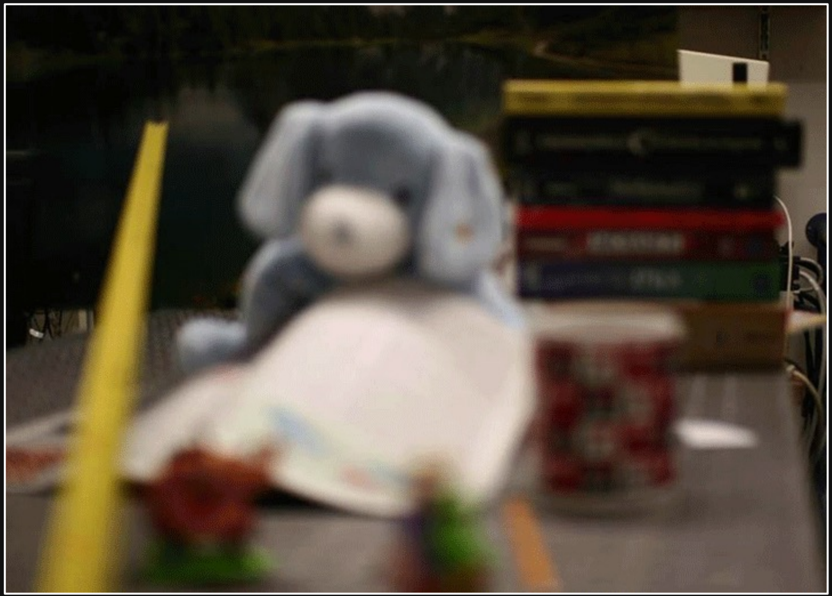
# CS/ECE 766 : Computer Vision

University of Wisconsin-Madison

## Depth from Focus

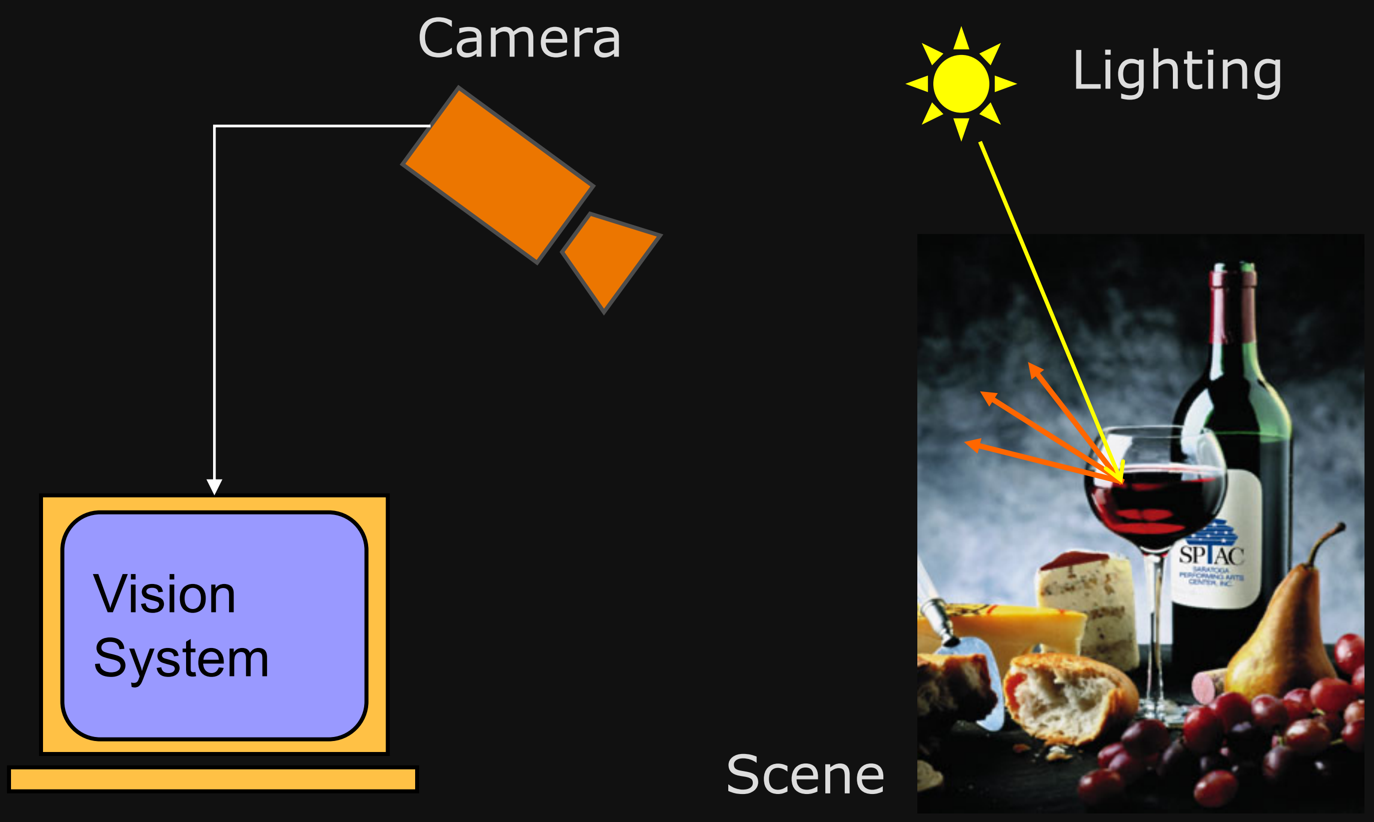


# CS/ECE 766 : Computer Vision

University of Wisconsin-Madison

# Review: Photometric Stereo

# From 2D to 3D




We need to understand the relation between lighting, surface reflectance and image intensity.

# Photometric Stereo

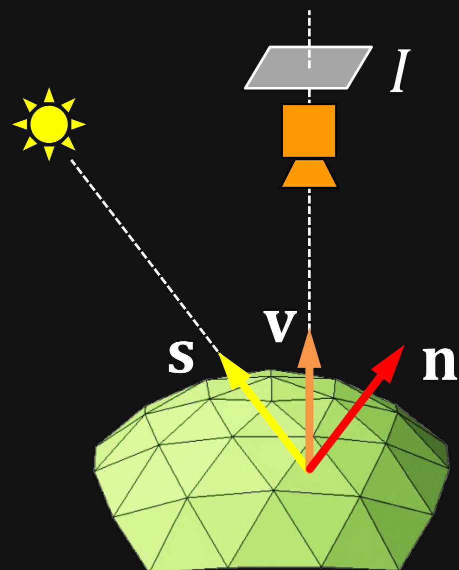


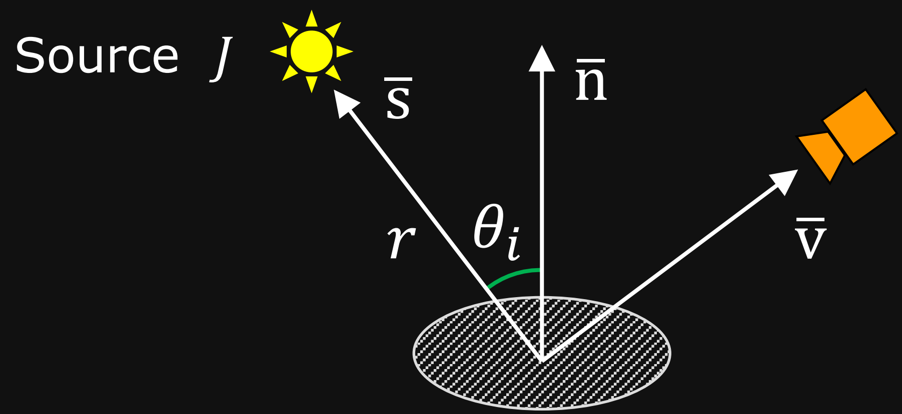

Image Intensity $I = \mathcal{F}($Source Direction $\mathbf{s}$, (Known)

(Known) Surface Normal $\mathbf{n}$, (Unknown)

Surface Reflectance) (Known)

# Photometric Stereo: Lambertian Case

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]

# Photometric Stereo: Lambertian Case

Image irradiance 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 $\mathbf{s_i} = \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix}$

We can write this in matrix format.

