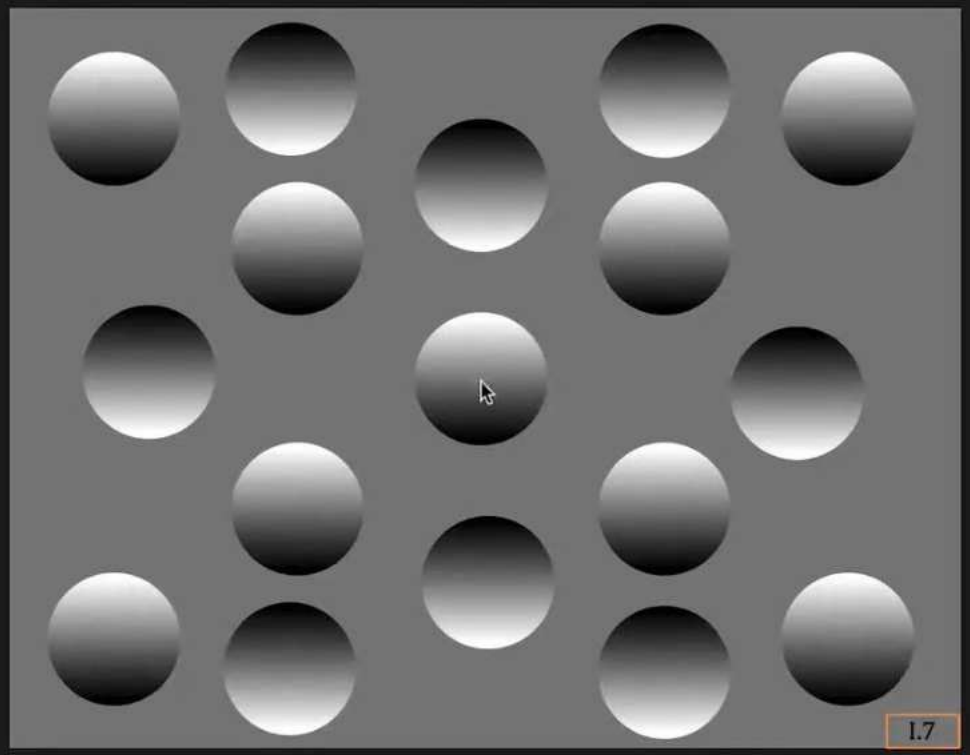
Both images have similar shading variation but the top image sugges
three cylinders lit vertically and the bottom one a corrugated sheet li
from the right.

[Ramachandran 1990]



EYE AND BRAIN

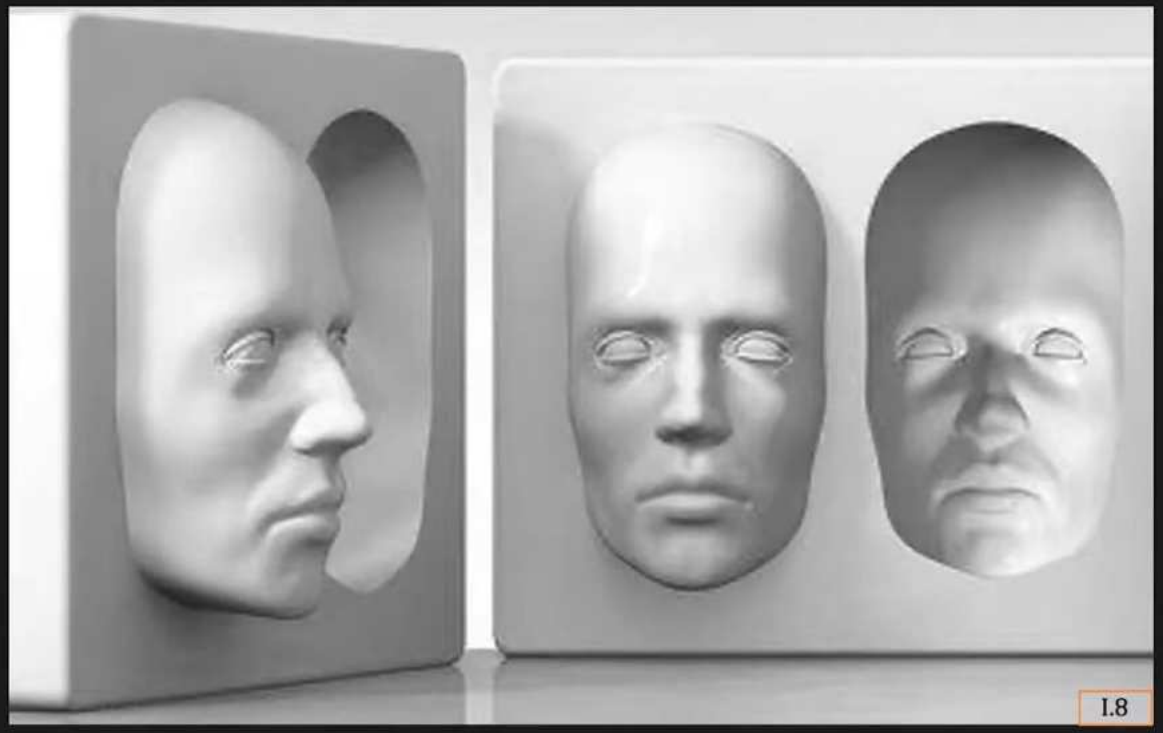
[Ramachandran 1990]



EYE AND BRAIN

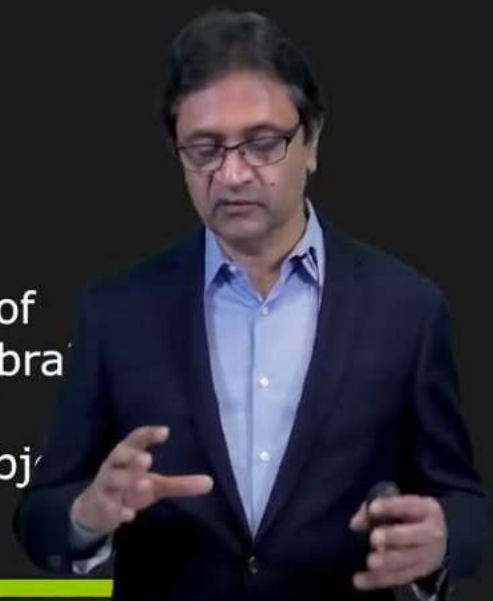
[Ramachandran 1990]

We Tend to "See" the Familiar



Hollow-Mask interiors lit from above produce an eerie impression of protruding face lit from below. In interpreting shaded images the bra usually assumes light shining from above but here it rejects the assumptions in order to interpret the images as normal, convex obj

[Ramachandran 1990]

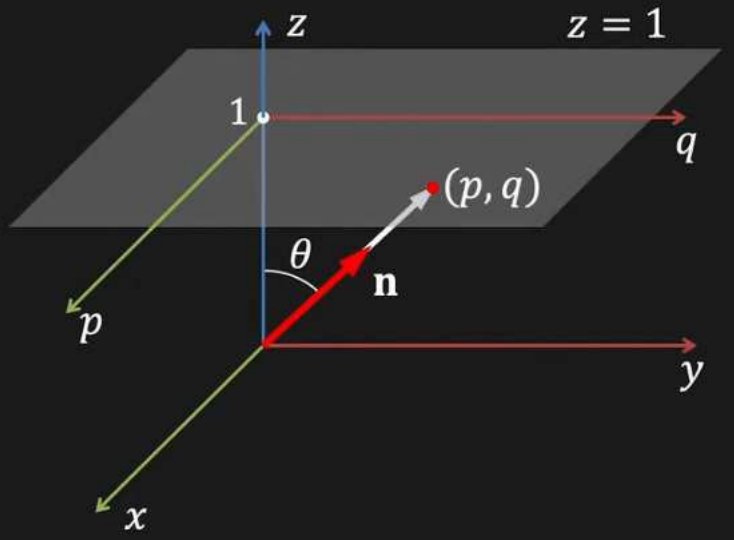


Shape From Shading?

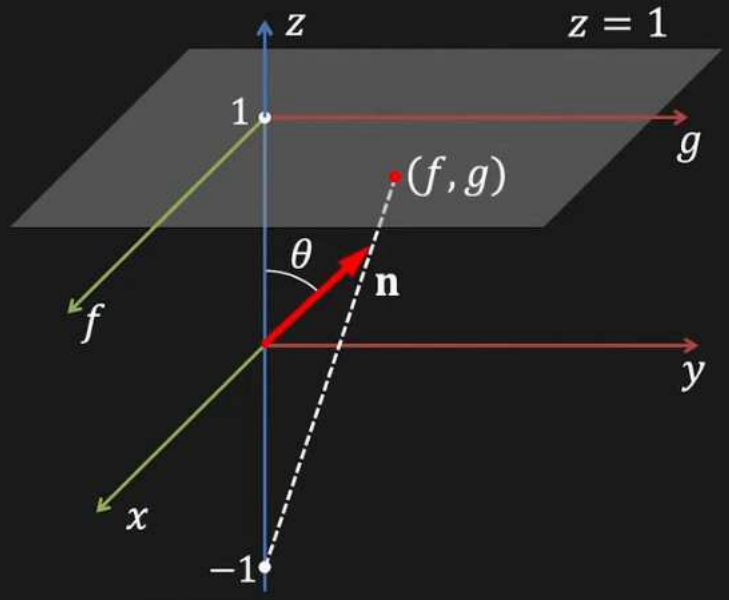
Need Assumptions and Constraints

Stereographic Projection: fg Space

pq Space



fg Space



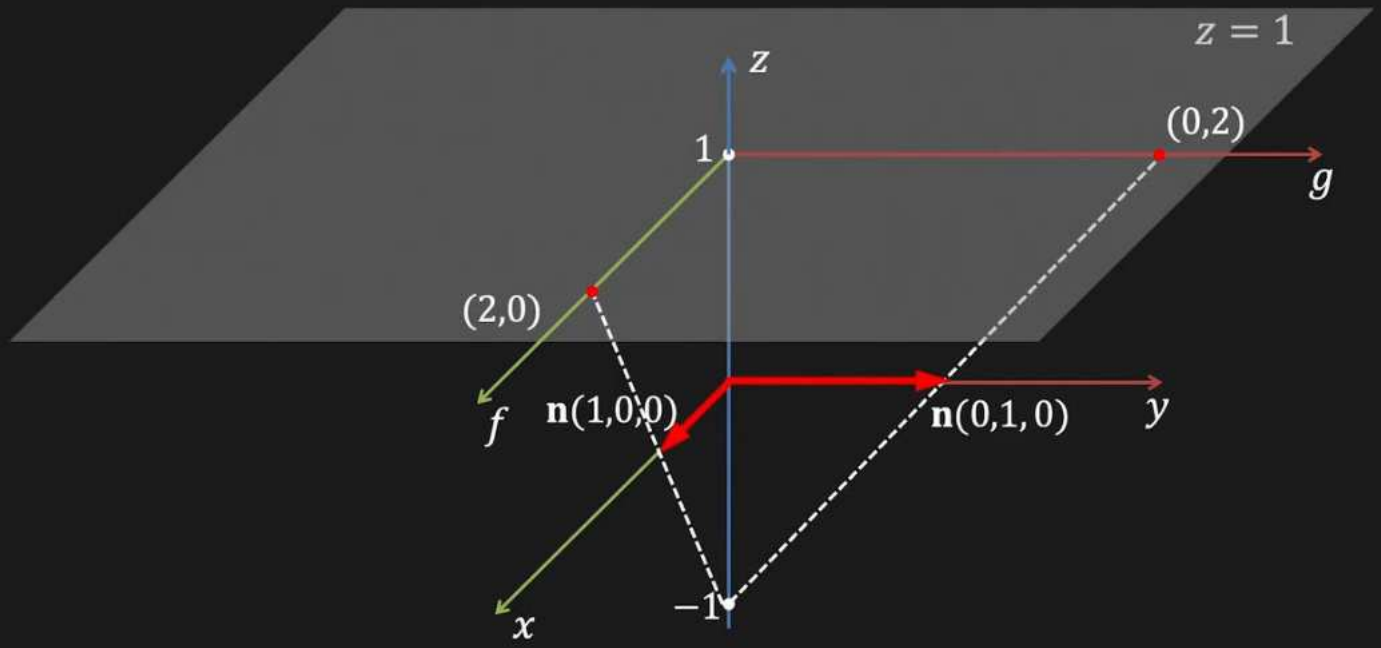
Problem:

p or q is infinite when $\theta = 90°$

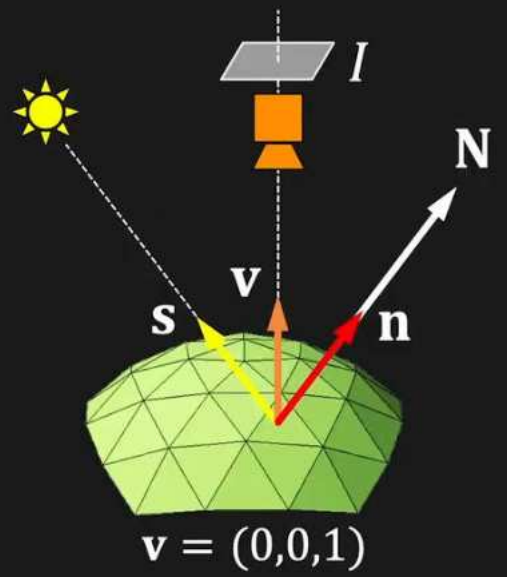
$$f = \frac{2p}{1 + \sqrt{p^2 + q^2 + 1}}$$

$$g = \frac{2q}{1 + \sqrt{p^2 + q^2 + 1}}$$

Stereographic Projection: fg Space



Reflectance Map $R_s(f,g)$



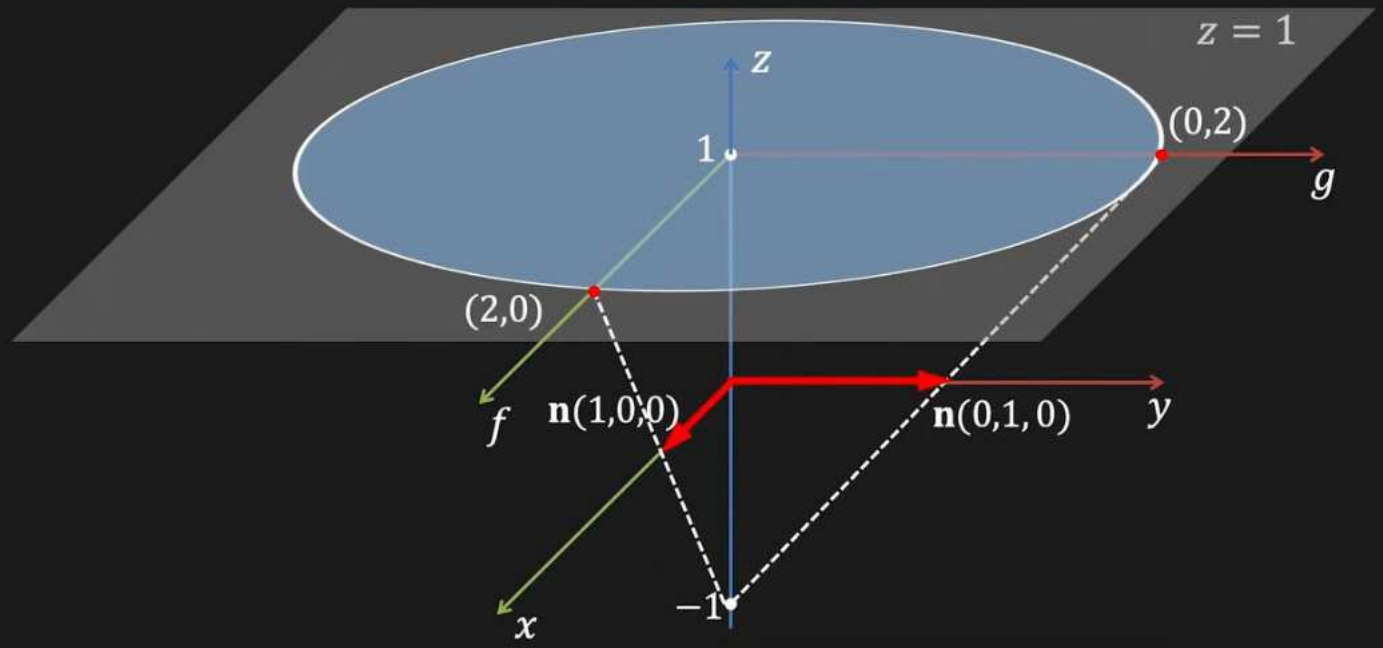
Surface normal:

$\mathbf{n} \equiv (p,q) \equiv (f,g)$

For a given source direction **s** and surface reflectance, reflectance map relates image intensity to its surface gradients:

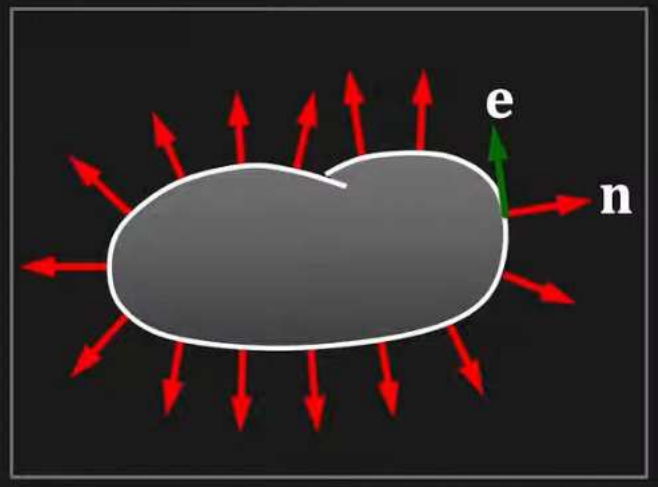
$$I = R_s(f,g)$$

Stereographic Projection: fg Space

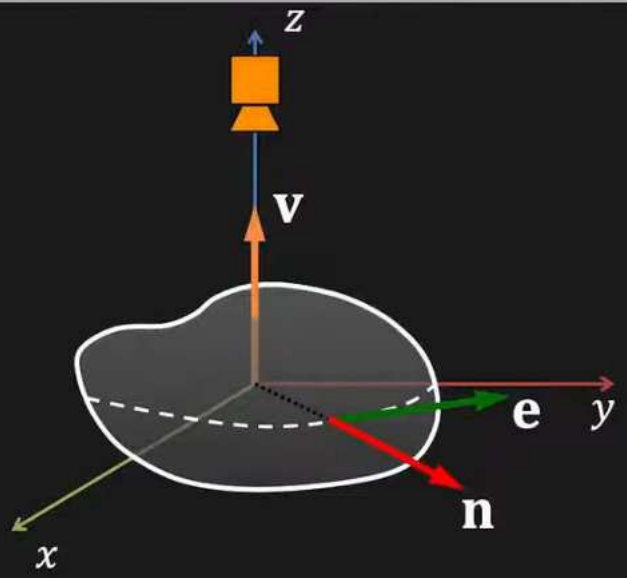


All possible values of surface gradients
lie within a circle of radius 2 on the fg Plane.