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

# Photometric Stereo: Lambertian Case

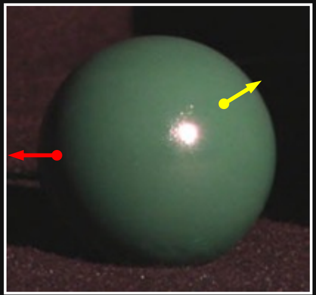
$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \frac{\rho}{\pi} \underbrace{\begin{bmatrix} s_{x_1} & s_{y_1} & s_{z_1} \\ s_{x_2} & s_{y_2} & s_{z_2} \\ s_{x_3} & s_{y_3} & s_{z_3} \end{bmatrix}}_{S_{3\times 3}\ \text{(Known)}} \mathbf{n} \Rightarrow \left\{ \begin{array}{l} I = S\mathbf{N} \\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

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

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

# Calibration-based Photometric Stereo

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



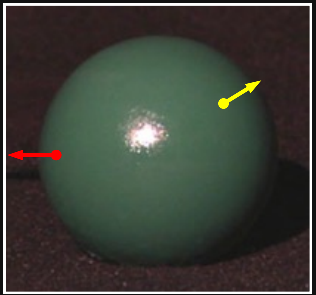
Calibration Sphere


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

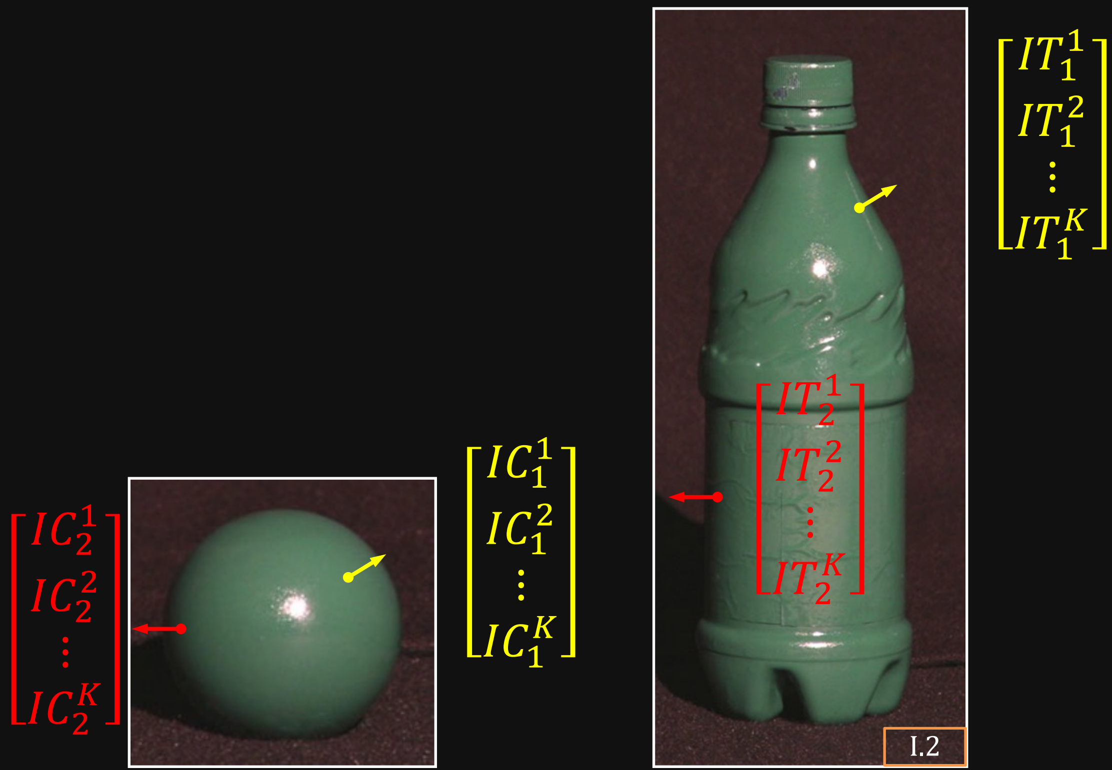

Scene

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

[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 normal produce the same set of intensities under different lighting.

[Silver 1980]

# Calibration Procedure

Image 1

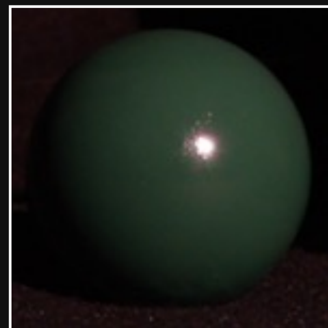


Image 2

…

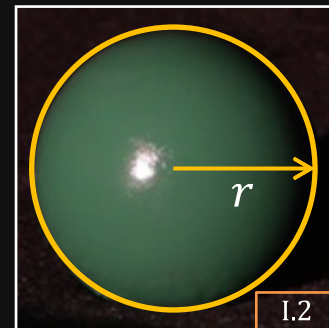
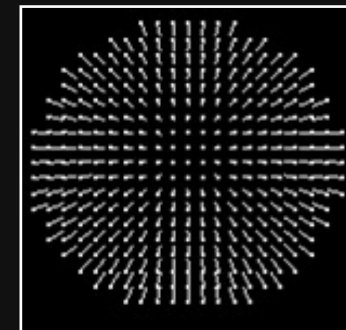



Image K



Surface Normals $(p, q, 1)$

Create a lookup table: $(I_1, I_2, \dots, I_K) \rightarrow (p, q)$

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

| $I_1$ | $I_2$ | … | $I_K$ | $p$ | $q$ |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

[Silver 1980]

# Looking Up Surface Normal

Surface normal estimation:

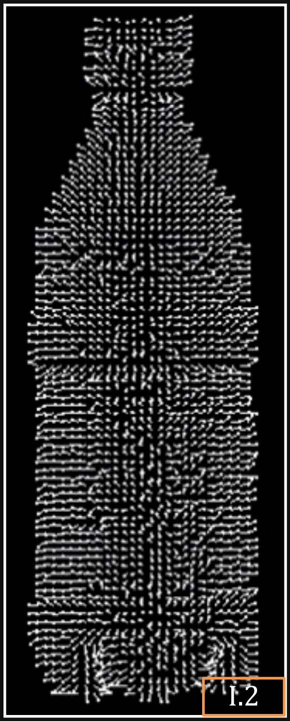
- Capture $K$ images of the scene object under the same $K$ light sources.
- For each pixel, use lookup table to map $(I_1, I_2, \dots, I_K) \rightarrow (p, q)$

Image 1

Image 2

…

Image K




Estimated surface normals

# CS/ECE 766 : Computer Vision

University of Wisconsin-Madison

## Depth from Focus

# Depth from Focus/Defocus

Methods to compute depth by analyzing the amount of focus or defocus in an image.

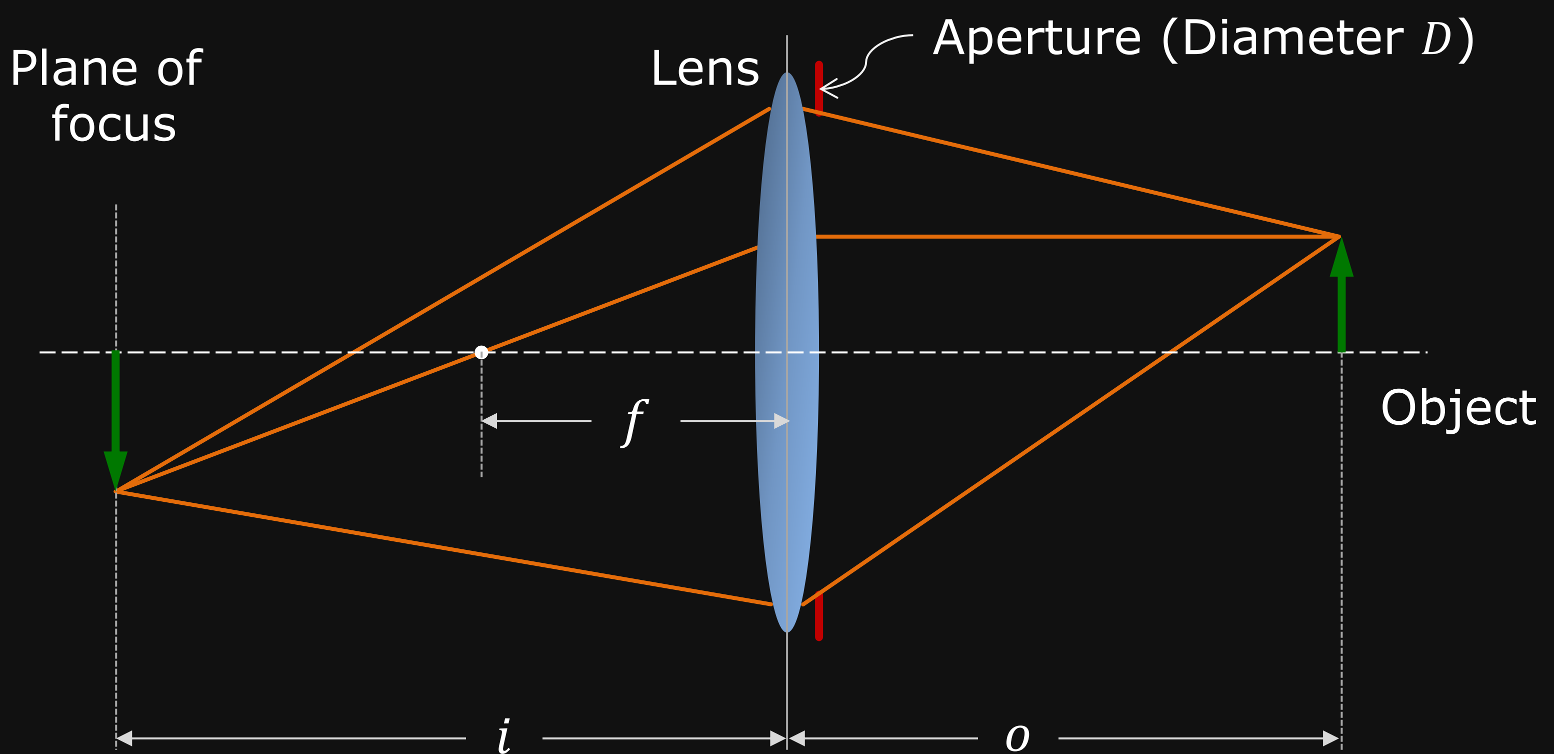
Topics:

(1) Geometry of Defocus

(2) Depth from Focus

(3) Depth from Defocus

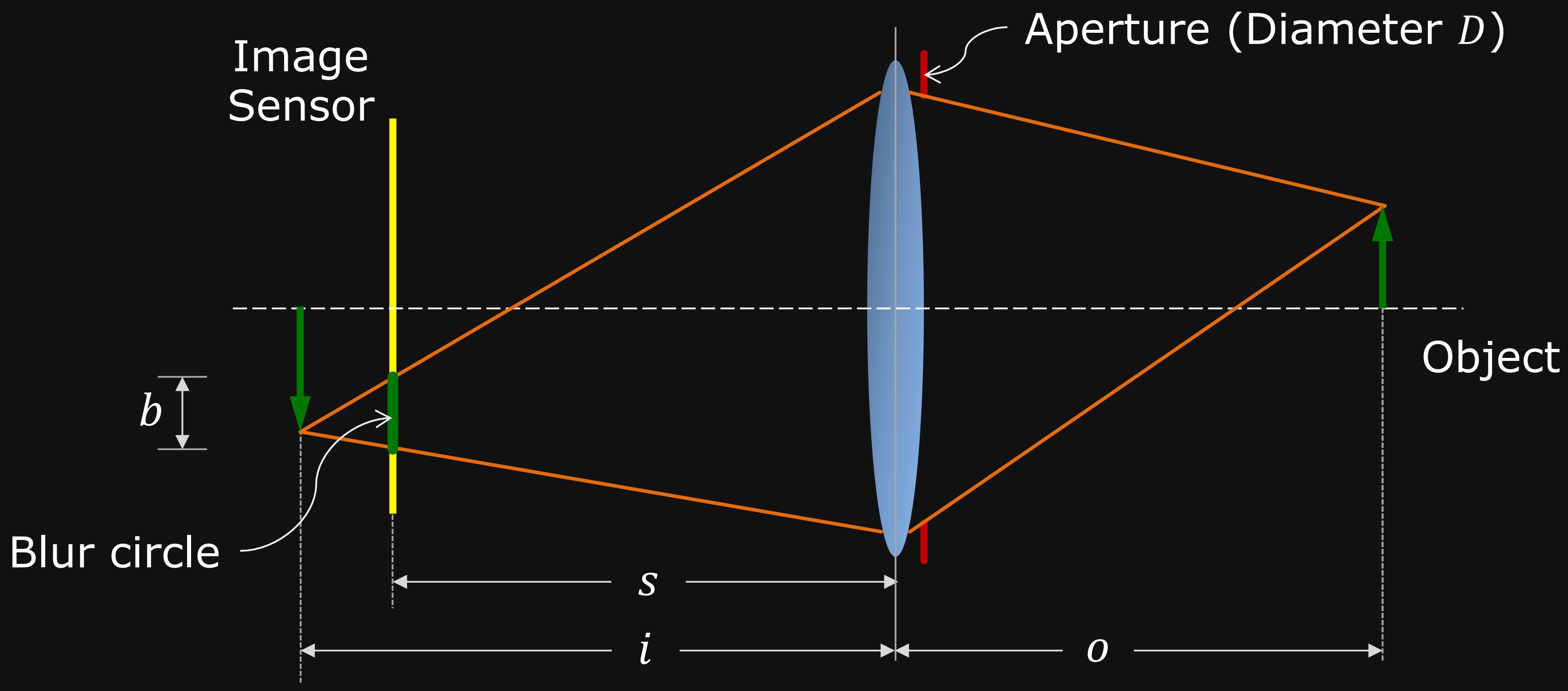
# Review: Gaussian Lens Law




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

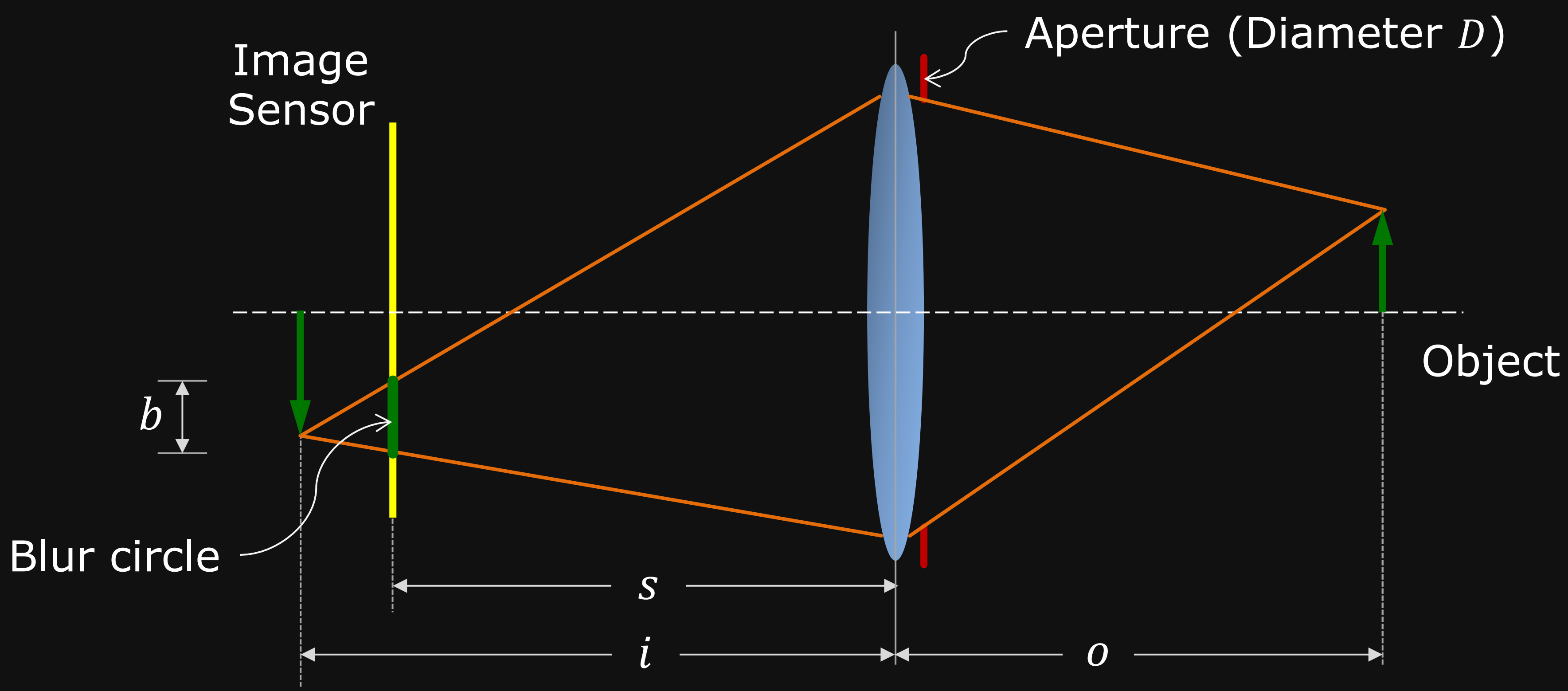
$f$: Focal length

# Geometry of Image Defocus



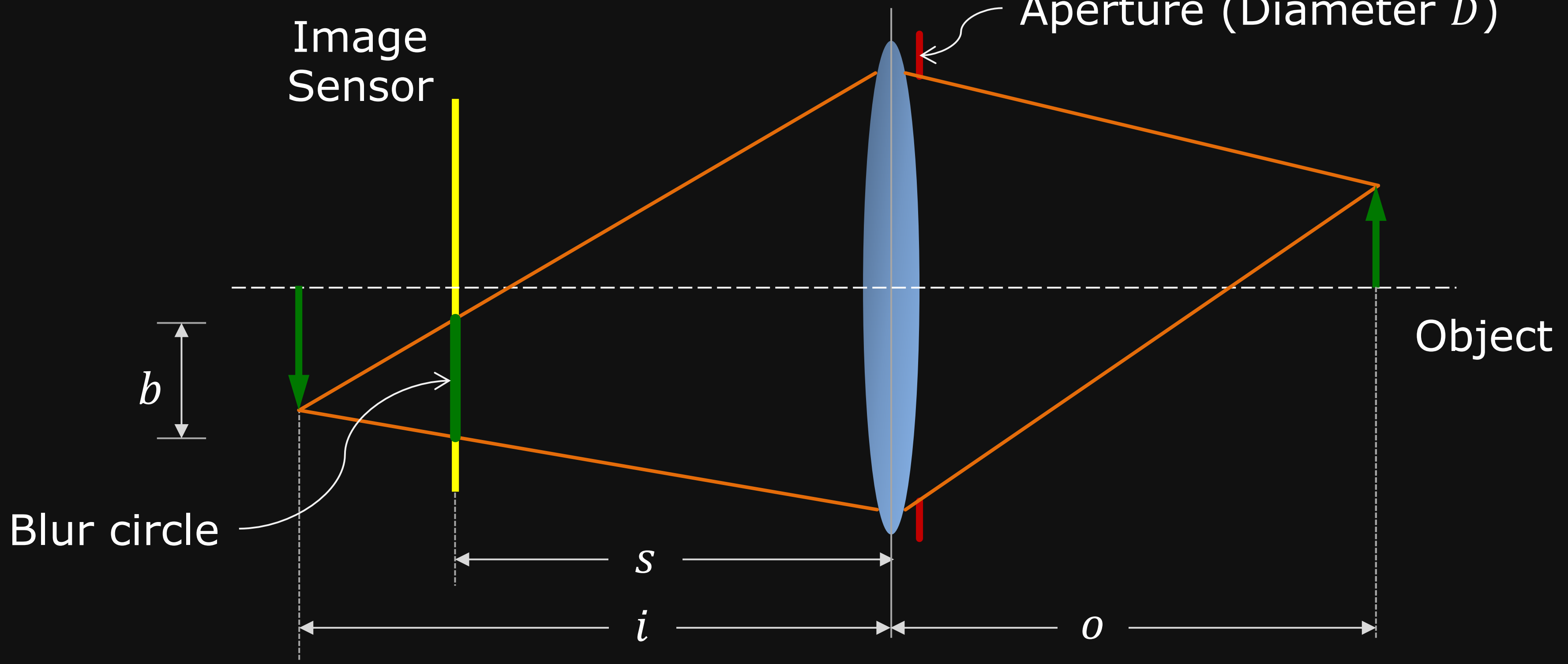

From Similar Triangles: $\frac{b}{D} = \left|\frac{i-s}{i}\right|$

# Image Defocus and Sensor Location




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

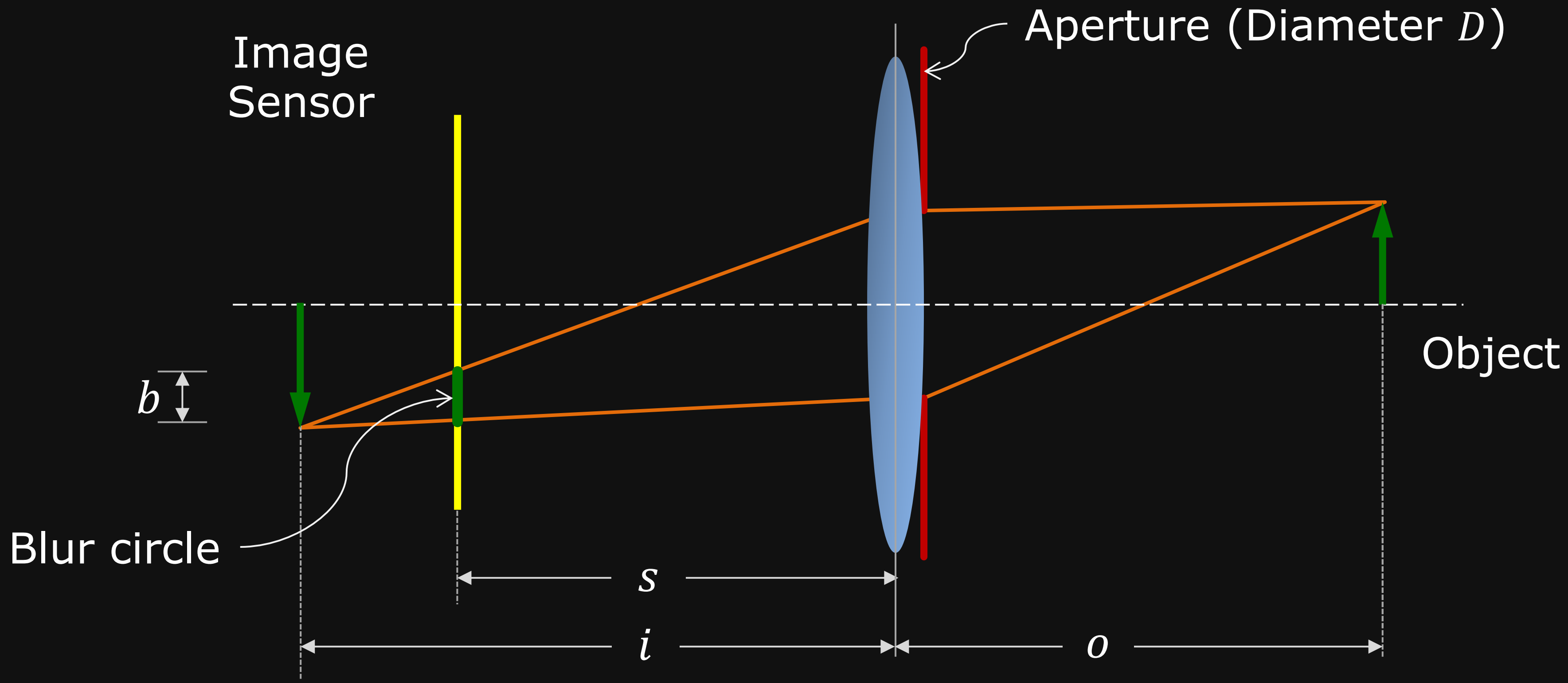
# Image Defocus and Sensor Location




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

Farther the sensor from the plane of focus, larger the blur circle $b$

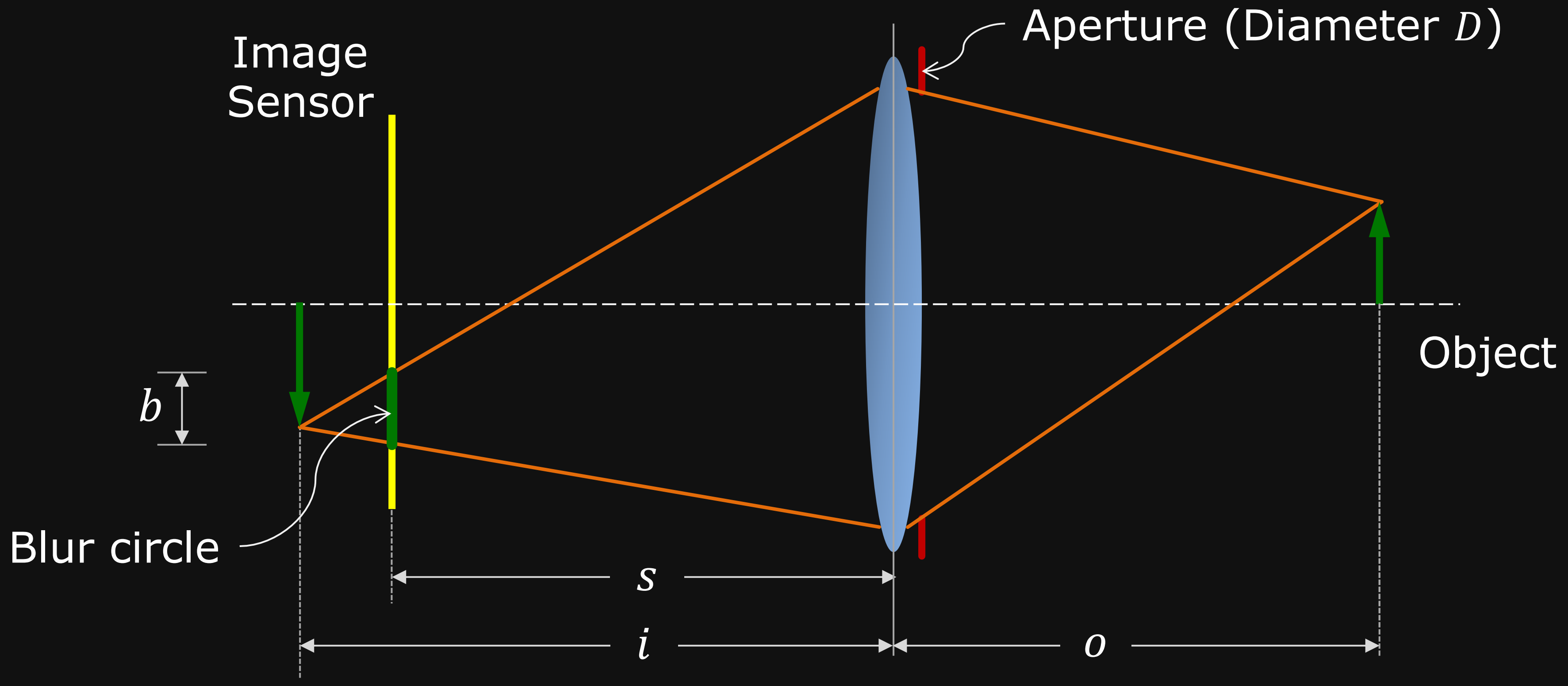