Occluding Boundaries



Captured Image



$\mathbf{n} \perp \mathbf{v} \quad \mathbf{n} \perp \mathbf{e}$

($\mathbf{e}$ and $\mathbf{v}$ are known)

$$\mathbf{n} = \mathbf{e} \times \mathbf{v}$$

Surface gradients (f, g) on occluding boundary are known and can be used as boundary conditions.

Image Irradiance Constraint

Assumption: Image irradiance (intensity) should equal the reflectance map. That is, $I(x, y) = R_s(f, g)$.

Minimize:

$$e_r = \iint (I(x, y) - R_s(f, g))^2 \, dx \, dy$$

Aim: Penalize errors between image irradiance and reflectance map.

Smoothness Constraint

Assumption: Object surface is smooth. That is, the gradient values (f, g) vary slowly.

Minimize:

$$e_s = \iint \left(f_x^2 + f_y^2\right) + \left(g_x^2 + g_y^2\right) dx\, dy$$

where: $f_x = \frac{\partial f}{\partial x}$, $f_y = \frac{\partial f}{\partial y}$, $g_x = \frac{\partial g}{\partial x}$ and $g_y = \frac{\partial g}{\partial y}$

Aim: Penalize rapid changes in f and g during surface estimation.

Shape from Shading

Find surface gradients (f, g) at all image points that minimize the function:

$$e = e_s + \lambda e_r$$

where:

e_s: Smoothness Constraint

e_r: Image Irradiance Error

λ: Weight

[Ikeuchi 1981]

Shape from Shading

Find surface gradients (f, g) at all image points that minimize the function:

$$e = e_s + \lambda e_r$$

where:

e_s: Smoothness Constraint

e_r: Image Irradiance Error

λ: Weight

Known surface gradients (f, g) on occluding boundary are held constant.

[Ikeuchi 1981]

Numerical Shape from Shading

Smoothness Error at point (i, j):

$$e_{s\,i,j} = \frac{1}{4}\Big(\big(f_{i+1,j} - f_{i,j}\big)^2 + \big(f_{i,j+1} - f_{i,j}\big)^2 + \big(g_{i+1,j} - g_{i,j}\big)^2 + \big(g_{i,j+1} - g_{i,j}\big)^2\Big)$$

Image Irradiance Error at point (i, j):

$$e_{r\,i,j} = \Big(I_{i,j} - R_s\big(f_{i,j}, g_{i,j}\big)\Big)^2$$

[Ikeuchi 1981]

Numerical Shape from Shading

Smoothness Error at point (i,j):

$$e_{s\,i,j} = \frac{1}{4}\Big(\big(f_{i+1,j} - f_{i,j}\big)^2 + \big(f_{i,j+1} - f_{i,j}\big)^2 + \big(g_{i+1,j} - g_{i,j}\big)^2 + \big(g_{i,j+1} - g_{i,j}\big)^2\Big)$$

Image Irradiance Error at point (i,j):

$$e_{r\,i,j} = \Big(I_{i,j} - R_s\big(f_{i,j}, g_{i,j}\big)\Big)^2$$

Find $\big(f_{i,j}, g_{i,j}\big)$ for all (i,j) that minimizes:

$$\boxed{e = \sum_i \sum_j \Big(e_{s\,i,j} + \lambda\, e_{r\,i,j}\Big)}$$

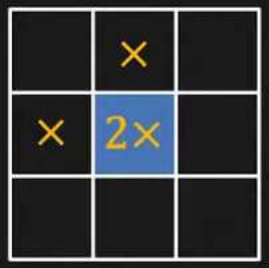
[Ikeuchi 1981]

Numerical Shape from Shading

If $(f_{k,l}, g_{k,l})$ minimizes e, then $\frac{\partial e}{\partial f_{k,l}} = 0$ and $\frac{\partial e}{\partial g_{k,l}} = 0$

Given an image of size $N \times N$, there are $2N^2$ unknowns.
(N^2 $f_{i,j}$'s and N^2 $g_{i,j}$'s)

However, note that each $f_{i,j}$ and $g_{i,j}$ appears in 4 terms in $e = \sum_i \sum_j \left(e_{s_{i,j}} + \lambda\, e_{r_{i,j}} \right)$:



Numerical Shape from Shading

If $(f_{k,l}, g_{k,l})$ minimizes e, then $\frac{\partial e}{\partial f_{k,l}} = 0$ and $\frac{\partial e}{\partial g_{k,l}} = 0$

Therefore:

Eq 1: $$\frac{\partial e}{\partial f_{k,l}} = 2(f_{k,l} - \bar{f}_{k,l}) - 2\lambda \left(I_{k,l} - R_s(f_{k,l}, g_{k,l})\right) \left.\frac{\partial R_s}{\partial f}\right|_{f_{k,l}, g_{k,l}} = 0$$

Eq 2: $$\frac{\partial e}{\partial g_{k,l}} = 2(g_{k,l} - \bar{g}_{k,l}) - 2\lambda \left(I_{k,l} - R_s(f_{k,l}, g_{k,l})\right) \left.\frac{\partial R_s}{\partial g}\right|_{f_{k,l}, g_{k,l}} = 0$$

where $\bar{f}_{k,l}$ and $\bar{g}_{k,l}$ are local averages:

$$\bar{f}_{k,l} = \frac{1}{4}\left(f_{k+1,l} + f_{k-1,l} + f_{k,l+1} + f_{k,l-1}\right)$$

$$\bar{g}_{k,l} = \frac{1}{4}\left(g_{k+1,l} + g_{k-1,l} + g_{k,l+1} + g_{k,l-1}\right)$$

Numerical Shape from Shading

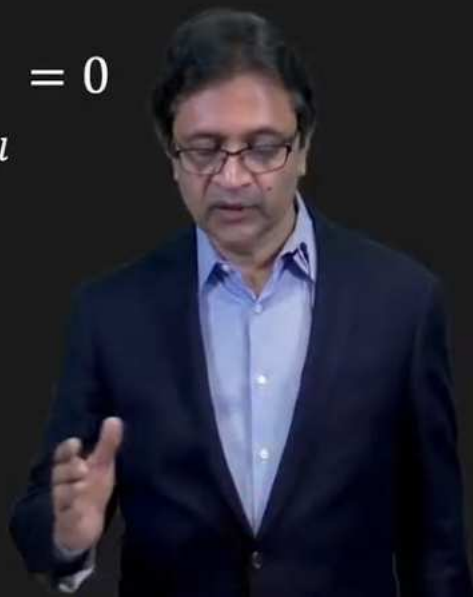
If $(f_{k,l}, g_{k,l})$ minimizes e, then $\frac{\partial e}{\partial f_{k,l}} = 0$ and $\frac{\partial e}{\partial g_{k,l}} = 0$

Therefore:

Eq 1: $$\frac{\partial e}{\partial f_{k,l}} = 2\left(f_{k,l} - \bar{f}_{k,l}\right) - 2\lambda\left(I_{k,l} - R_s\left(f_{k,l}, g_{k,l}\right)\right)\left.\frac{\partial R_s}{\partial f}\right|_{f_{k,l},\, g_{k,l}} = 0$$

Eq 2: $$\frac{\partial e}{\partial g_{k,l}} = 2\left(g_{k,l} - \bar{g}_{k,l}\right) - 2\lambda\left(I_{k,l} - R_s\left(f_{k,l}, g_{k,l}\right)\right)\left.\frac{\partial R_s}{\partial g}\right|_{f_{k,l},\, g_{k,l}} = 0$$

Moving $f_{k,l}$ and $g_{k,l}$ to one side, we get……



Iterative Solution

Update Rule:

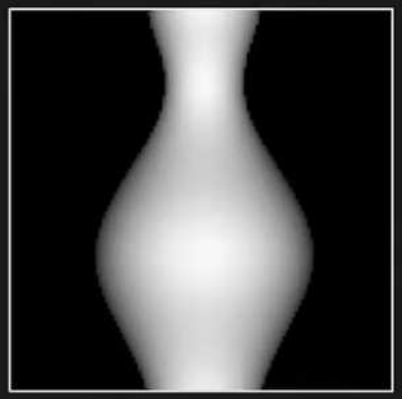
$$f_{k,l}^{(n+1)} = \bar{f}_{k,l}^{(n)} + \lambda \left(I_{k,l} - R_s \left(f_{k,l}^{(n)}, g_{k,l}^{(n)} \right) \right) \left. \frac{\partial R_s}{\partial f} \right|_{f_{k,l}^{(n)}, g_{k,l}^{(n)}}$$

$$g_{k,l}^{(n+1)} = \bar{g}_{k,l}^{(n)} + \lambda \left(I_{k,l} - R_s \left(f_{k,l}^{(n)}, g_{k,l}^{(n)} \right) \right) \left. \frac{\partial R_s}{\partial g} \right|_{f_{k,l}^{(n)}, g_{k,l}^{(n)}}$$

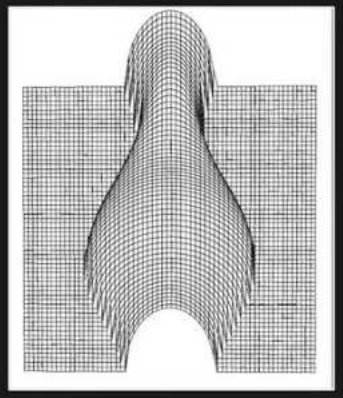
n: iteration

- Use known normals to fix (f, g) values on occluding boundary. Initialize the rest to (0,0)
- Iteratively compute (f, g) until the solution has converged.

Results: Synthetic Objects

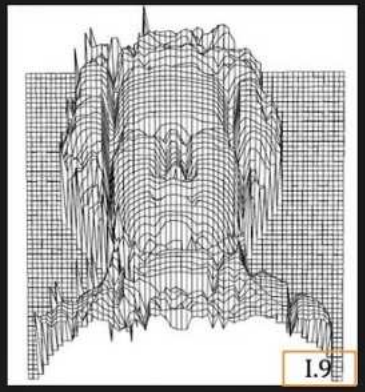


Scene



Recovered Shape

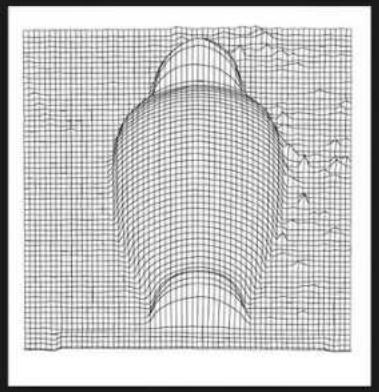
Scene



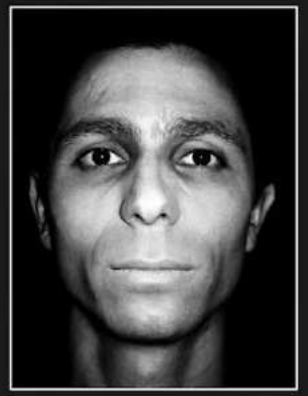
Recovered Shape

Results: Real Objects

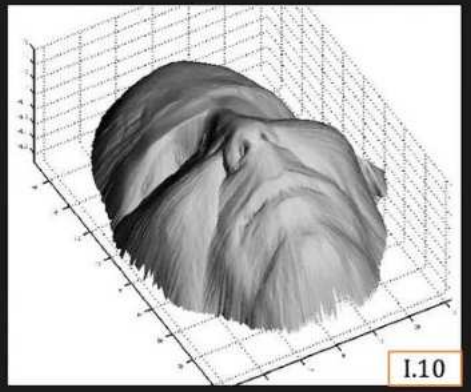
Scene



Recovered Shape

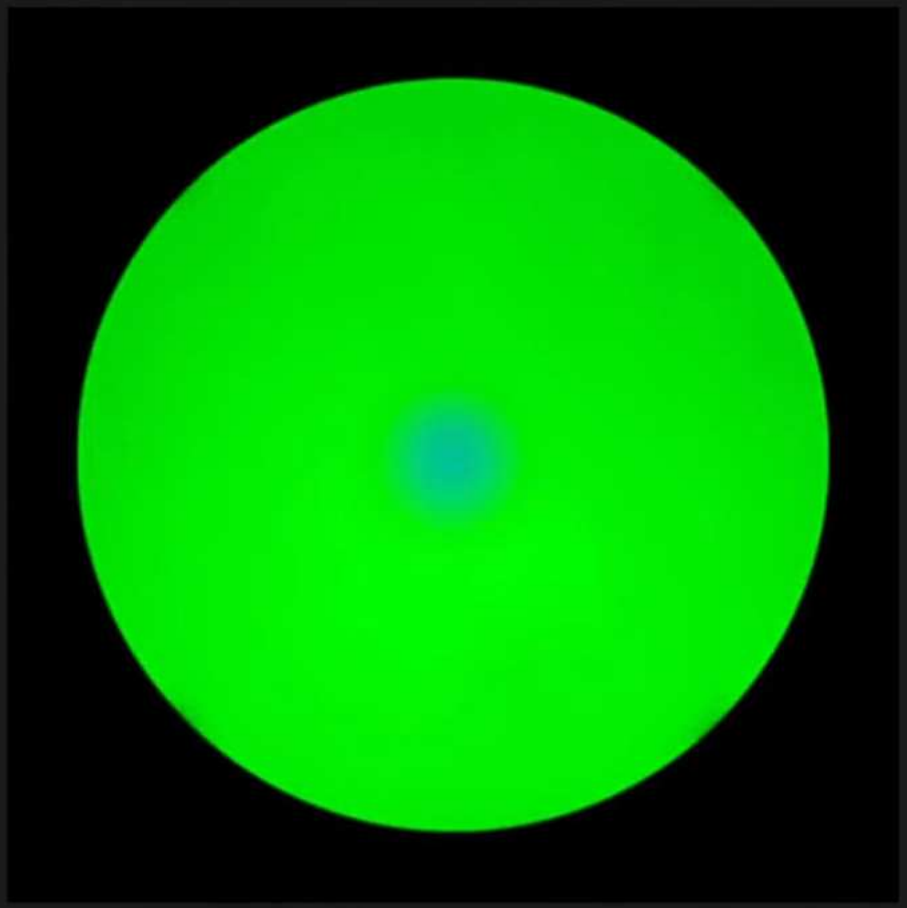


Scene



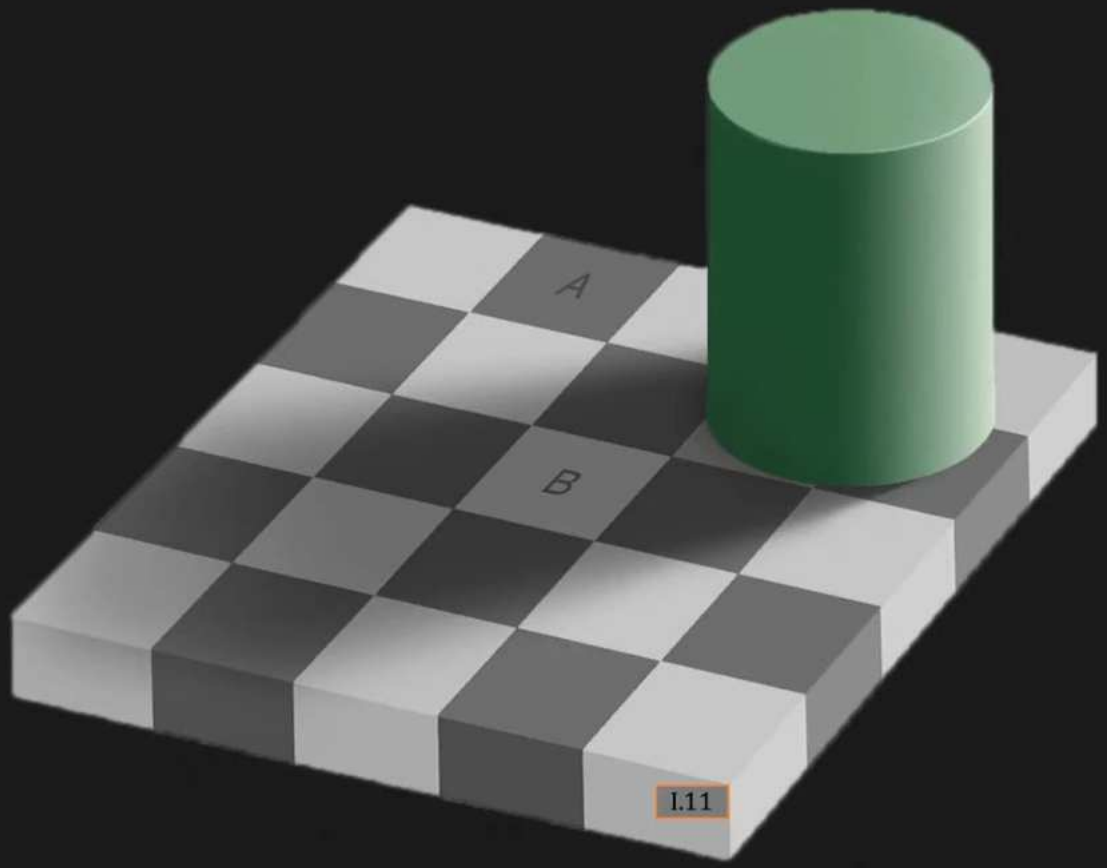
Recovered Shape

Shading Illusions: Fading Disk



Fading Disk Illusion

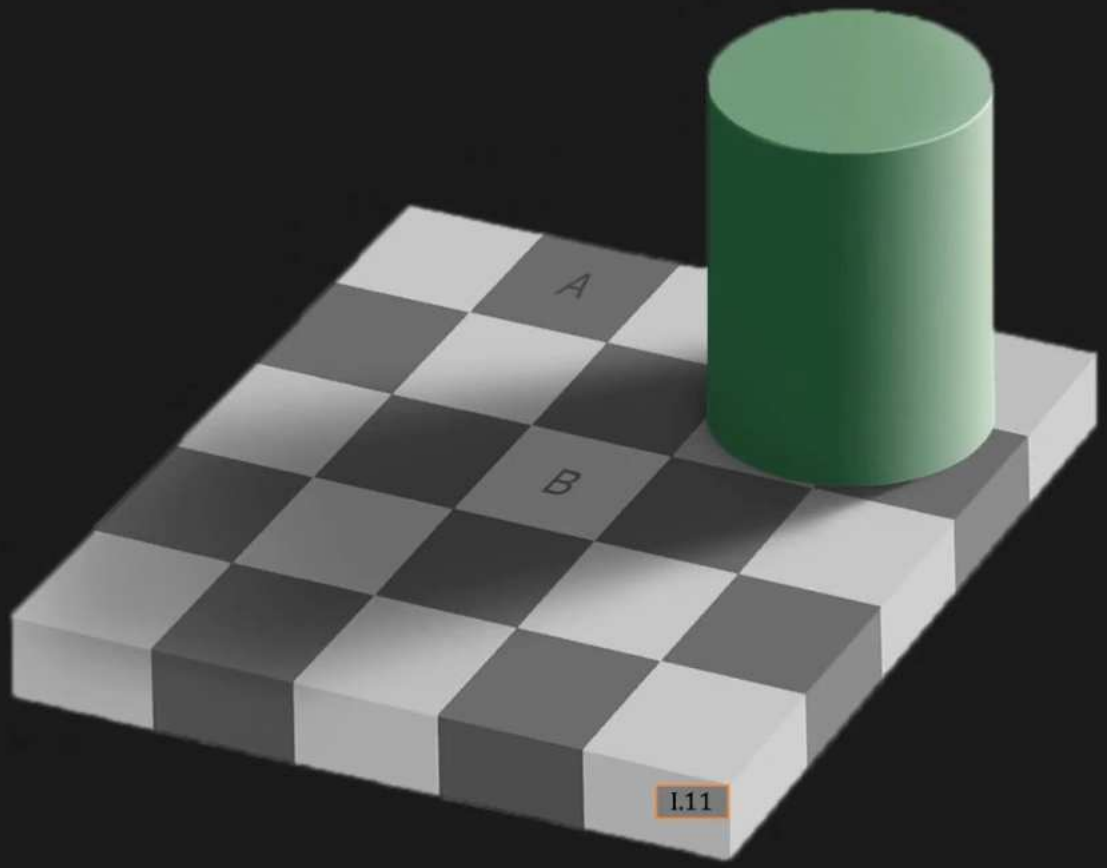
Shading Illusions: Checker Shadow



B seems Brighter than A

[Adelson 1995]

Shading Illusions: Checker Shadow



B seems Brighter than A

[Adelson 1995]

Depth from Defocus

Methods to compute depth by analyzing the degree of focus or defocus in images.

Topics:

(1) Point Spread Function

(2) Depth from Focus

Depth from Defocus

Methods to compute depth by analyzing the degree of focus or defocus in images.

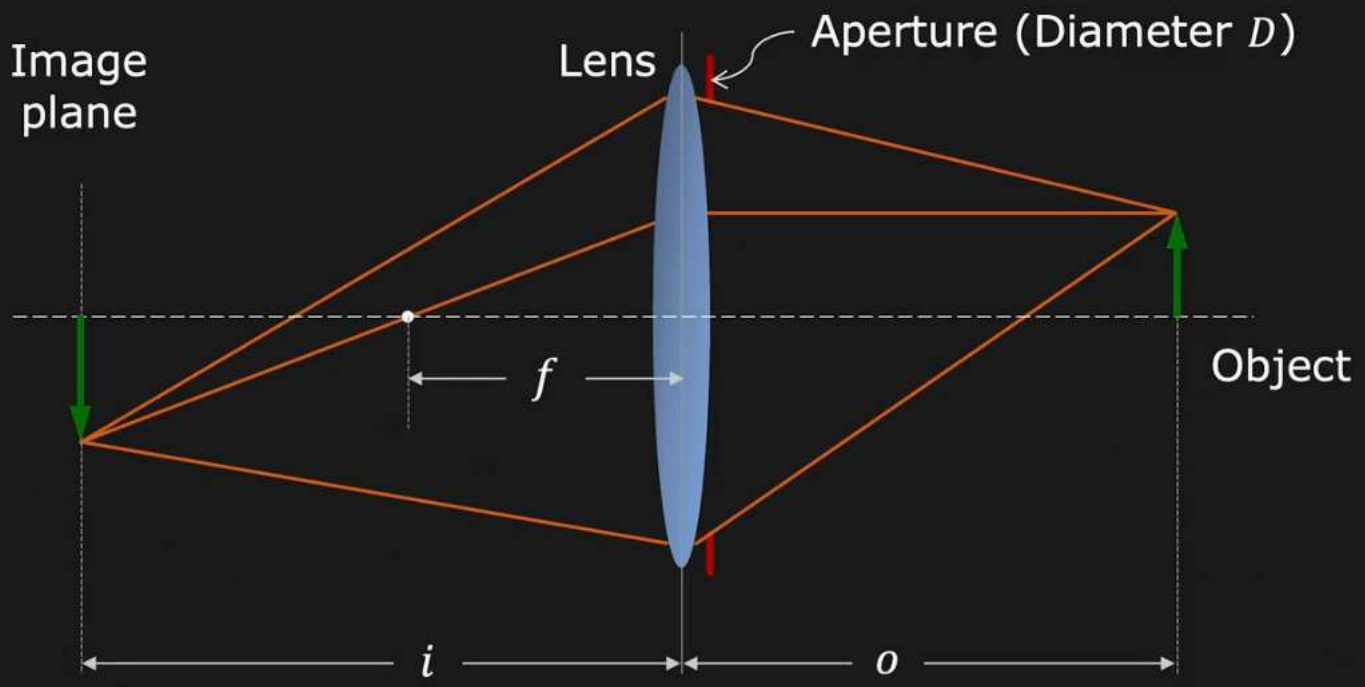
Topics:

(1) Point Spread Function

(2) Depth from Focus

(3) Depth from Defocus

Gaussian Lens Law

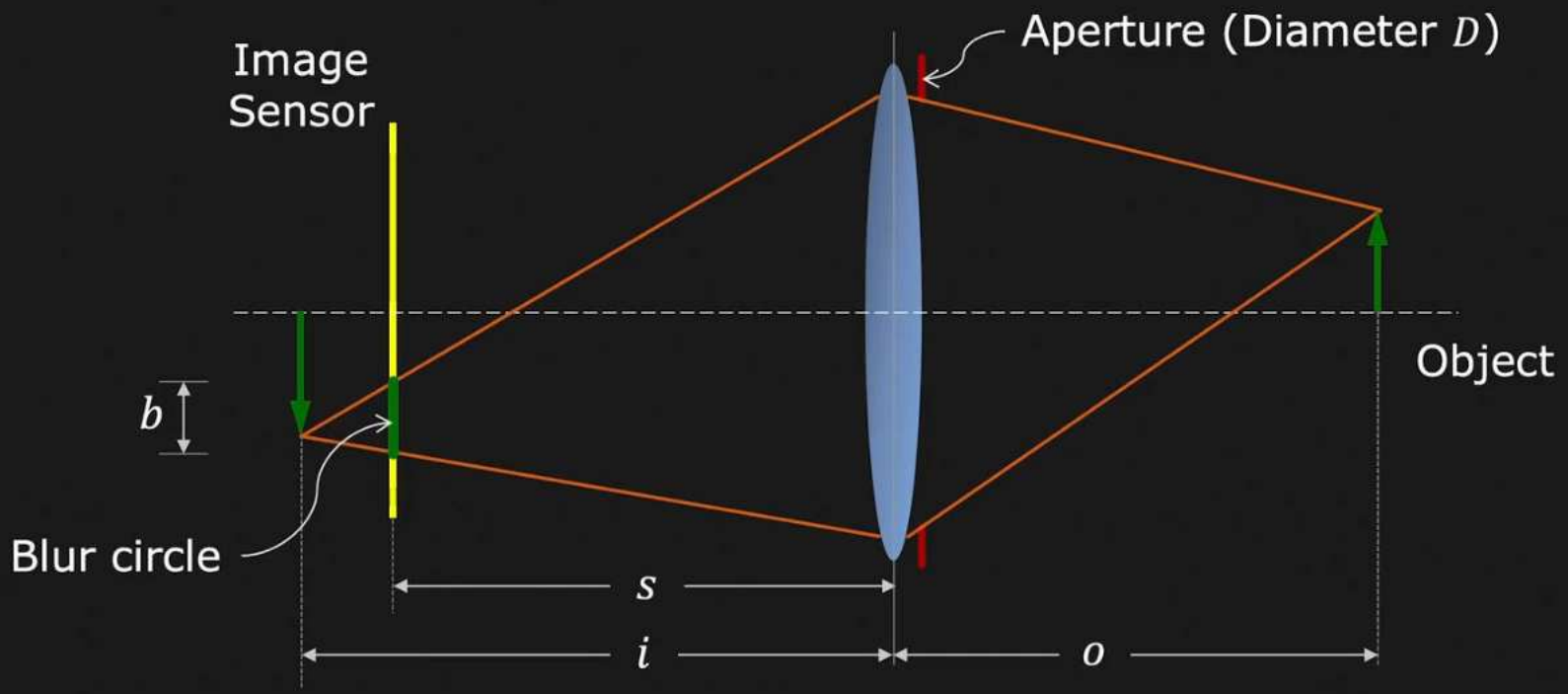


Gaussian lens law: $\frac{1}{f} = \frac{1}{i} + \frac{1}{o}$

f: Focal length

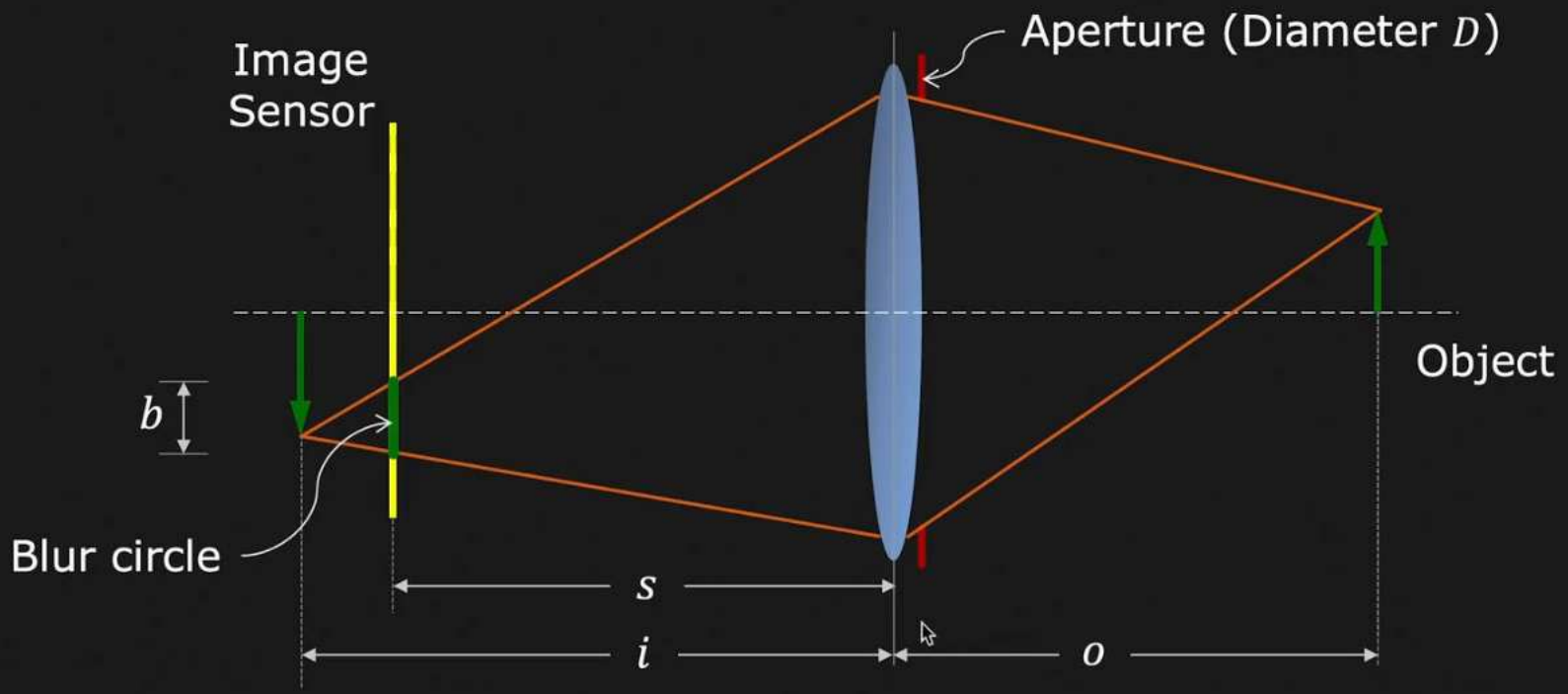
REVIEW

Geometry of Image Defocus



From Similar Triangles: $\frac{b}{D} = \left|\frac{i - s}{i}\right|$

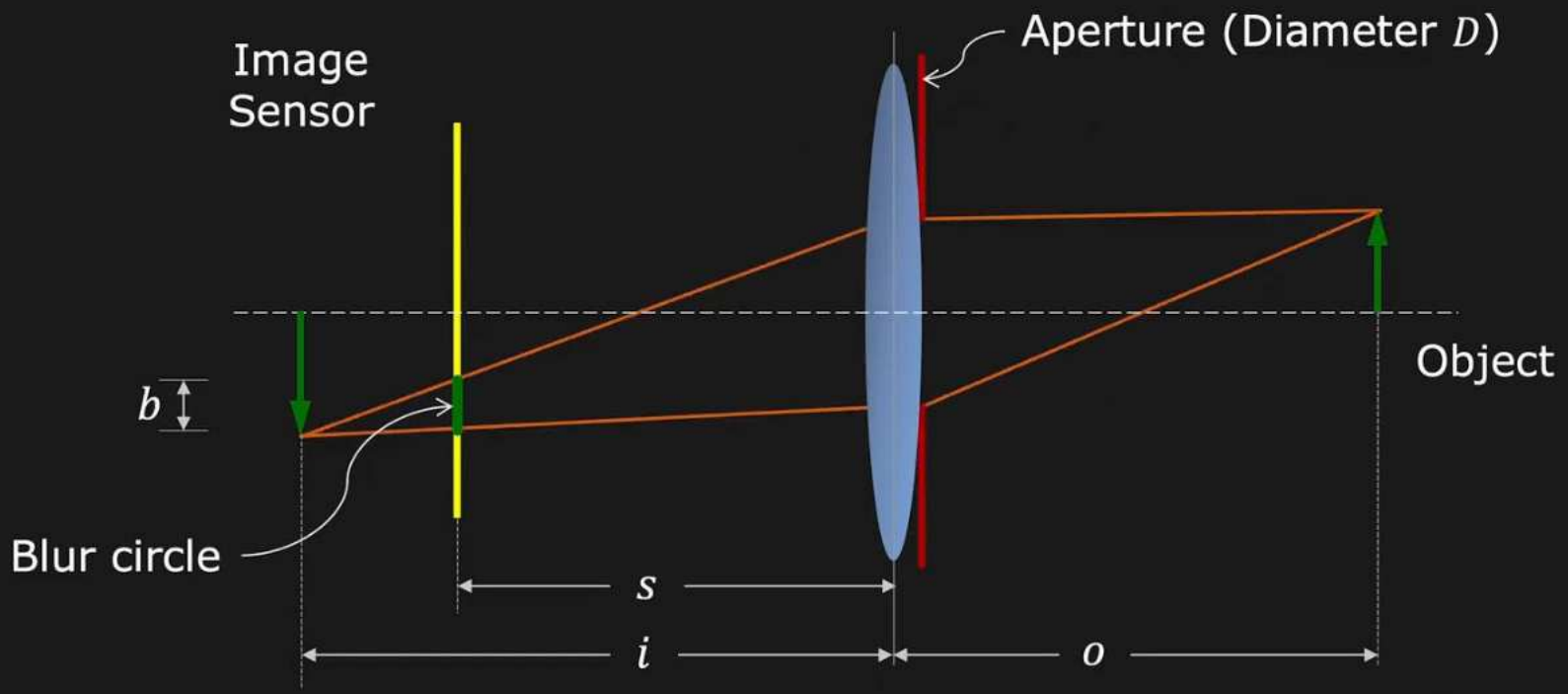
Image Defocus and Sensor Location



Blur circle diameter: $$b = D\left|1 - \frac{s}{i}\right|$$

REVIEW

Image Defocus and Aperture Size



Blur circle diameter: $b = D\left|1 - \frac{s}{i}\right|$

Smaller the aperture size D, smaller the blur circle b

Point Spread Function (PSF)

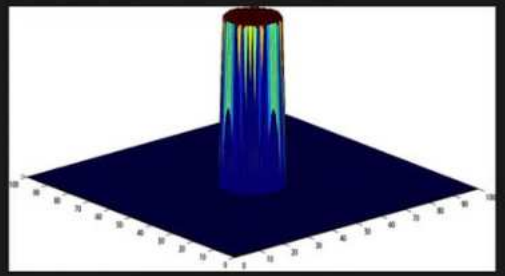
In practice, due to diffraction, lens aberrations and image sensing, the PSF often appears like a Gaussian function.

Point Spread Function (PSF)

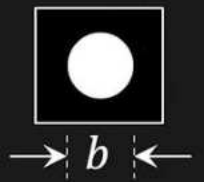
Point Spread Function (PSF): The response of a camera system to a point source (an impulse).