# Image Defocus and Aperture Size




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

Smaller the aperture size $D$, smaller the blur circle $b$

# Depth From Focus (DFF)




Place image sensor at $s = i$

# Depth From Focus (DFF)

Take images with different focus settings by moving the sensor

# Depth From Focus

For each small patch in image, determine 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: $s = 51.25\,mm$

$f = 50\,mm$

$o = 2.05\,m$

Problem: How to find the best focused image?

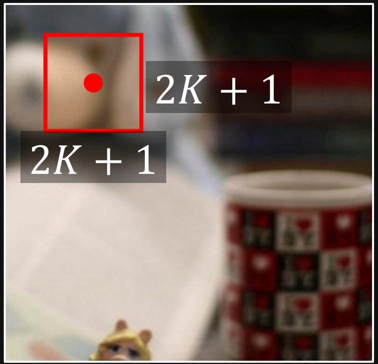
# Focus Measure

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

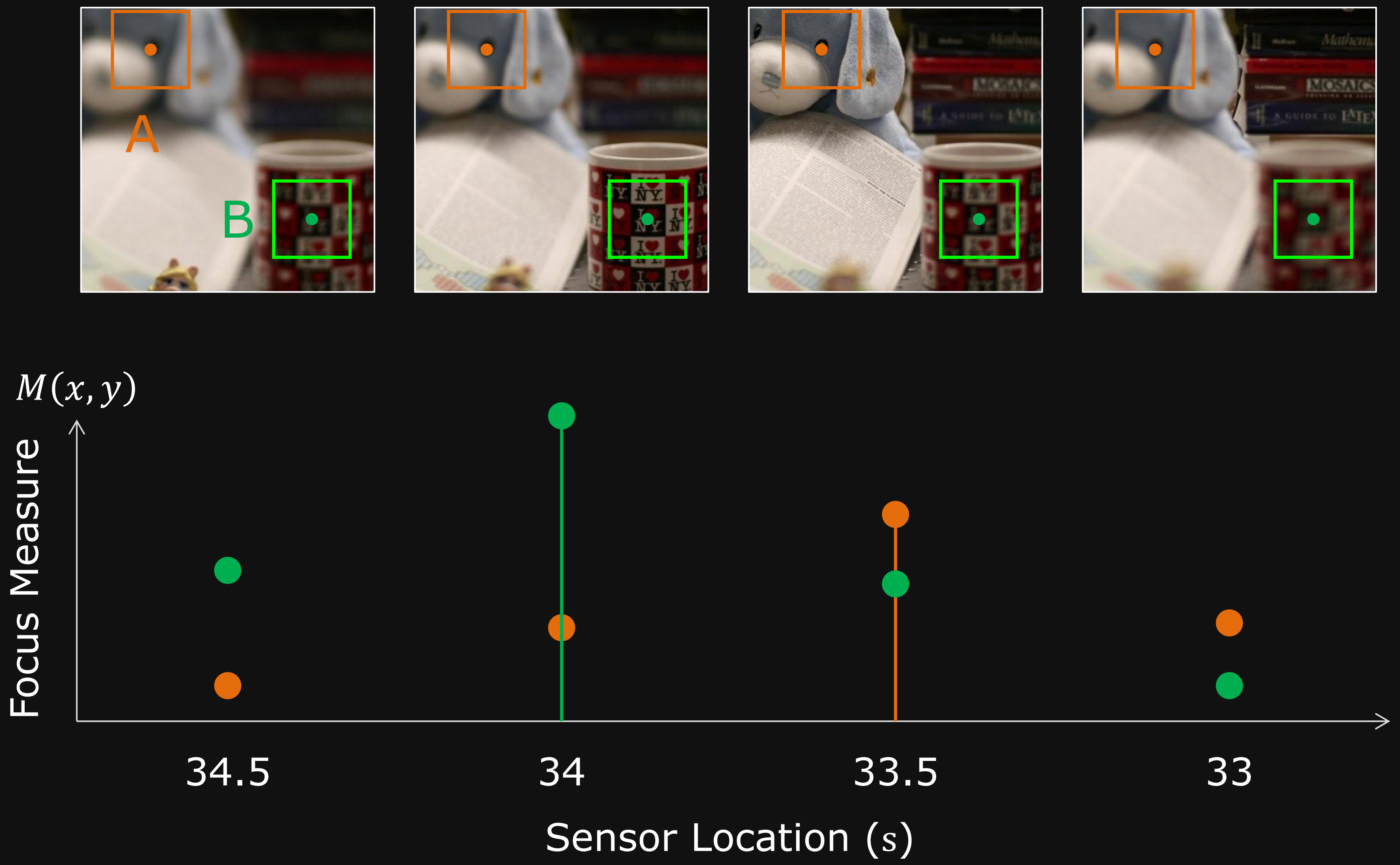
For example, use: $\nabla_M{}^2 f = \left|\frac{\partial^2 f}{\partial x^2}\right| + \left|\frac{\partial^2 f}{\partial y^2}\right|$ (Similar to Laplacian)

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

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



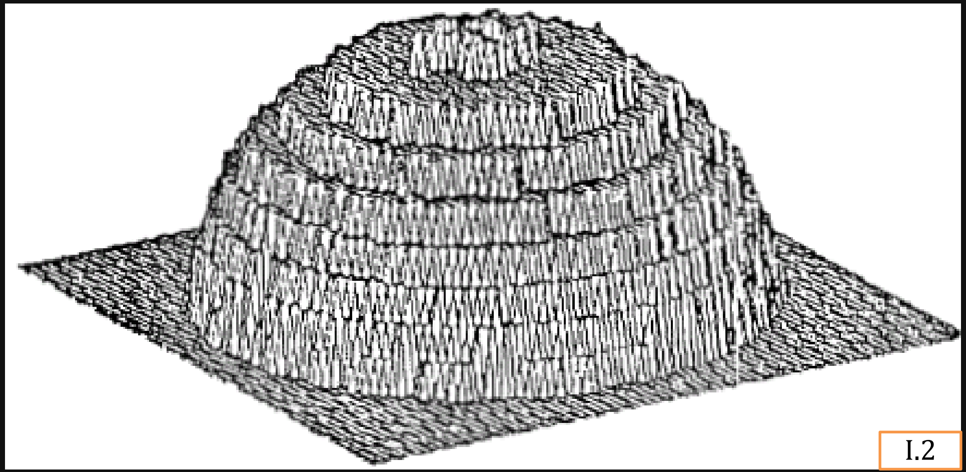
# Depth from Focus



# Depth from Focus

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

Scene
(Metal ball with rough surface)

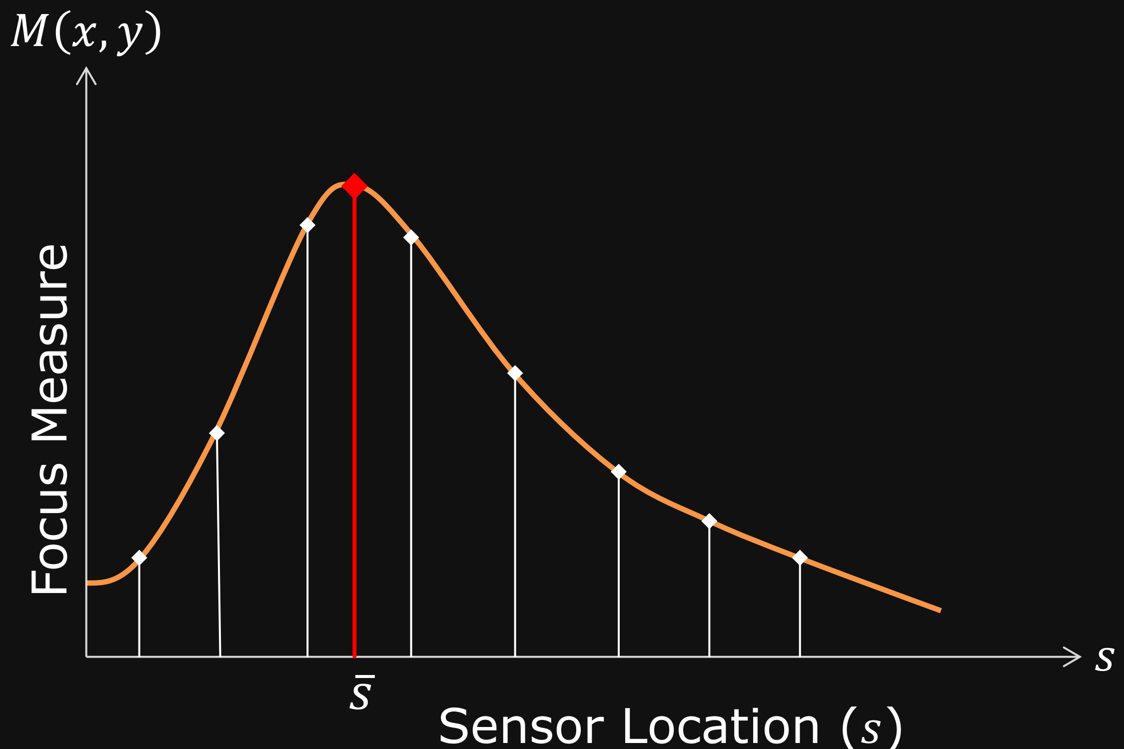



Shape obtained using Depth from Focus

Solution: Take images using many sensor locations. OR...

# Depth Estimation Using Gaussian Interpolation

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

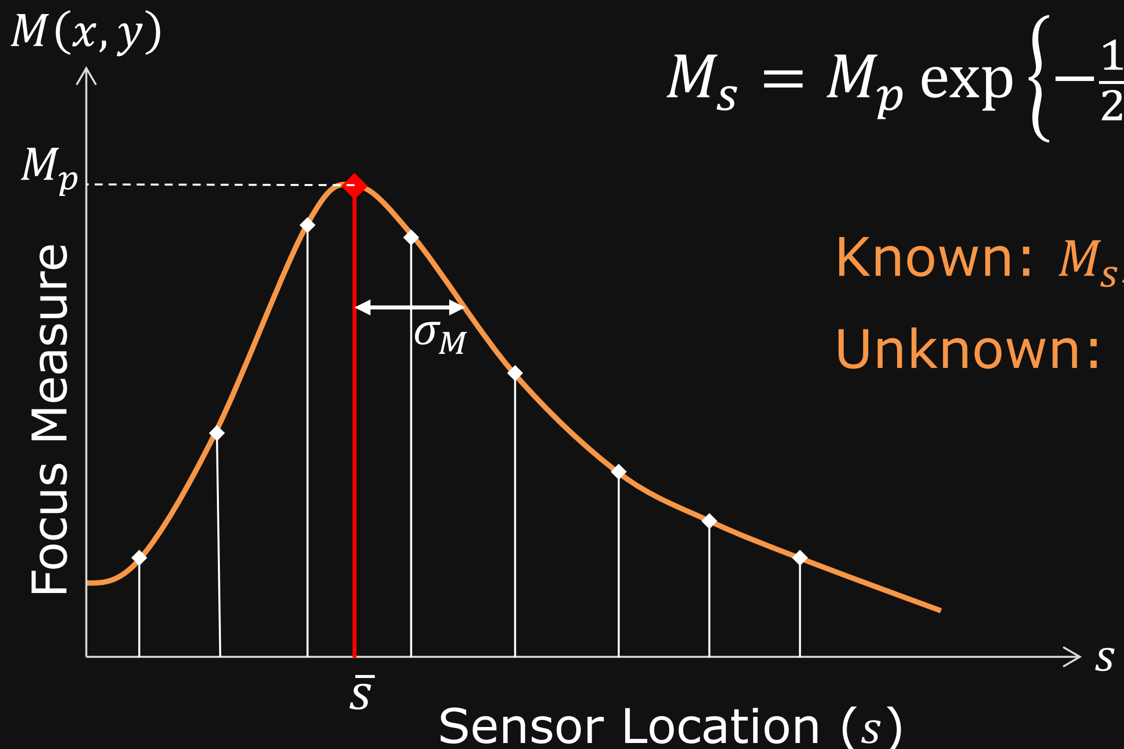



Mean of the Gaussian may be used as the sensor location corresponding to the "best focus."

[Nayar 1994]

# Depth Estimation Using 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: $M_s, s$

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

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

# Gaussian Interpolation

If we can "linearize" the problem, we would need only three samples to estimate $\bar{s}$ as there are only three unknowns $(\bar{s},\ M_p,\ \sigma_M)$.