Pillbox (Disk) PSF:

$$h(x,y) = \begin{cases} 4/\pi b^2, & x^2 + y^2 \leq b^2/4 \\ 0, & otherwise \end{cases}$$



$h(x,y)$

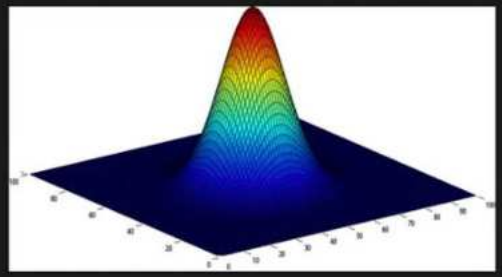


Point Spread Function (PSF)

In practice, due to diffraction, lens aberrations and image sensing, the PSF often appears like a Gaussian function.

Gaussian PSF:

$$h(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



$h(x,y)$

$\sigma = b/2$

(approximation)

Defocus as Convolution

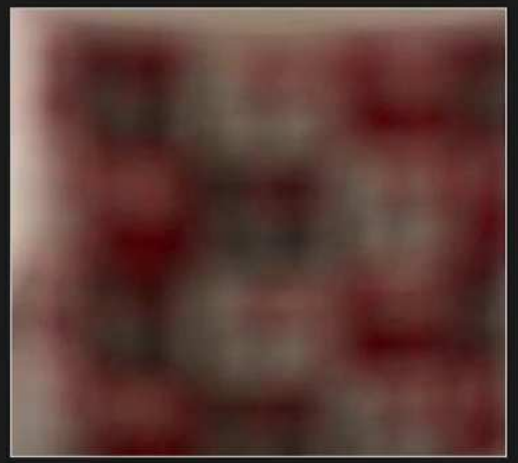
Within a region where scene depth is constant…

$f_0(x,y)$

$*$

$h(x,y)$

$=$

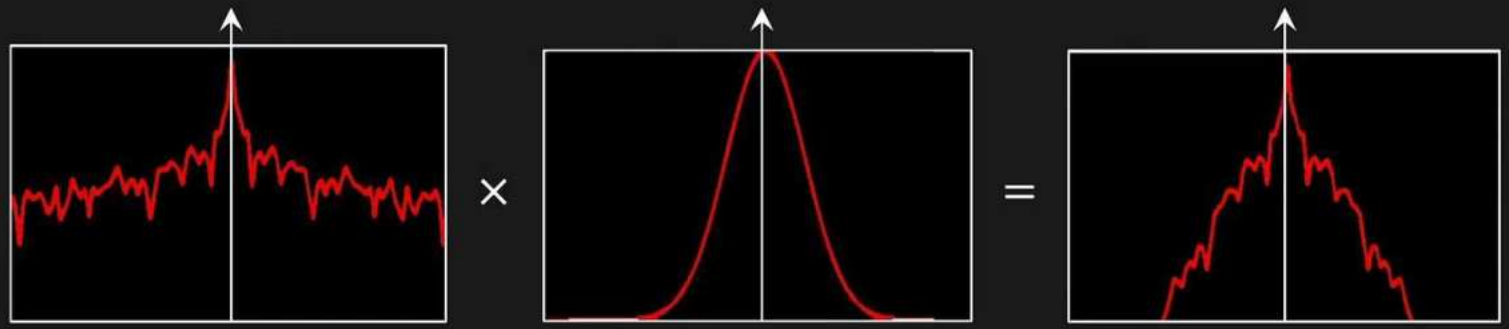


$f(x,y)$

…Defocus is linear and shift invariant, and therefore can be expressed as a convolution.

Defocus in Fourier Domain

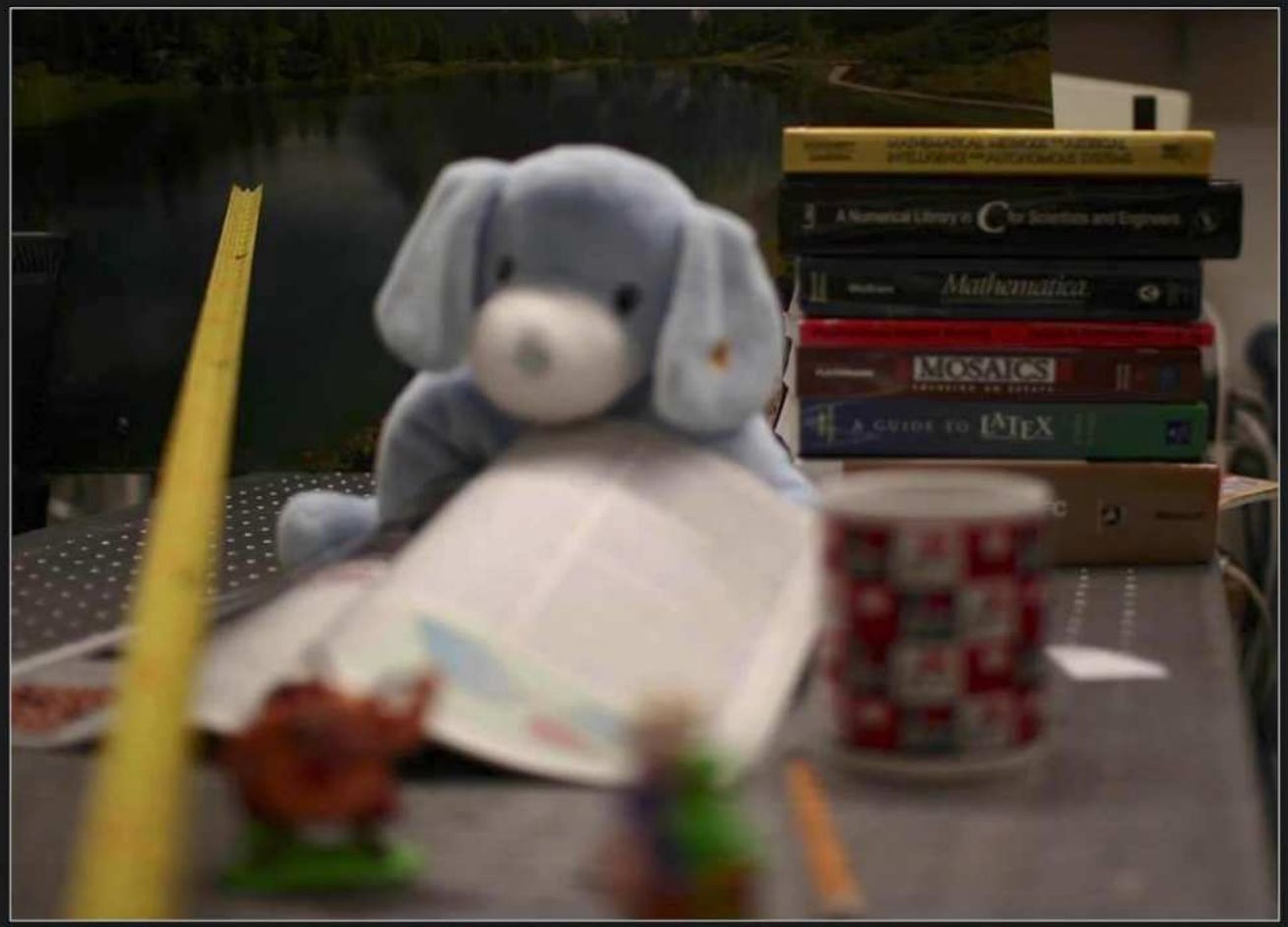
Defocus can be represented as product of Fourier transforms



(1D slice of Fourier transform)

Depth From Focus

Take images with different focus settings by moving the sensor



Depth From Focus

For each small image patch, find when it is best focused.

$s = 50.95$ $s = 51.10$ $s = 51.25$ $s = 51.40$ $s = 51.55$ $s = 51.70$ $s = 51.85$ (mm)

Obtain scene depth using gaussian lens law.

$$\frac{1}{f} = \frac{1}{s} + \frac{1}{o} \quad \Rightarrow \quad \boxed{o = \frac{sf}{s - f}}$$

Depth From Focus

For each small image patch, find when it is best focused.

$s = 50.95$ $s = 51.10$ $s = 51.25$ $s = 51.40$ $s = 51.55$ $s = 51.70$ $s = 51.85$ (mm)

Obtain scene depth using gaussian lens law.

$$\frac{1}{f} = \frac{1}{s} + \frac{1}{o} \quad \Rightarrow \quad \boxed{o = \frac{sf}{s - f}}$$

Ex: $s = 51.25\ mm$

$f = 50\ mm$

$o = 2.05\ m$

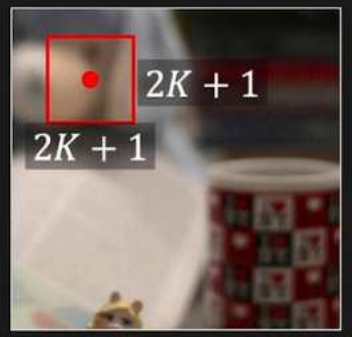
Focus Measure

Since defocus attenuates high frequencies, use a high-pass filter to measure amount of high frequency content within each small patch.

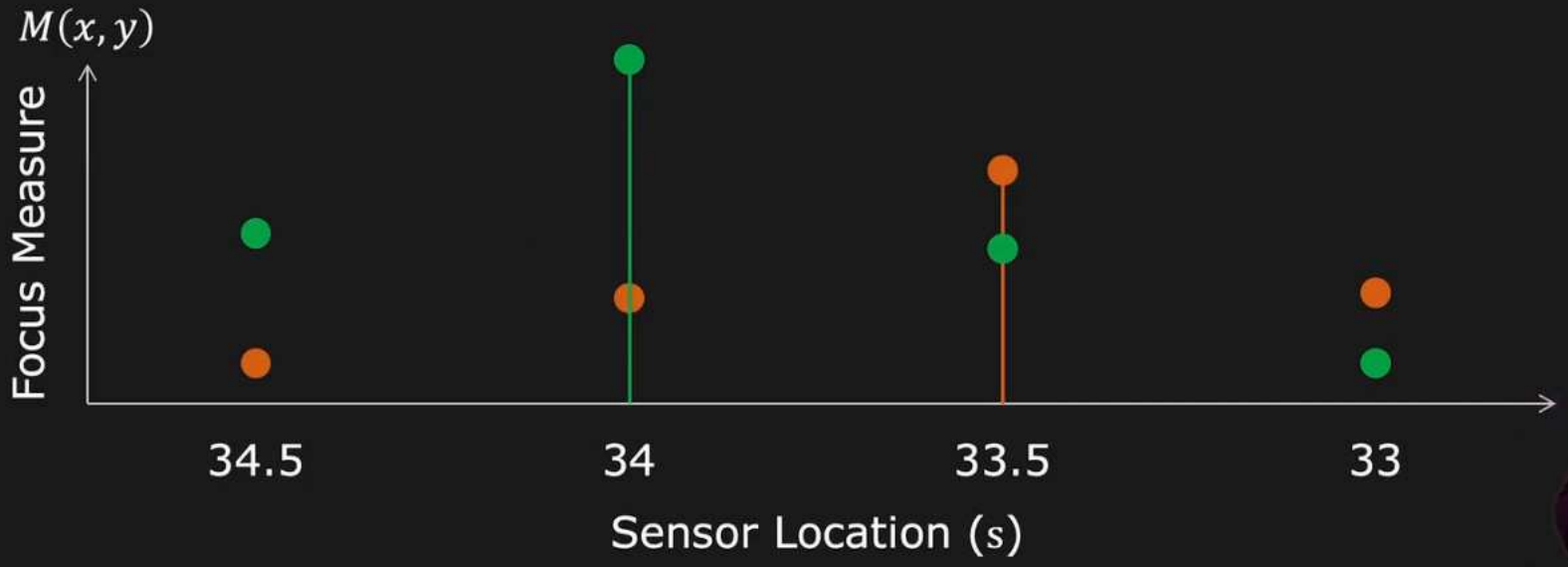
For example, use: $\nabla_M{}^2 f = \left|\frac{\partial^2 f}{\partial x^2}\right| + \left|\frac{\partial^2 f}{\partial y^2}\right|$ (Similar to Laplacian)

Focus Measure: Sum of the square of (modified) Laplacian responses within a small window.

$$M(x,y) = \sum_{i=x-K}^{x+K} \sum_{j=y-K}^{y+K} \nabla_M{}^2 f(i,j)$$



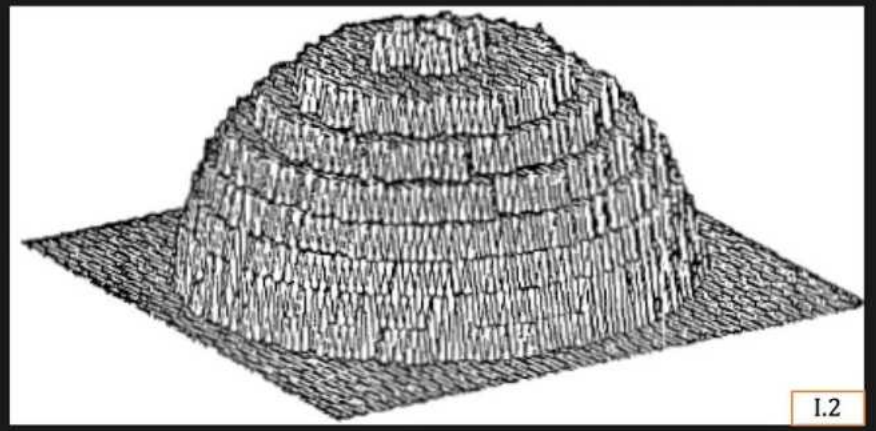
Depth from Focus



Depth from Focus

Depths can have only N values where N is the number of sensor locations.

Scene
(Metal ball with rough surface)

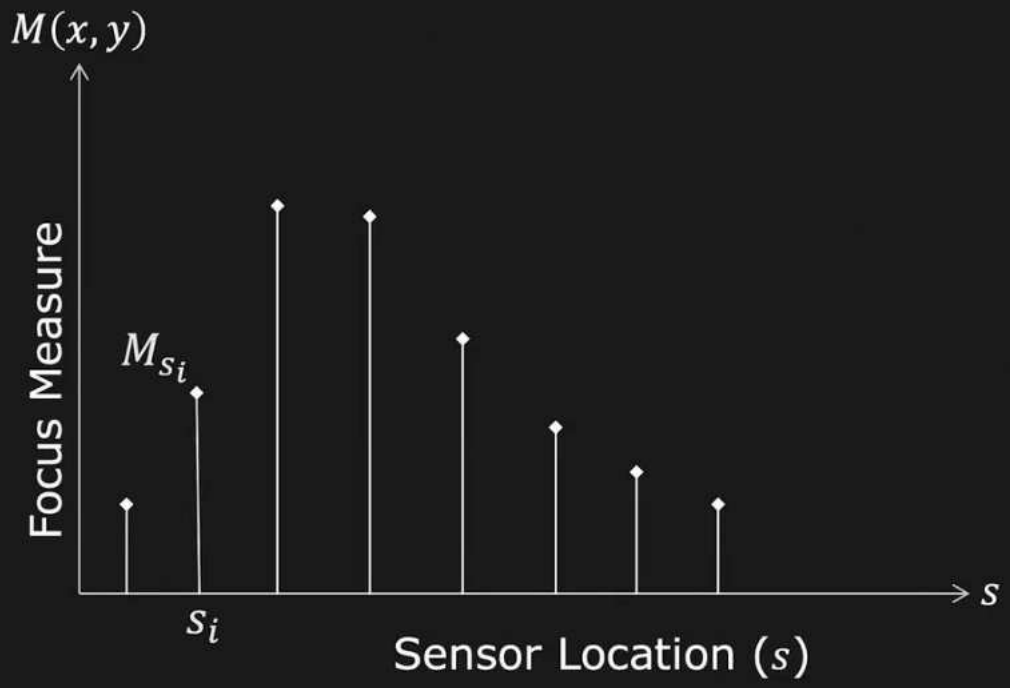


Shape obtained using Depth from Focus

Solution: Take images using many sensor locations. OR.

Gaussian Interpolation

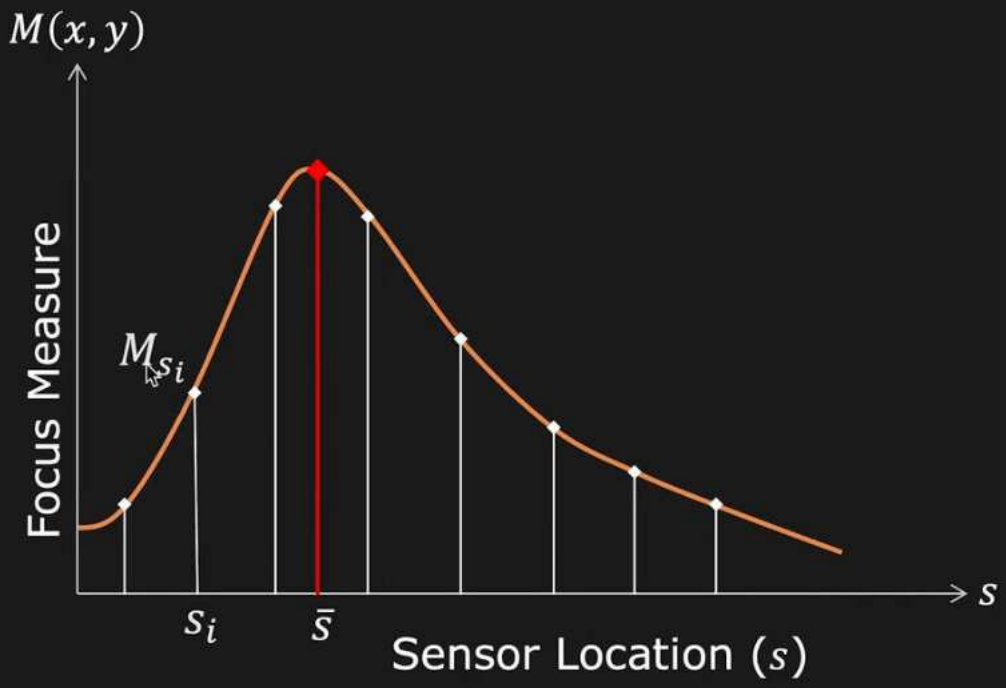
Peak of focus measure curve is a Gaussian-like function.



[Nayar 1994]

Gaussian Interpolation

Peak of focus measure curve is a Gaussian-like function.