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

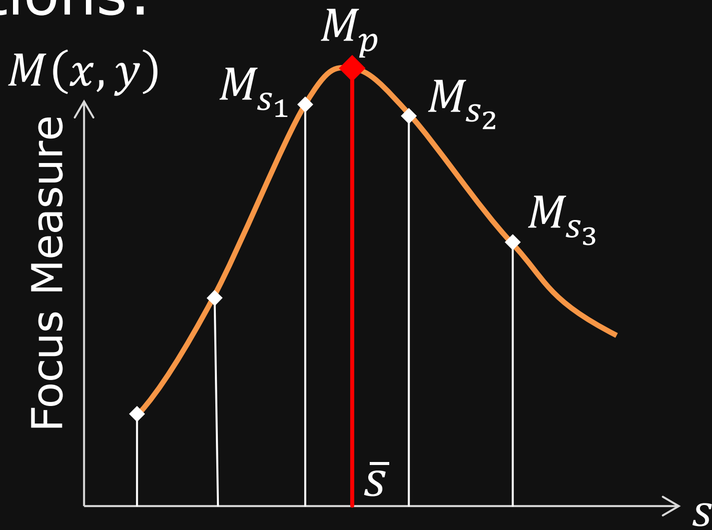
Taking the natural logarithm: $\ln M_s = \ln M_p - \frac{1}{2}\left(\frac{s-\bar{s}}{\sigma_M}\right)^2$

Equations for the 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 highest values.

# Depth Estimation Using Gaussian Interpolation

Solving for $\bar{s}$:

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

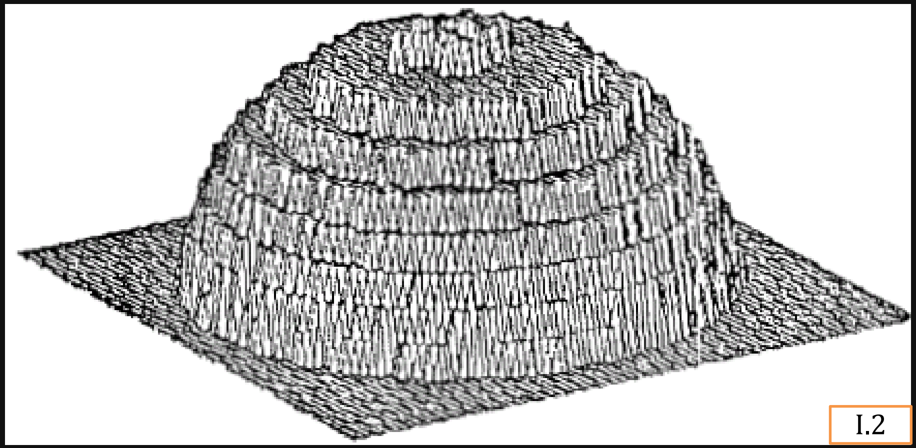
$$- \frac{\left(\ln M_{s_2} - \ln M_{s_1}\right)(s_2^2 - s_3^2)}{2(s_3 - s_2)\left\{\left(\ln M_{s_2} - \ln M_{s_1}\right) + \left(\ln M_{s_2} - \ln M_{s_3}\right)\right\}}$$

Finally, obtain scene depth using Gaussian Lens Law.

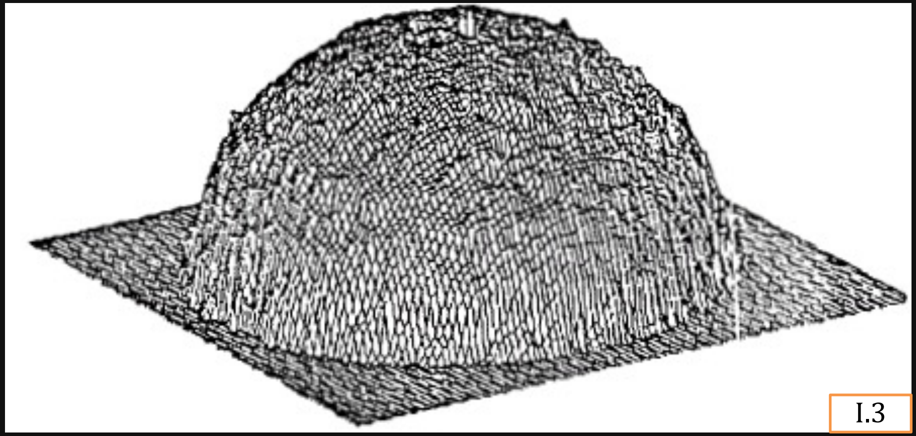
$$\boxed{o = \frac{\bar{s} f}{\bar{s} - f}}$$

# Depth from Focus: Result

Scene
(Metal ball with
rough surface)

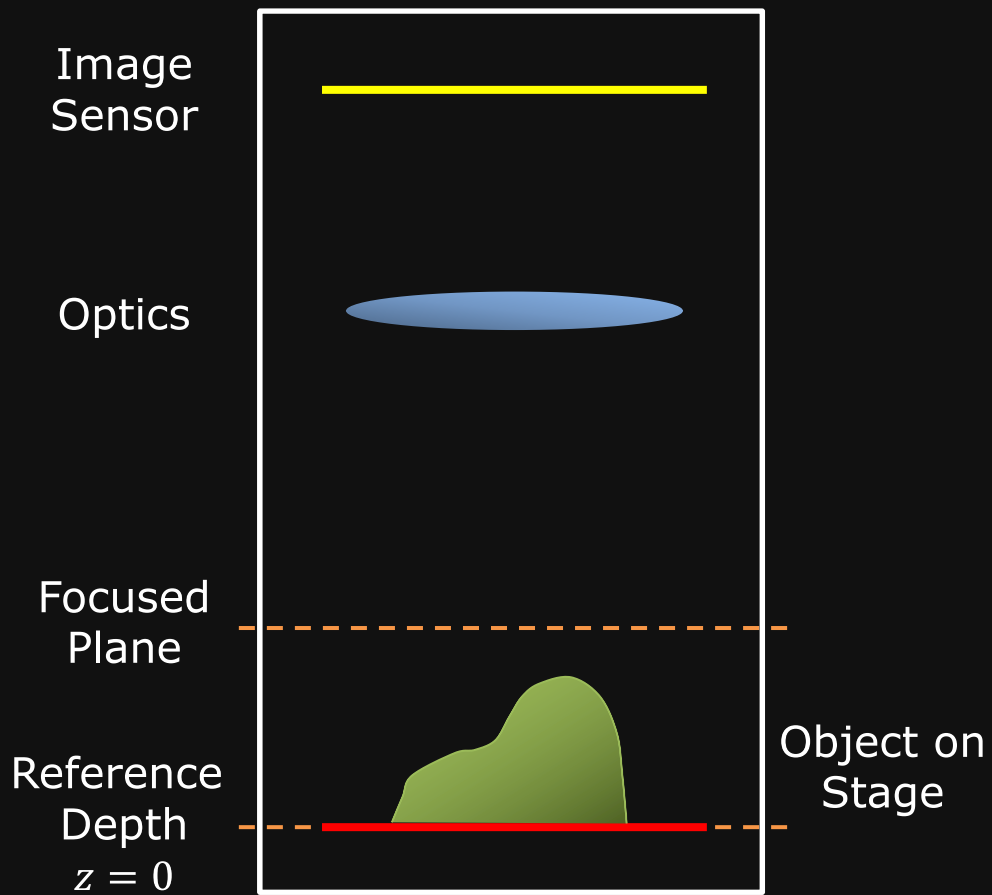



Depth without Gaussian Interpolation




Depth using Gaussian Interpolation

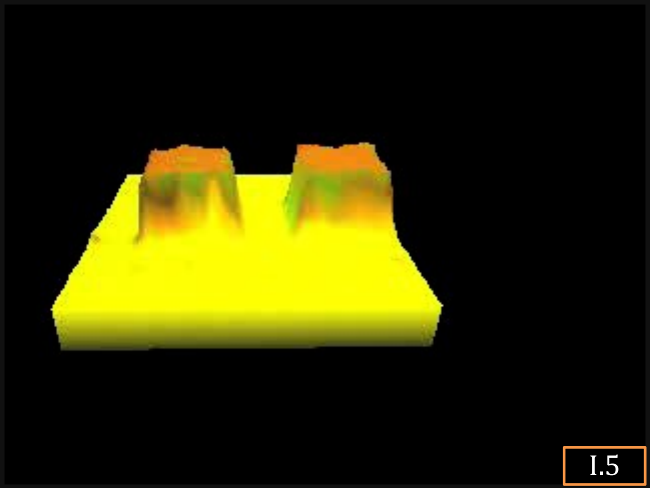
# A Depth from Focus (DFF) System



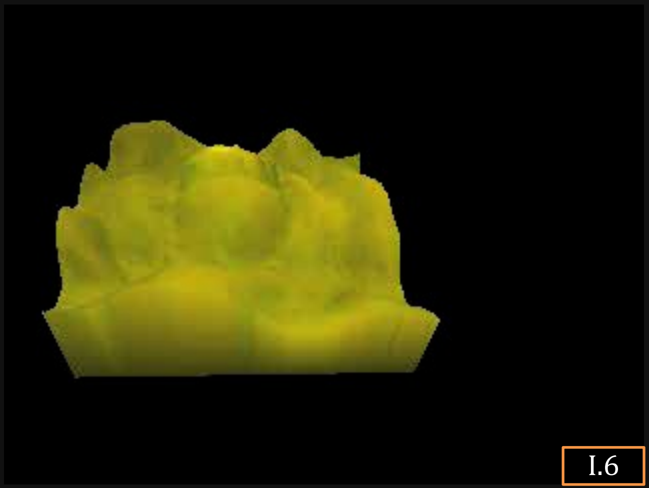

DFF in a Microscope


[Nayar 1994]