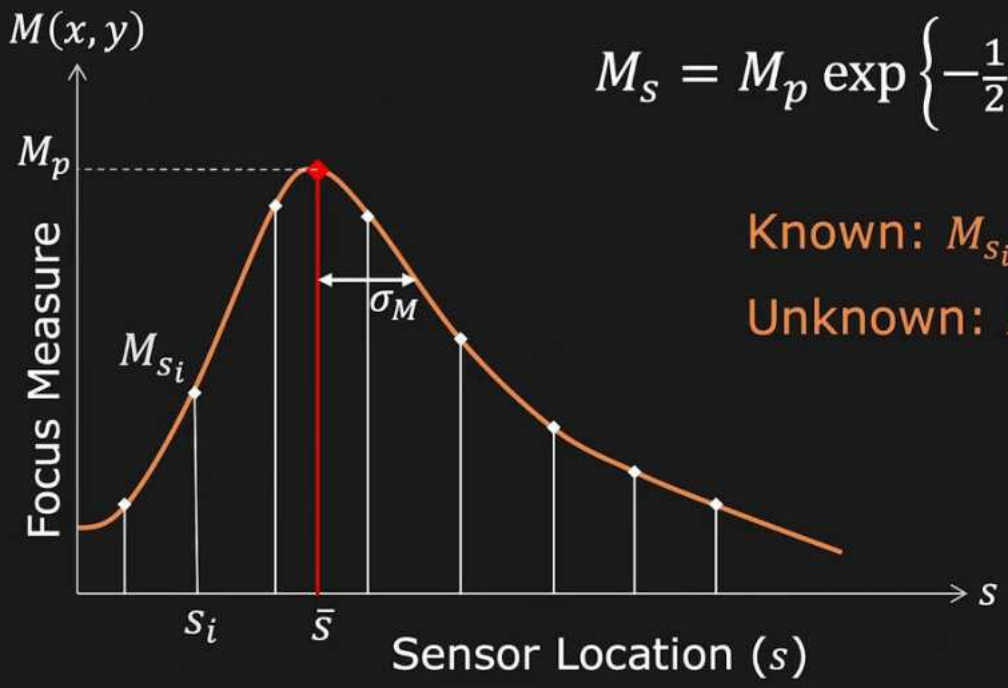


Mean of the Gaussian may be used as the sensor location corresponding to the "best focus."

[Nayar 1994]

Gaussian Interpolation

Peak of focus measure curve is a Gaussian-like function.



$$M_s = M_p \exp\left\{-\frac{1}{2}\left(\frac{s-\bar{s}}{\sigma_M}\right)^2\right\}$$

Known: M_{s_i}, s_i

Unknown: $M_p, \bar{s}, \sigma_M$

How many (M_{s_i}, s_i) samples do we need to estimate $\bar{s}$

Gaussian Interpolation

Linearize problem to solve for $(\bar{s}, M_p, \sigma_M)$.

Gaussian: $M_s = M_p \exp\left\{-\frac{1}{2}\left(\frac{s-\bar{s}}{\sigma_M}\right)^2\right\}$

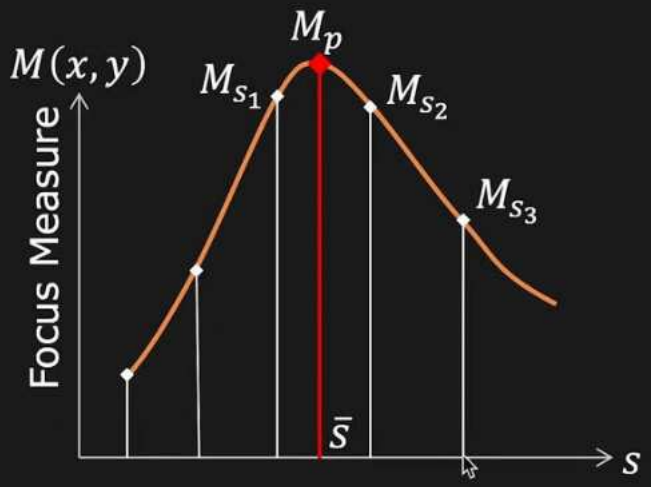
Taking the natural logarithm: $\ln M_s = \ln M_p - \frac{1}{2}\left(\frac{s-\bar{s}}{\sigma_M}\right)^2$

Three sensor positions:

$$\ln M_{s_1} = \ln M_p - \frac{1}{2}\left(\frac{s_1-\bar{s}}{\sigma_M}\right)^2$$

$$\ln M_{s_2} = \ln M_p - \frac{1}{2}\left(\frac{s_2-\bar{s}}{\sigma_M}\right)^2$$

$$\ln M_{s_3} = \ln M_p - \frac{1}{2}\left(\frac{s_3-\bar{s}}{\sigma_M}\right)^2$$



Where, $M_{s_1}, M_{s_2}, M_{s_3}$ are the three largest values.

Depth Estimation

Solving for $\bar{s}$:

$$\bar{s} = +\frac{\left(\ln M_{s_2} - \ln M_{s_3}\right)\left(s_2^2 - s_1^2\right)}{2(s_3 - s_2)\left\{\left(\ln M_{s_2} - \ln M_{s_1}\right) + \left(\ln M_{s_2} - \ln M_{s_3}\right)\right\}}$$

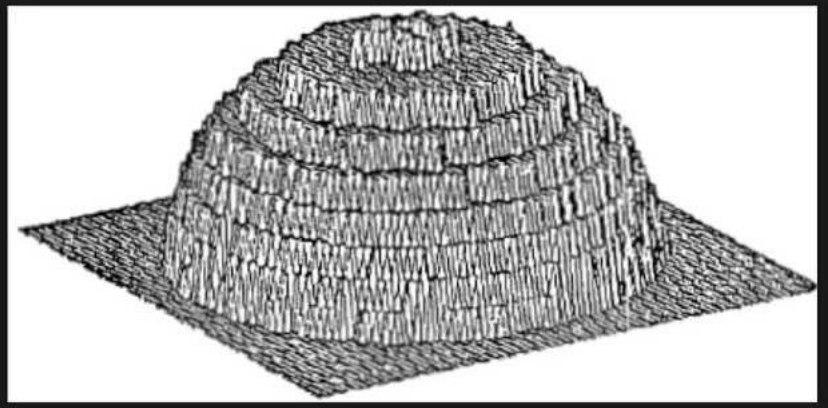
$$-\frac{\left(\ln M_{s_2} - \ln M_{s_1}\right)\left(s_2^2 - s_3^2\right)}{2(s_3 - s_2)\left\{\left(\ln M_{s_2} - \ln M_{s_1}\right) + \left(\ln M_{s_2} - \ln M_{s_3}\right)\right\}}$$

Obtain scene depth using Gaussian Lens Law:

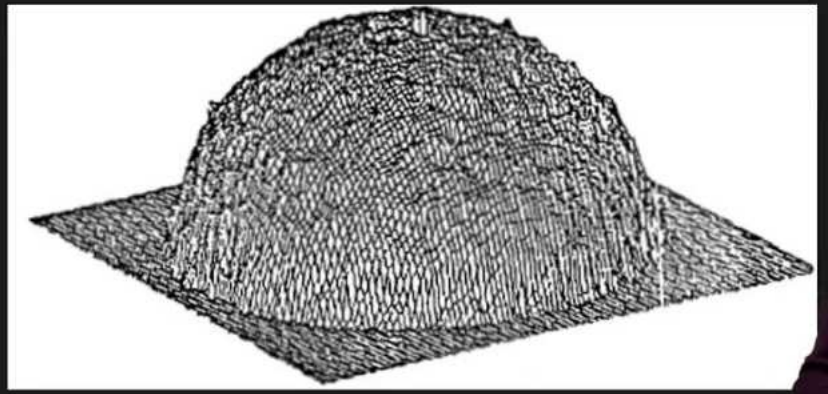
$$o = \frac{\bar{s} f}{\bar{s} - f}$$

Depth from Focus: Result

Scene
(Metal ball with
rough surface)

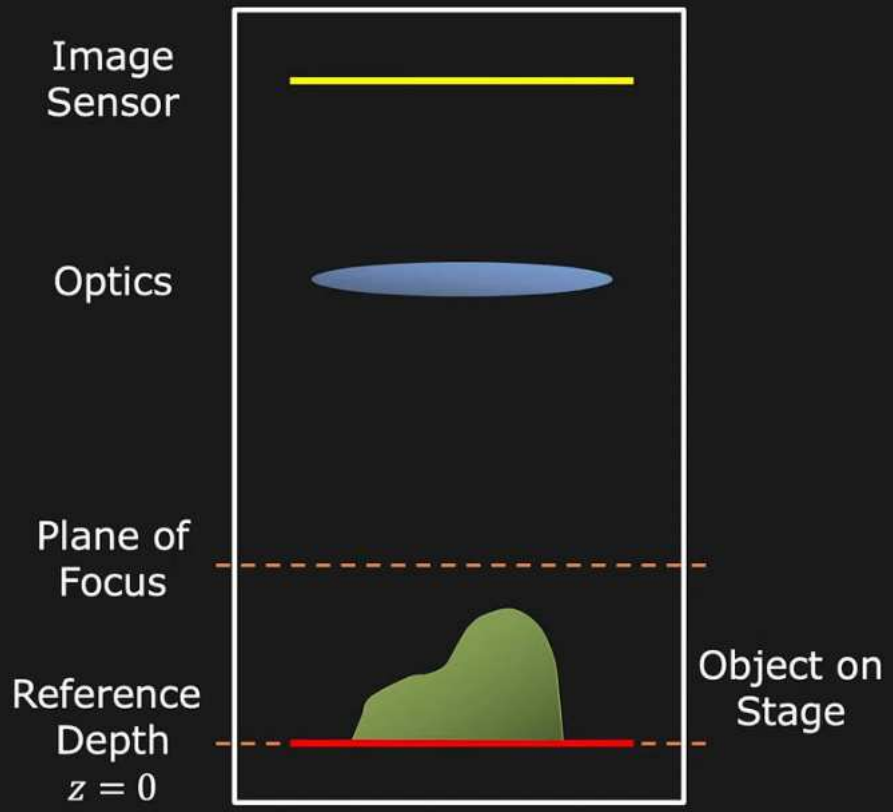


Depth without Gaussian Interpolation



Depth using Gaussian Interpolation

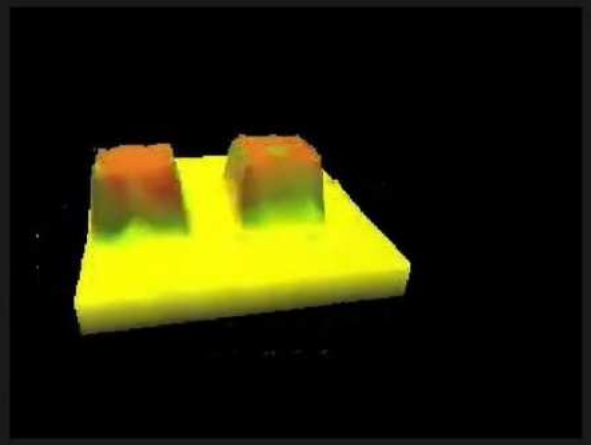
A Depth from Focus System



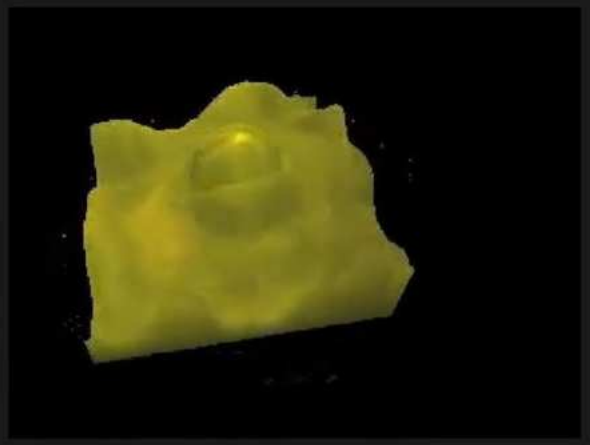
DFF in a Microscope

[Nayar 1994]

Depth from Focus System: Results



Structures on Silicon Wafer
(13 microns in height)



Leaf Stomata (air vent)
(30 microns in height)

[Nayar 1994]

Depth from Defocus

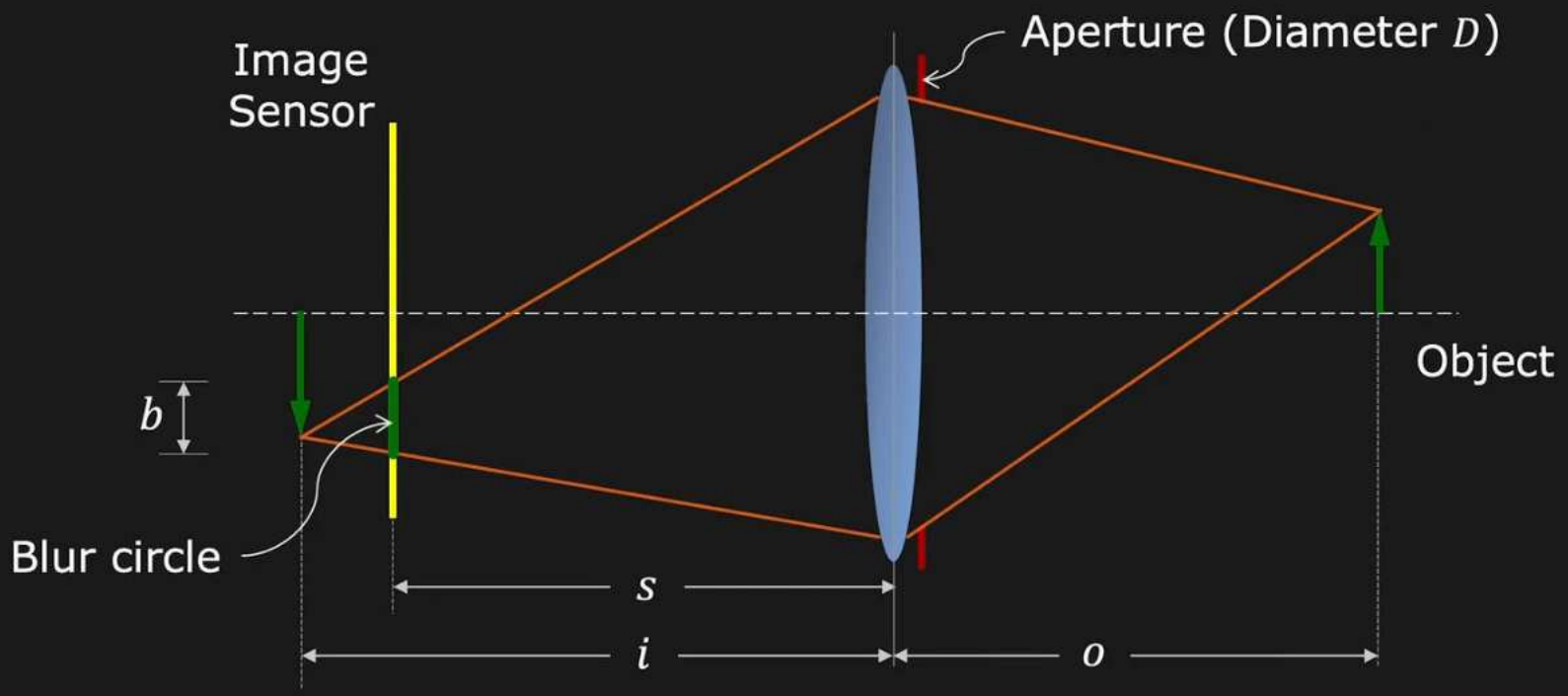
Given an image, depth of a scene point can be computed if we know how much it is defocused.

Captured image

Image Patches

PSFs

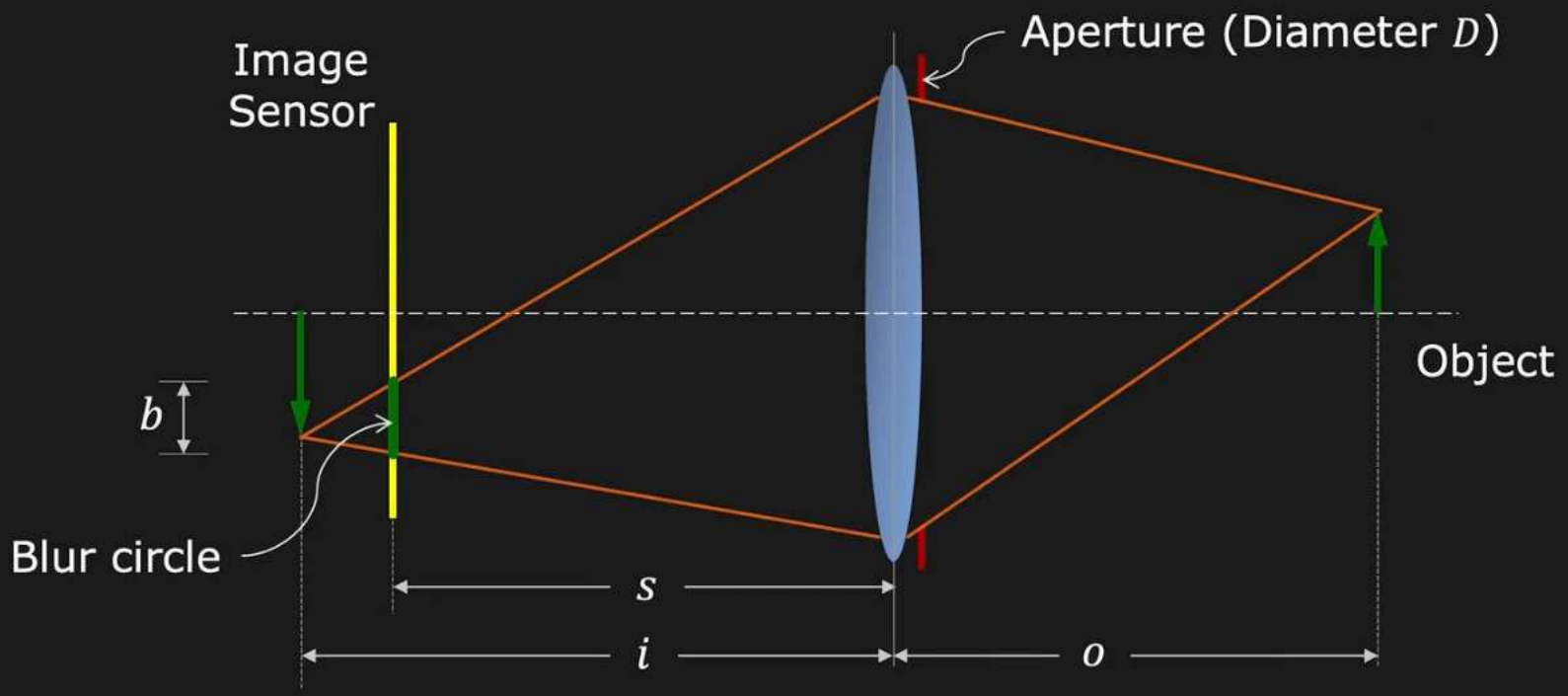
Depth From Defocus



We know that: $\frac{b}{D} = \frac{i - s}{i}$ and $o = \frac{if}{i - f}$

$$\Rightarrow i = \frac{Ds}{D - b} \qquad \Rightarrow o = \frac{sf}{s - f + b(f/D)}$$

Depth From Defocus



Given b, s, D and f, we get Object Distance:

$$o = \frac{sf}{s - f + b(f/D)}$$

f/D: F-Number

Depth from Defocus

Can we estimate blur size b from a single image?

$*$

$=$

Scene
$f(x, y)$

Defocus PSF
$h(x, y)$

Image Patch
$g(x, y)$

Impossible: One equation, two unknowns

Depth from Defocus

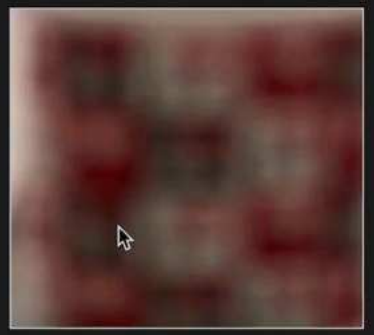
What if we have two images with different defocus?

$f(x,y)$

$*$

$h_{\sigma_1}(x,y)$

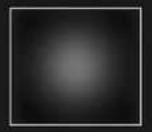
$=$



$g_1(x,y)$

$f(x,y)$

$*$



$h_{\sigma_2}(x,y)$

$=$

$g_2(x,y)$

Depth from Defocus

What if we have two images with different defocus?

$f(x,y)$ * 

$h_{\sigma_1}(x,y)$ = 

$g_1(x,y)$

$f(x,y)$ * 

$h_{\sigma_2}(x,y)$ = 

$g_2(x,y)$

Two equations, three unknowns

The Third Equation

If the two images were taken with two known apertures, then their blur sizes are related.

Blur circles:

$$b_1 = D_1 \left|1 - \frac{s}{i}\right| \qquad b_2 = D_2 \left|1 - \frac{s}{i}\right|$$

$$\sigma_1 = b_1/2 \qquad \sigma_2 = b_2/2$$

$$\frac{\sigma_1}{\sigma_2} = \frac{D_1}{D_2}$$

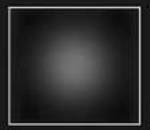
Depth from Defocus

For each image patch we have:

Three unknowns

$f(x,y)$

σ_1



σ_2

Three equations

$$g_1(x,y) = f(x,y) * h_{\sigma_1}(x,y)$$

$$g_2(x,y) = f(x,y) * h_{\sigma_2}(x,y)$$

$$\sigma_1/\sigma_2 = D_1/D_2$$

In Fourier Domain

$$G_1(u,v) = F(u,v) \times H_{\sigma_1}(u,v)$$

$$G_2(u,v) = F(u,v) \times H_{\sigma_2}(u,v)$$

$$\sigma_1/\sigma_2 = D_1/D_2$$

A Naïve DFD Algorithm

Cancel out $F(u,v)$:

$$\frac{G_1(u,v)}{G_2(u,v)} = \frac{F(u,v)\times H_{\sigma_1}(u,v)}{F(u,v)\times H_{\sigma_2}(u,v)} = \frac{H_{\sigma_1}(u,v)}{H_{\sigma_2}(u,v)}$$

Substitute for $H_{\sigma_1}(u,v)$ and $H_{\sigma_2}(u,v)$:

$$\frac{G_1(u,v)}{G_2(u,v)} = \frac{\exp(-2\pi^2(u^2+v^2)\sigma_1^2)}{\exp(-2\pi^2(u^2+v^2)\sigma_2^2)}$$

Taking the natural logarithm on both sides:

$$\sigma_1^2 - \sigma_2^2 = \frac{\ln G_2(u,v) - \ln G_1(u,v)}{2\pi^2(u^2+v^2)}$$

$$\sigma_1/\sigma_2 = D_1/D_2$$

Solve the above to get σ_1 and σ_2. Use either one to obtain object distance (depth).

[Pentland 1987]

Reconstruction-Based Depth from Defocus

Find $(\sigma_1, \sigma_2, f(x,y))$ that minimizes Reconstruction Error:

$$E(\sigma_1, \sigma_2, f) = \|g_1 - (h_{\sigma_1} * f)\|^2 + \|g_2 - (h_{\sigma_2} * f)\|^2$$

We know that $\sigma_2 = \sigma_1 D_2/D_1$

Rewrite E as a 2-variable function:

$$E(\sigma_1, f) = \|g_1 - (h_{\sigma_1} * f)\|^2 + \|g_2 - (h_{(\sigma_1 D_2/D_1)} * f)\|^2$$

[Favaro 2003]

Computing Depth From Defocus

Find (σ_1, f) using: $\frac{\partial E}{\partial \sigma_1} = 0$ and $\frac{\partial E}{\partial f} = 0$

Using σ_1 compute size of blur circle: $b_1 = 2\sigma_1$

Object distance (depth):

$$o = \frac{s_1 f}{s_1 - f + b_1(f/D)}$$

Capturing Defocused Images

Method 1: Change Aperture

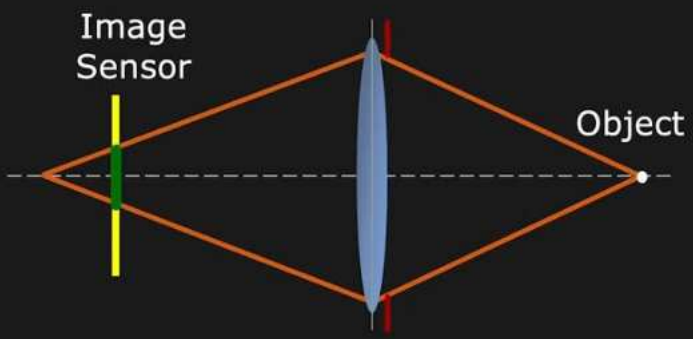


Image 1

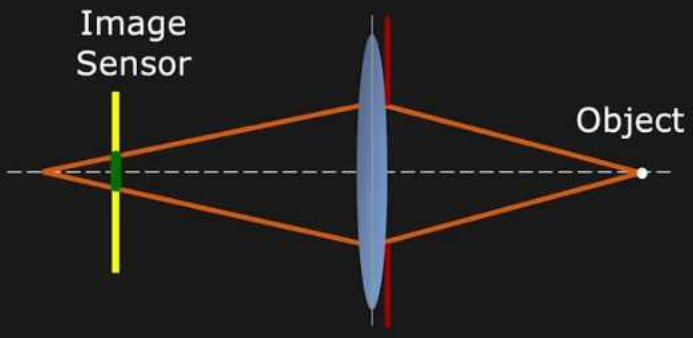


Image 2

Method 2: Move Sensor

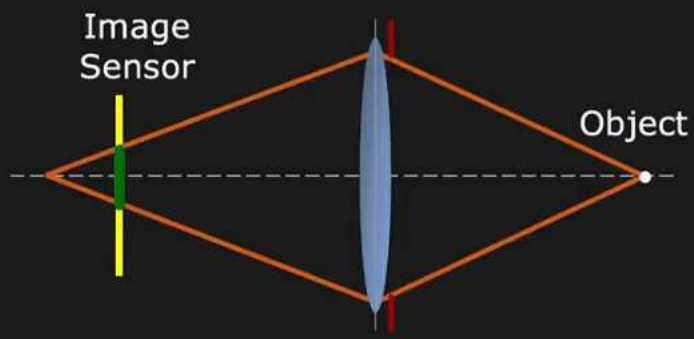


Image 1

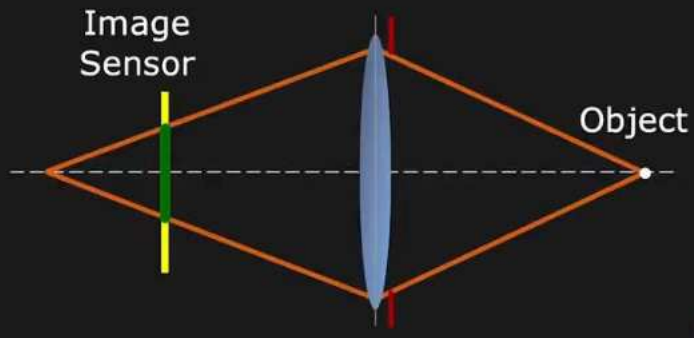
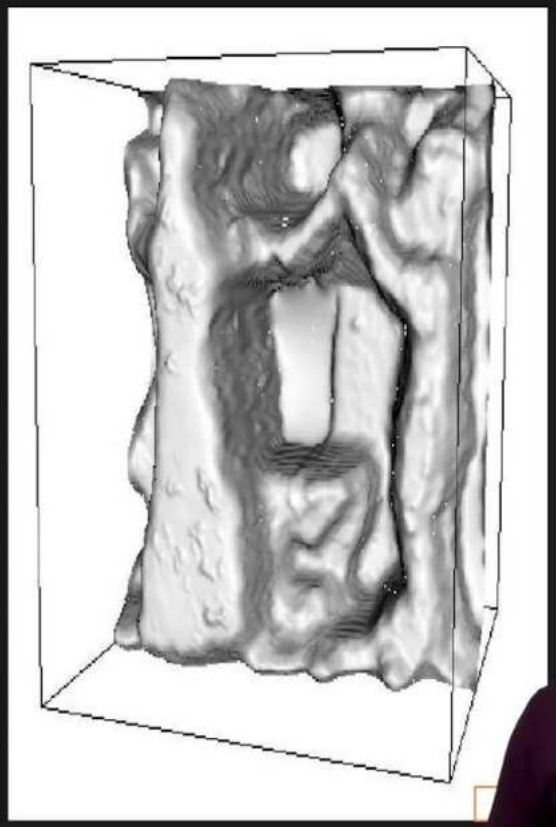


Image 2

Depth from Defocus: Result

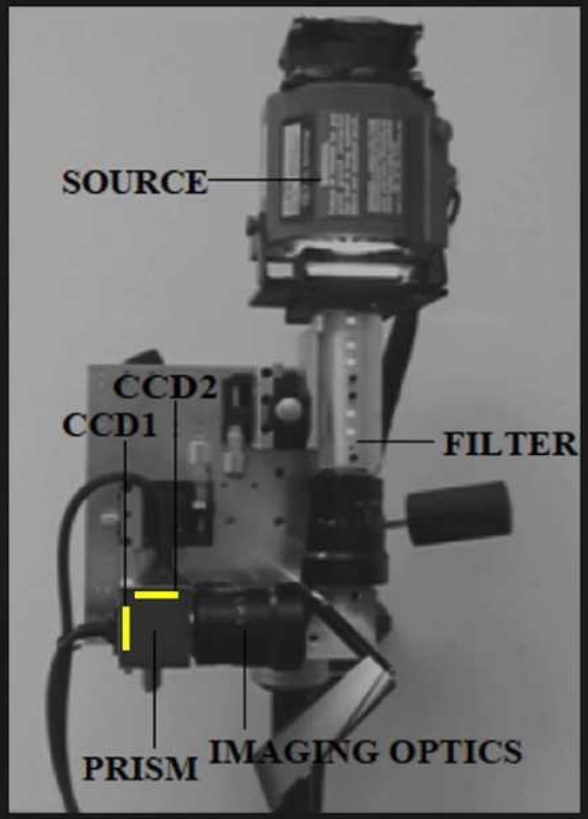
Far Focused

Near Focused



Estimated 3D shap

Depth from Defocus System



scene

Uses a fine illumination pattern to ensure surface texture

[Nayar 1996]

Depth from Defocus System

[Nayar 1996]

Depth from Defocus System

[Nayar 1996]

Active Illumination Methods

Shree K. Nayar

Columbia University

Topic: Active Illumination Methods, Module: Reconstruction I

First Principles of Computer Vision


© 2020 Shree K. Nayar

Photometric Stereo Systems

Shree K. Nayar

Columbia University

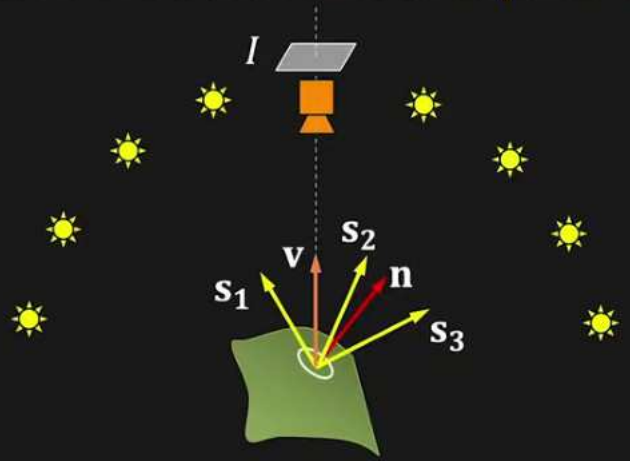
Topic: Active Illumination Methods, Module: Reconstruction I

First Principles of Computer Vision

Photometric Sampling

Given: Multiple images of object with known reflectance model with unknown parameters under different known sources.

Find: Surface normals and reflectance parameters.



[Nayar 1989]

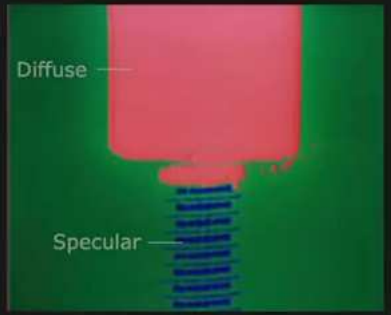
Photometric Sampling

[Nayar 1989]

Recovering Shape and Reflectance

Object

Surface Normals



Reflectance

[Nayar 1989]

Light Stage

[Wenger 2005]

Relighting

[Wenger 2005]

Real-Time Shape and Reflectance

Surface Normals and Reflectance

[Wenger 2005]

Real-Time Shape and Reflectance

Surface Normals and Reflectance

[Wenger 2005]

Real-Time Shape and Reflectance

Surface Normals and Reflectance

Relighting

[Wenger 2005]

Structured Light Range Finding

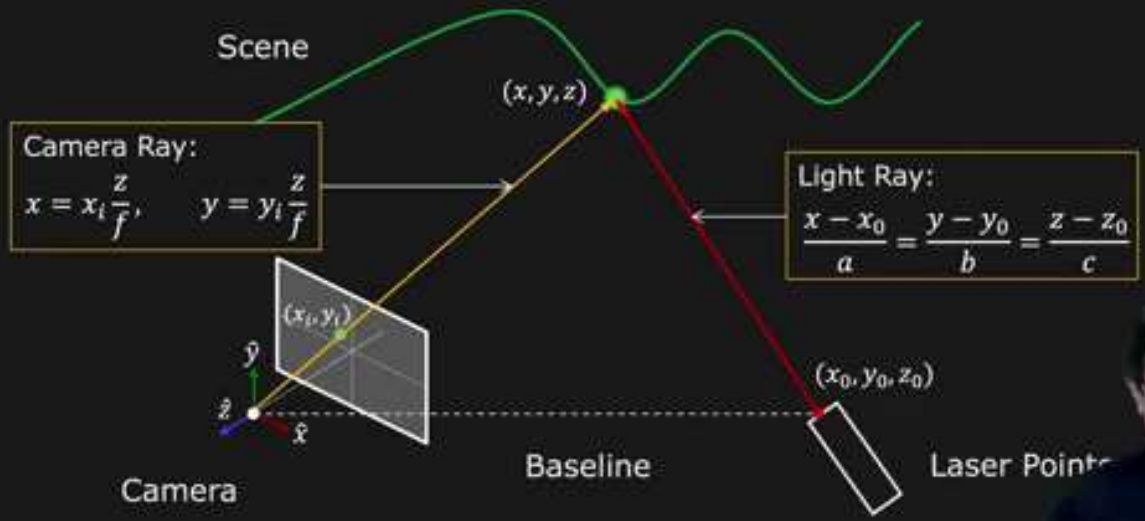
Shree K. Nayar

Columbia University

Topic: Active Illumination Methods, Module: Reconstruction I

First Principles of Computer Vision

Point Based Range Finding



Scene Point (x, y, z)= Camera Ray ∩ Light Ray

Detecting the Illuminated Point

Background Image (I_B)

Captured Image with Pointer (I_P)

$(I_P - I_B)$

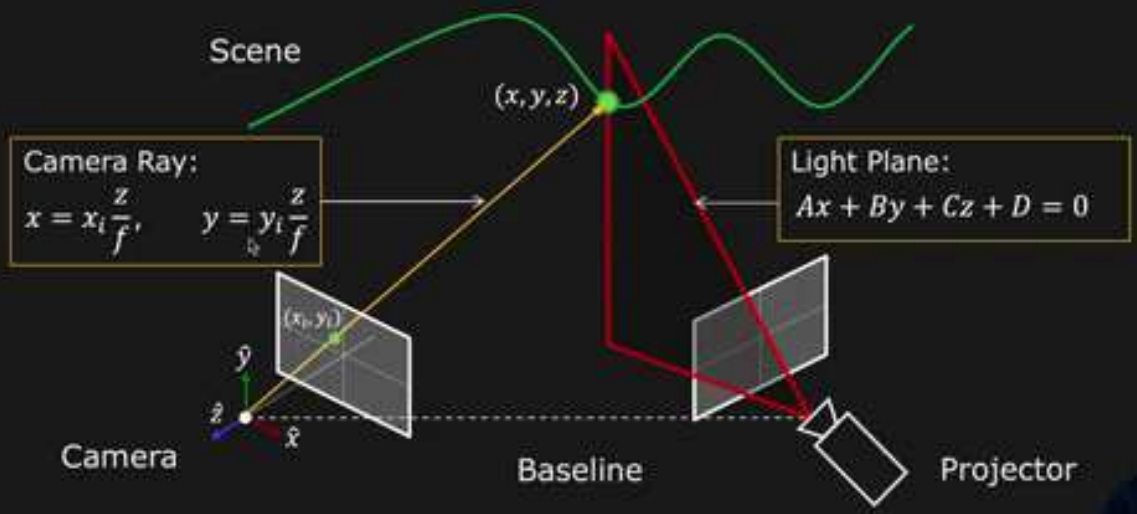
How Many Images?

One image per pixel

For 640x480 image: >300,000 images!

At 30 frames per second (fps): ~3 hours!

Light Stripe Based Range Finding



Scene Point (x, y, z)= Camera Ray ∩ Light Plane

$$z = \frac{-Df}{Ax_i + By_i + Cf}$$

How Many Images?

What camera sees

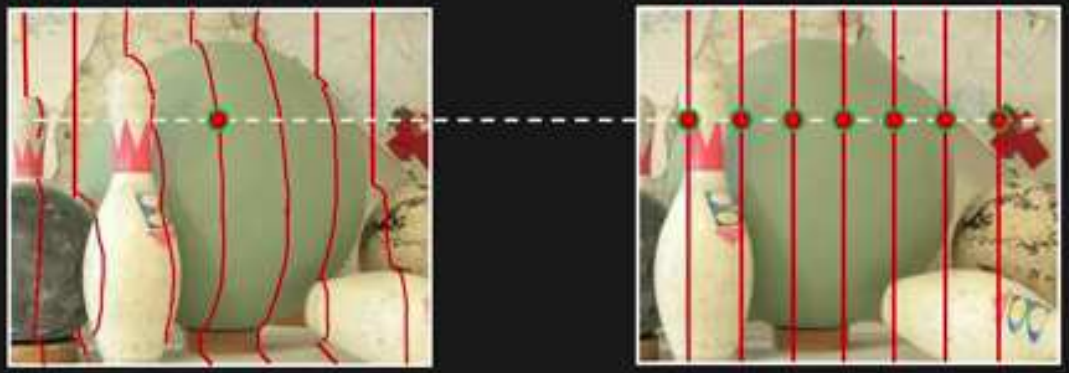
What projector "sees"

One image per column

For 640x480 image: still 640 images!

At 30 fps: ~21s

Can we do Multiple Stripes at Once?



What camera sees

What projector "sees"

Ambiguous!

Binary Coded Structured Light

Bit 1	0	0	0	1	1	1	1
Bit 2	0	1	1	0	0	1	1
Bit 3	1	0	1	0	1	0	1
(Binary)	(001)	(010)	(011)	(100)	(101)	(110)	(111)
Stripe Numbers	1	2	3	4	5	6	7

[Posdamer 1981]

Binary Coded Structured Light

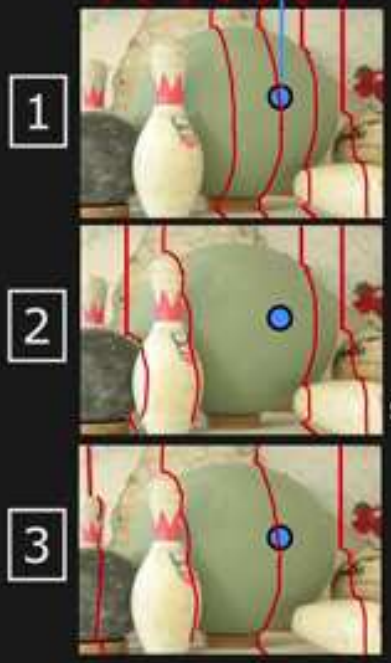


Image		Projection Pattern						
1	Bit 1	0	0	0	1	1	1	1
2	Bit 2	0	1	1	0	0	1	1
3	Bit 3	1	0	1	0	1	0	1
(Binary) Stripe Numbers		(001) 1	(010) 2	(011) 3	(100) 4	(101) 5	(110) 6	(111) 7

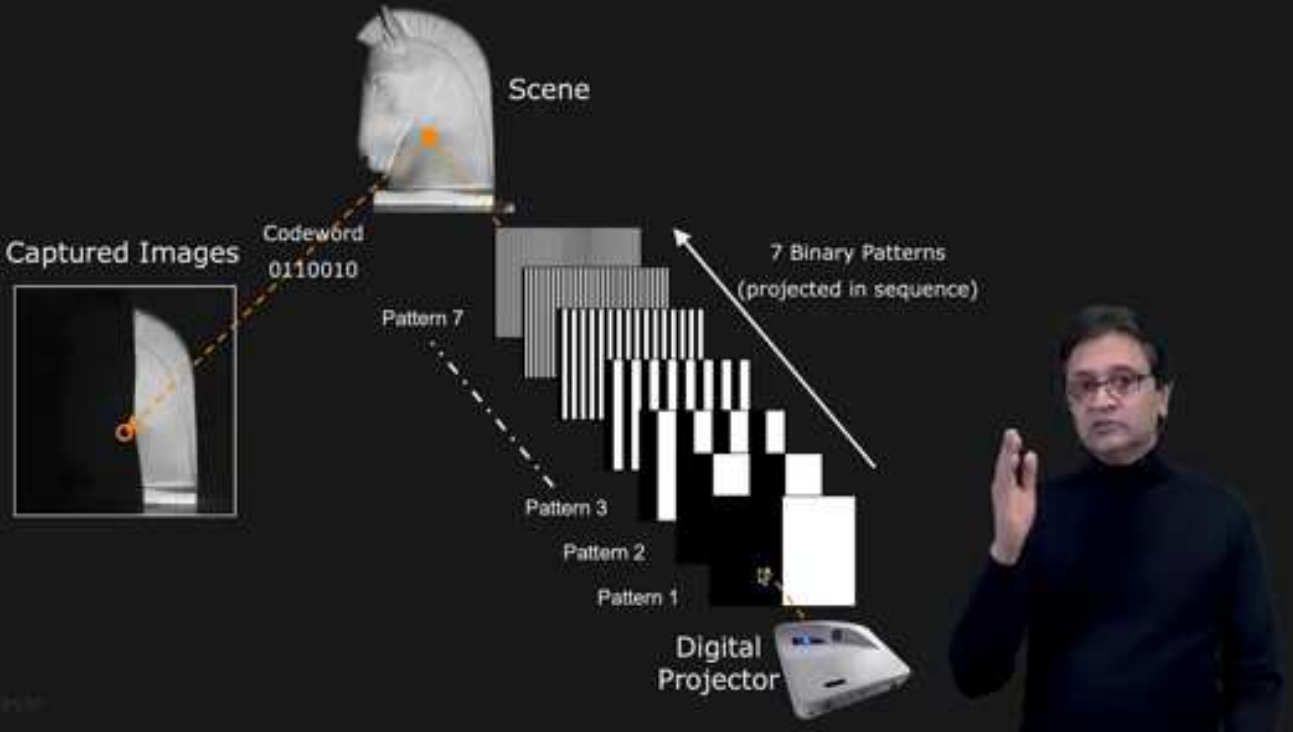
7 stripes in 3 images!

In general, we can do
$2^n - 1$ stripes in n Images

Note: (000) is not an option. Hence, the

[Posdamer 1981]

Binary Coded Structured Light: Example



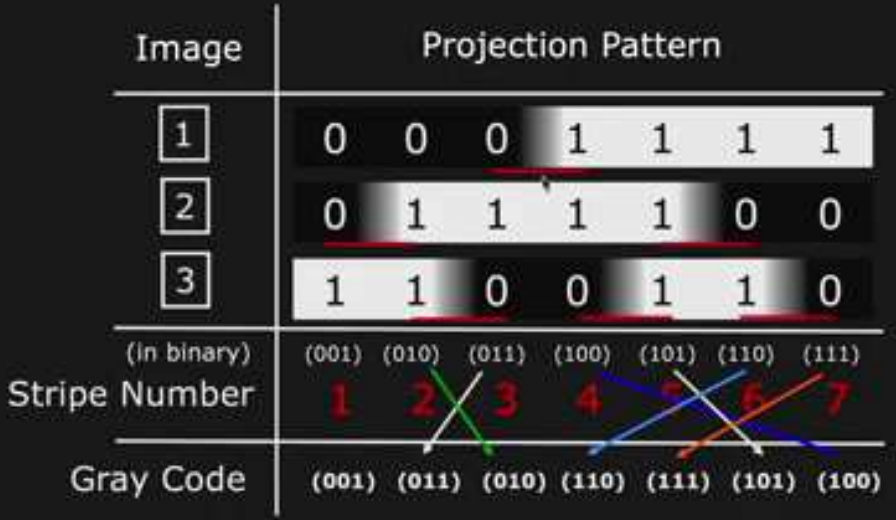
Binary Coded Structured Light: Example

Scene

Captured Images

3D Reconstruction

Gray Coding to Reduce Errors



We can have a maximum of 6 errors!

[Inokuchi 1984]

k-ary Methods

Coding	Base	Values
Binary	2	0, 1 (Off, On)
Ternary	3	0, 1, 2 (R, G, B), (Off, ½On, On)
k-ary	k	0, 1, 2,... k-1

Color Coding with R, G, B

[Caspi 1998]

Color Coding with R, G, B

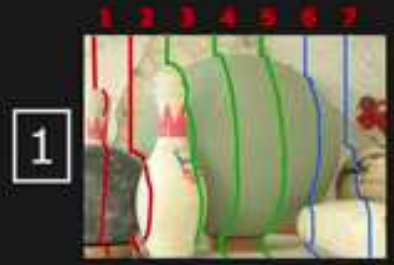


Image	Projection Pattern						
1	R	R	G	G	G	B	B
	G	B	R	G	B	R	G
(Ternary)	(01)	(02)	(10)	(11)	(12)	(20)	(21)
Stripe Numbers	1	2	3	4	5	6	7

[Caspi 1998]

Color Coding with R, G, B

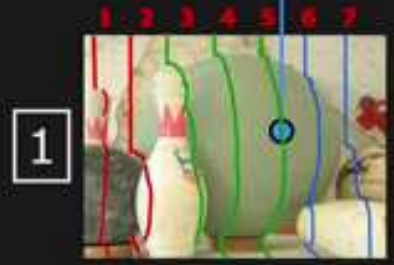


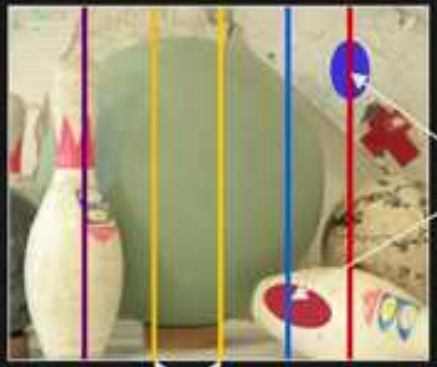
Image	Projection Pattern						
1	R	R	G	G	G	B	B
2	G	B	R	G	B	R	G
(Ternary) Stripe Numbers	(01) 1	(02) 2	(10) 3	(11) 4	(12) 5	(20) 6	(21) 7

Only 2 images needed!

With k levels, we get k^n stripes in n images. When one of the levels is 0 (OFF), we get k^n-1 stripes.

[Caspi 1998]

Color Structured Light: Remarks



- Similar colors hard to distinguish
- Some colors not reflected by some scene point

Phase Shifting Method

Shree K. Nayar

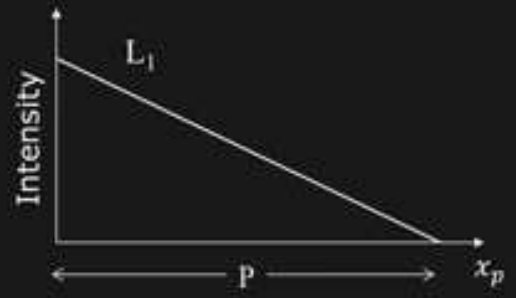
Columbia University

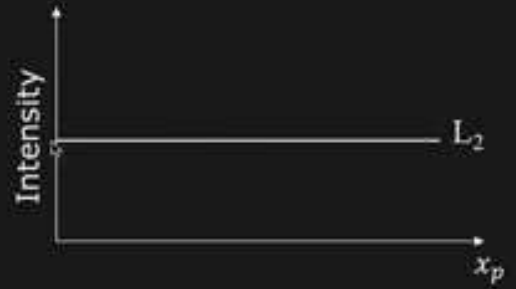
Topic: Active Illumination Methods, Module: Reconstruction I

First Principles of Computer Vision

Intensity Ratio Method

Projection Pattern



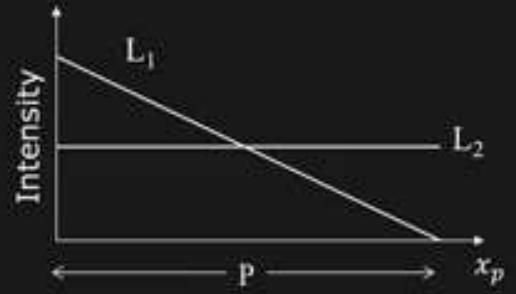


Captured Image

[Carrihill 1985]

Finding Correspondence

Projection Pattern



Scene

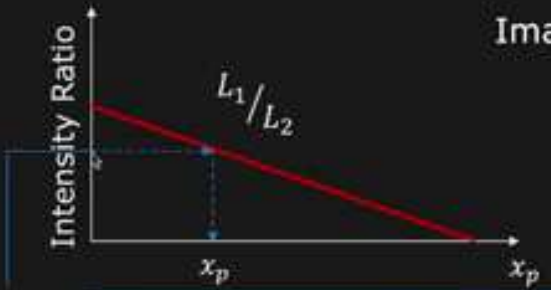
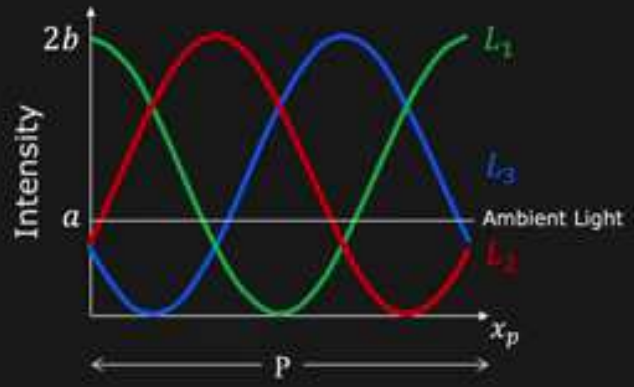


Image Intensities:

$$\frac{I_1 = \rho \cdot L_1}{I_2 = \rho \cdot L_2}$$

Phase Shift Method

Projected Pattern



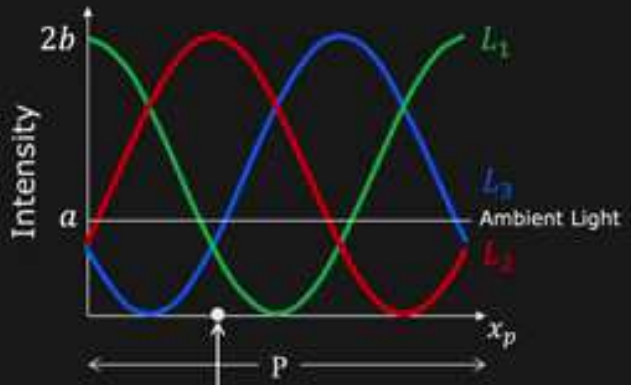
Captured Image

$$L_3(x_p) = a + b + b\cos\left(\frac{2\pi x_p}{P} + \frac{2\pi}{3}\right)$$

$$I_3(x_c, y_c) = \rho a + \rho b + \rho b\cos\left(\frac{2\pi x_p}{P} + \frac{2\pi}{3}\right)$$

Finding Correspondence

Projected Pattern



Captured Image

$$x_p = \frac{P}{2\pi}\tan^{-1}\left(\sqrt{3}\frac{I_2 - I_3}{2I_1 - I_2 - I_3}\right)$$

Structured Light for Depth Recovery

Method	Number of Images
Point based Structured Light	NM
Line based Structured Light	N
Binary Coded Structured Light	$\lceil \log_2(N+1) \rceil$
k-ary (Color) Coded Structured Light	$\lceil \log_k(N+1) \rceil$
Intensity Ratio Method	2
Phase Shifting Method	3

$[N, M]$: Camera Image Size, k: Number of colors, $\lceil x \rceil$: Smallest integer

Structured Light for Depth Recovery

Method	Number of Images
Point based Structured Light	NM
Line based Structured Light	N
Binary Coded Structured Light	$\lceil \log_2(N+1) \rceil$
k-ary (Color) Coded Structured Light	$\lceil \log_k(N+1) \rceil$
Intensity Ratio Method	2
Phase Shifting Method	3

$[N, M]$: Camera Image Size, k: Number of colors, $\lceil x \rceil$: Smallest integer

Structured Light Systems

Shree K. Nayar

Columbia University

Topic: Active Illumination Methods, Module: Reconstruction I

First Principles of Computer Vision

3D Visual Inspection System

Courtesy of Omron Corporation

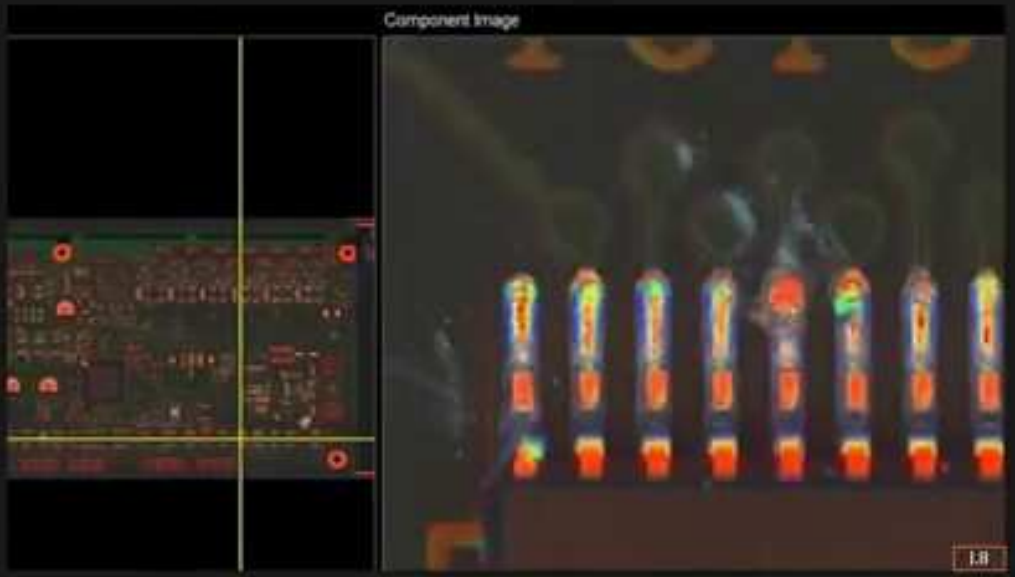
3D Visual Inspection System

Courtesy of Omron Corporation

3D Visual Inspection System

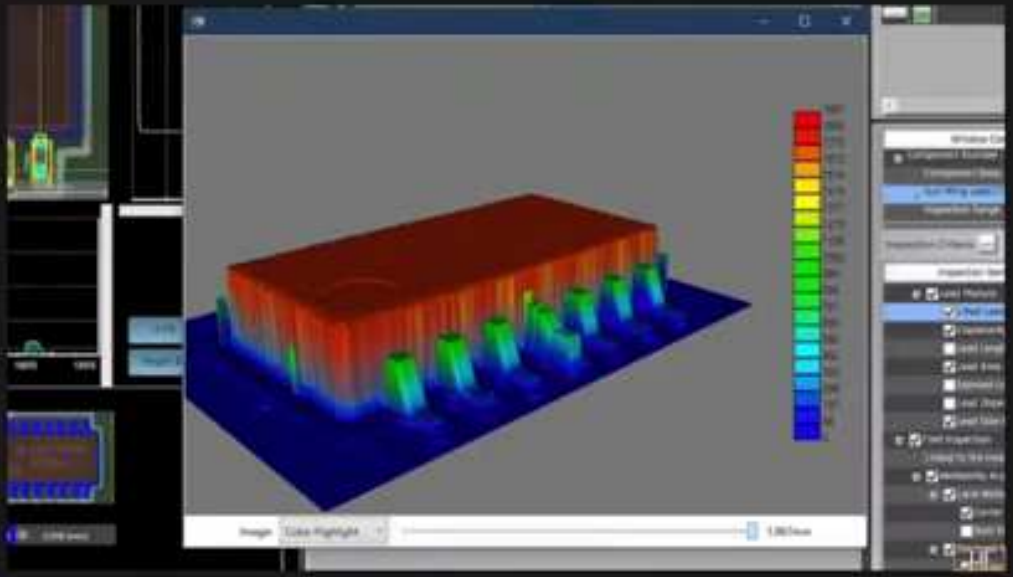
Courtesy of Omron Corporation

3D Visual Inspection System



Courtesy of Omron Corporation

3D Visual Inspection System



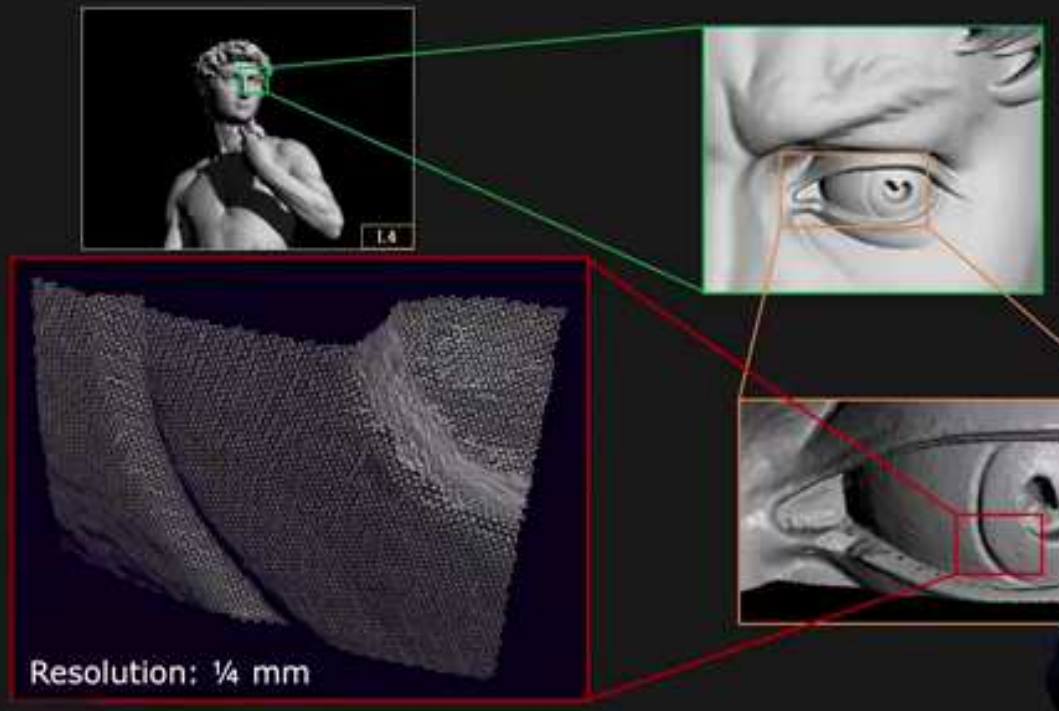
Courtesy of Omron Corporation

Digital Michelangelo Project

480 individually aimed scans
2 billion polygons
7,000 color images
32 gigabytes of data
30 nights of scanning
22 people

[Levoy 2000]

Virtual "David"



Great Buddha Project

Great Buddha, Nara

Digital Model

[Ikeuchi 2007]

Performance Capture with Light Stage

[Guo 2019]

Performance Capture with Light Stage

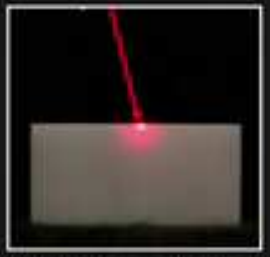
[Guo 2019]

Performance Capture with Light Stage

[Guo 2019]

Unsolved Problems

Optically uncooperative materials



Scattering surface
(marble)

Scattering environment
(underwater)

Specular surface
(mirror-like)

Transparent surface
(glass)

Fuzzy
(hair)

Time of Flight Method

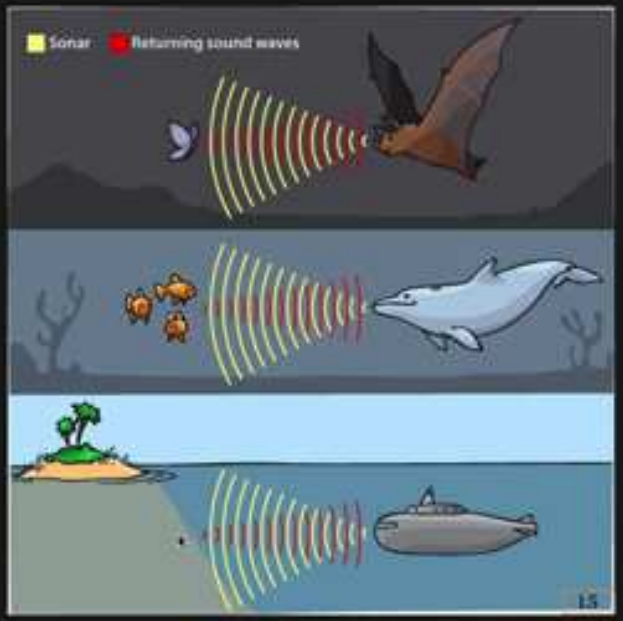
Shree K. Nayar

Columbia University

Topic: Active Illumination Methods, Module: Reconstruction I

First Principles of Computer Vision

Time of Flight

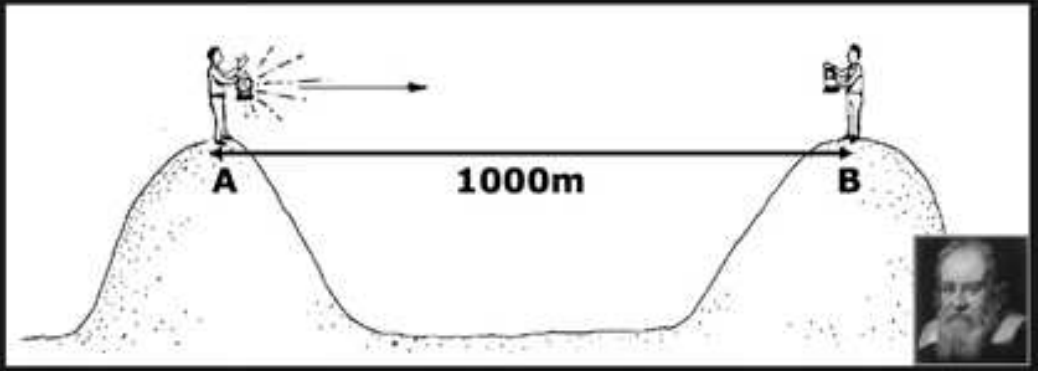


Echolocation

Measuring Time of Flight of Sound Waves

Time of Flight of Light

Does light travel or exist instantaneously?

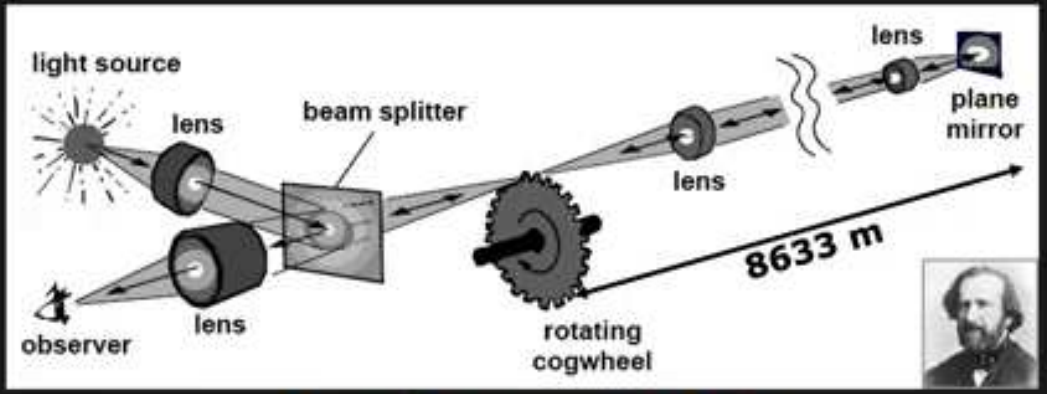


Galileo's experiment (early 1600s)

Time for light to travel back and forth: $t_{A\to B\to A} = \frac{2000}{3e^8} \cong 6.6\,\mu s$

Much faster than human reflexes for experiment to have succeed

Time of Flight of Light

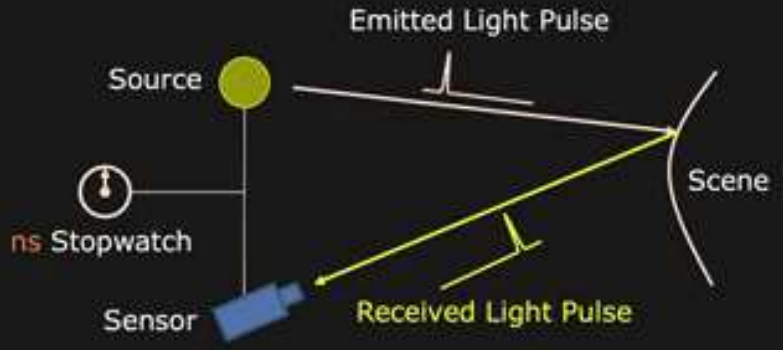


Fizeau's experiment (1849)

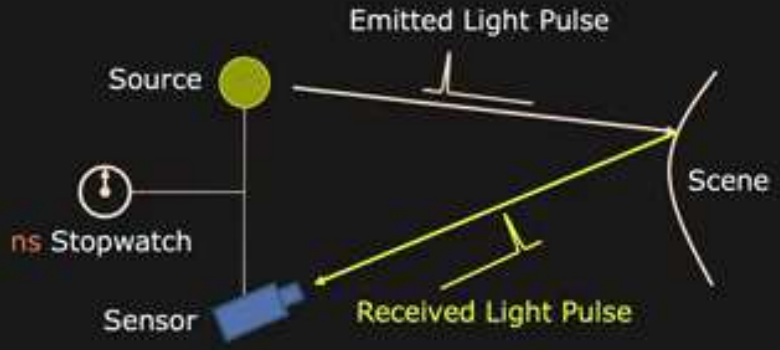
$$c_{computed} = 3.153 \times 10^8 m/s$$

$$c_{actual} = 2.998 \times 10^8 m/s$$

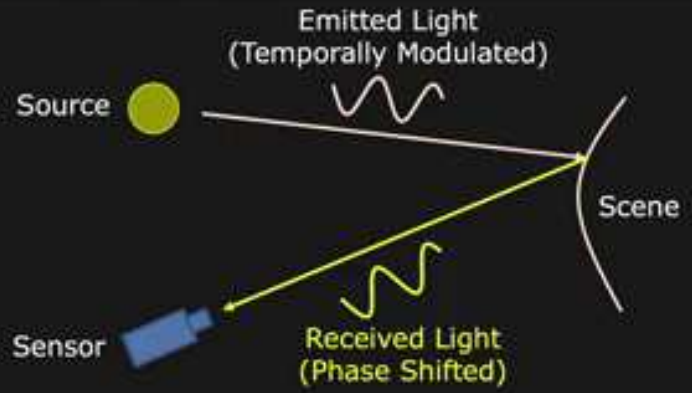
Time of Flight: Pulse Modulation

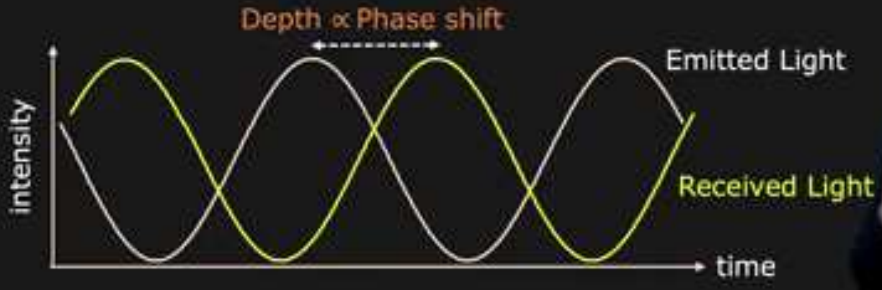


Time of Flight: Pulse Modulation

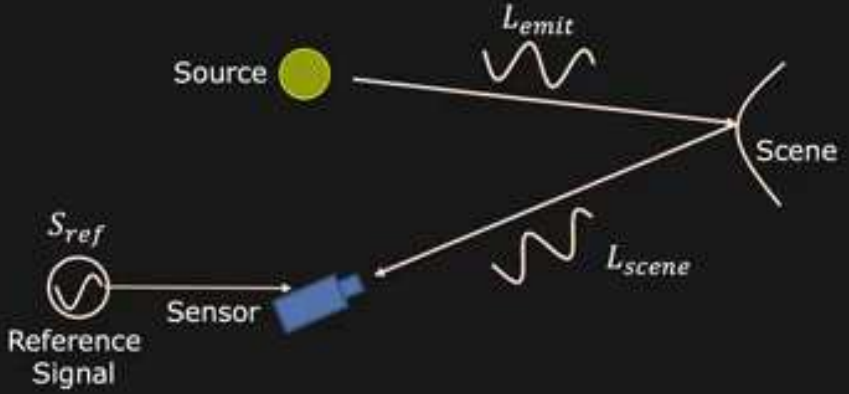


Time of Flight: Continuous Modulation





Phase Measurement By Correlation



Emitted Light: $L_{emit} = \cos(\omega t)$

Received Light: $L_{scene} = O + A\cos(\omega t - \varphi)$

Reference Signal: $S_{ref} = \cos(\omega t - \delta)$

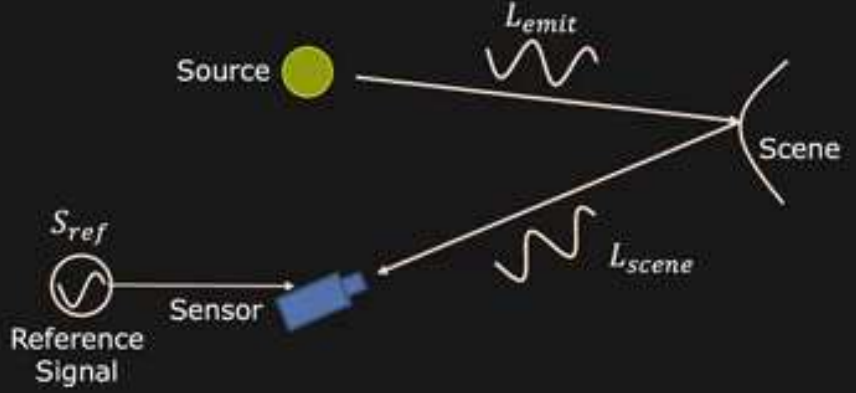
O: ambient light

φ: phase shift

A: albedo

δ: reference p'

Phase Measurement By Correlation



Measured Intensity: $I(\delta_i) = \int L_{scene}(t) \times S_{ref}(t, \delta_i) dt$

Simplifying: $I(\delta_i) = P + Q\cos(\delta_i - \varphi)$

Computing Depth From Phase

$$I(\delta_i) = P + Q\cos(\delta_i - \varphi)$$

Three measurements $I(\delta_1)$, $I(\delta_2)$, $I(\delta_3)$ are sufficient for computing phase φ.

Depth of scene point:

$$d = c\frac{\varphi}{4\pi f}$$

c: speed of light

φ: phase shift

$f = {}^{\omega}/_{2\pi}$: modulation frequency

Example: If $\varphi = \pi, f = 30MHz,$ then $d = 2.5m.$

Time-of-Flight Result: Example

Google Self-Driving Car Project

Time of Flight: 3D Camera

PMD Technologies