# Depth from Focus System: Results




Structures on Silicon Wafer
(13 microns in height)



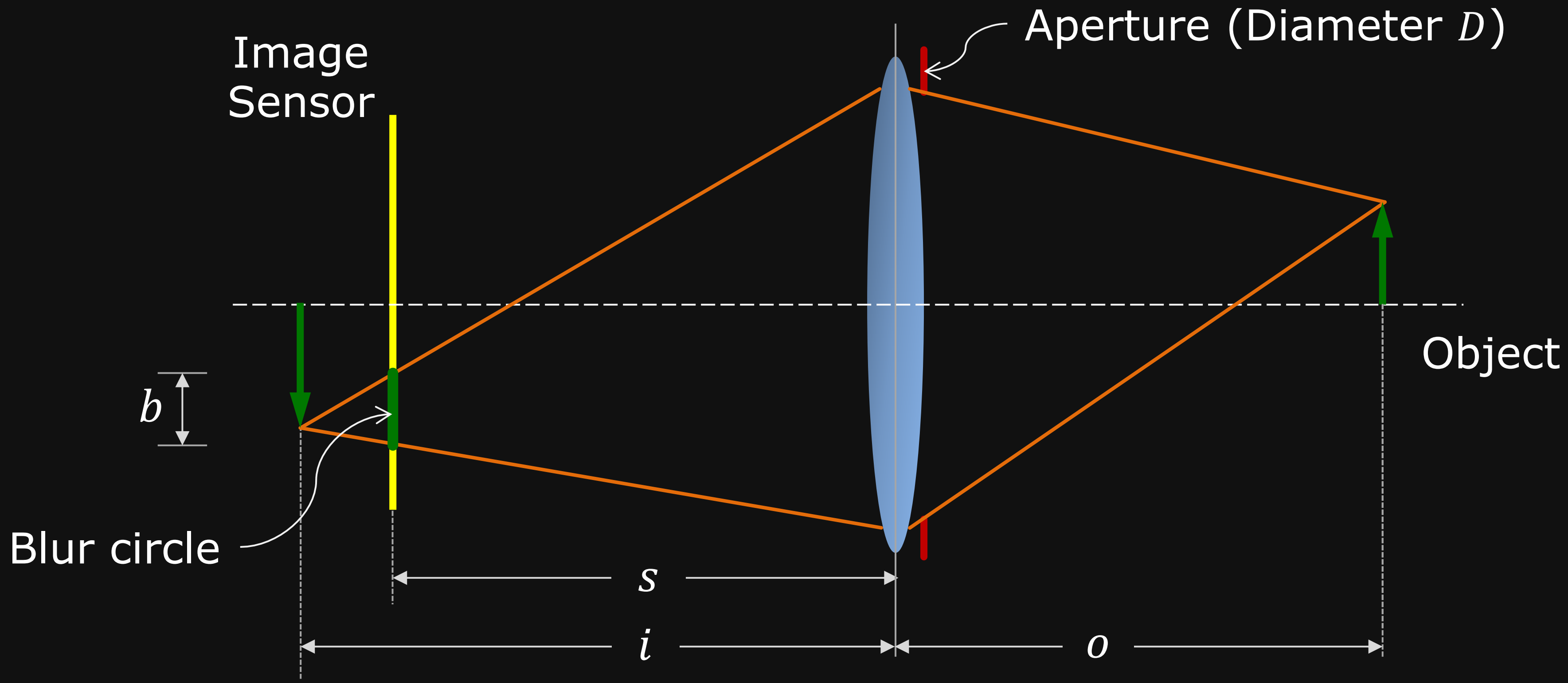

Leaf Stomata (air vent)
(30 microns in height)

[Nayar 1994]

# Depth from Defocus (DFD)

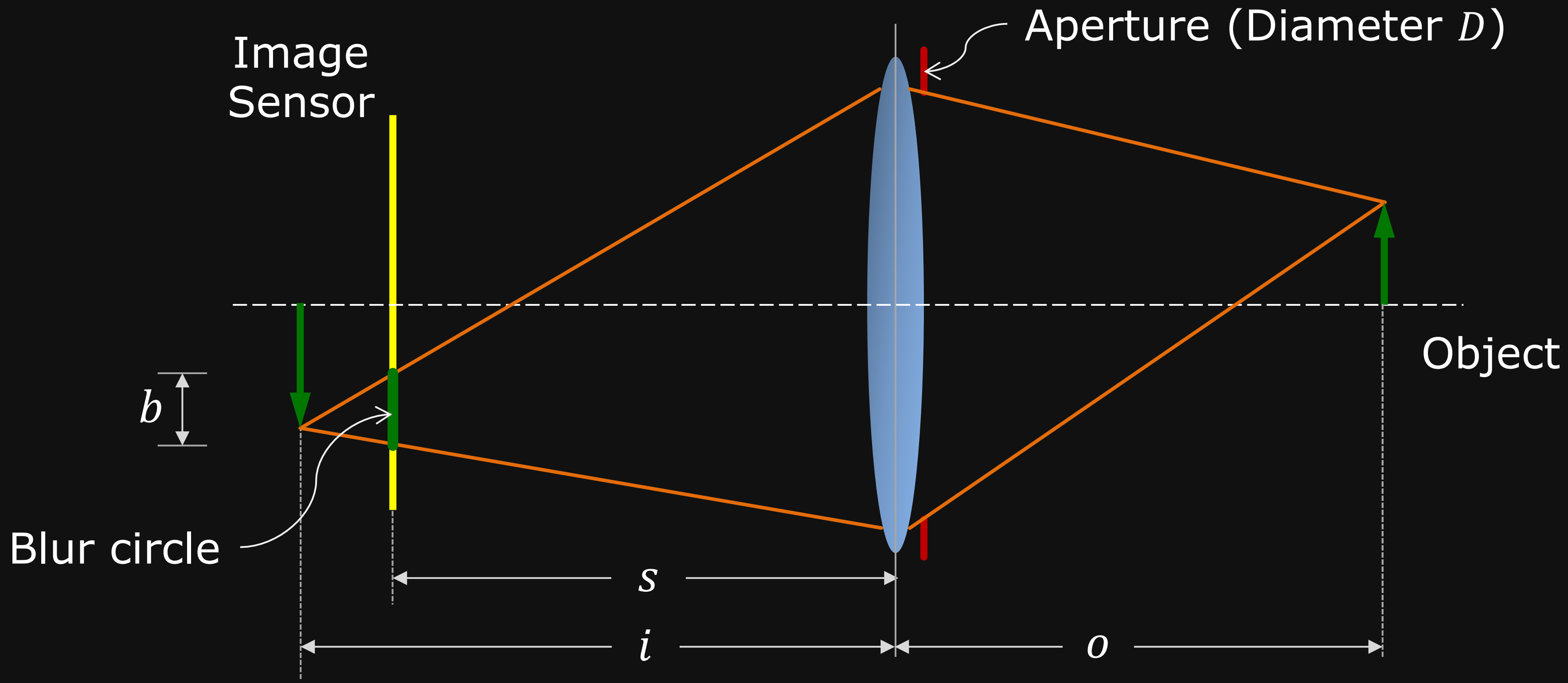
Given an image, the depth of a scene point can be computed if we know how much it is defocused.

# Depth from Defocus (DFD)




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

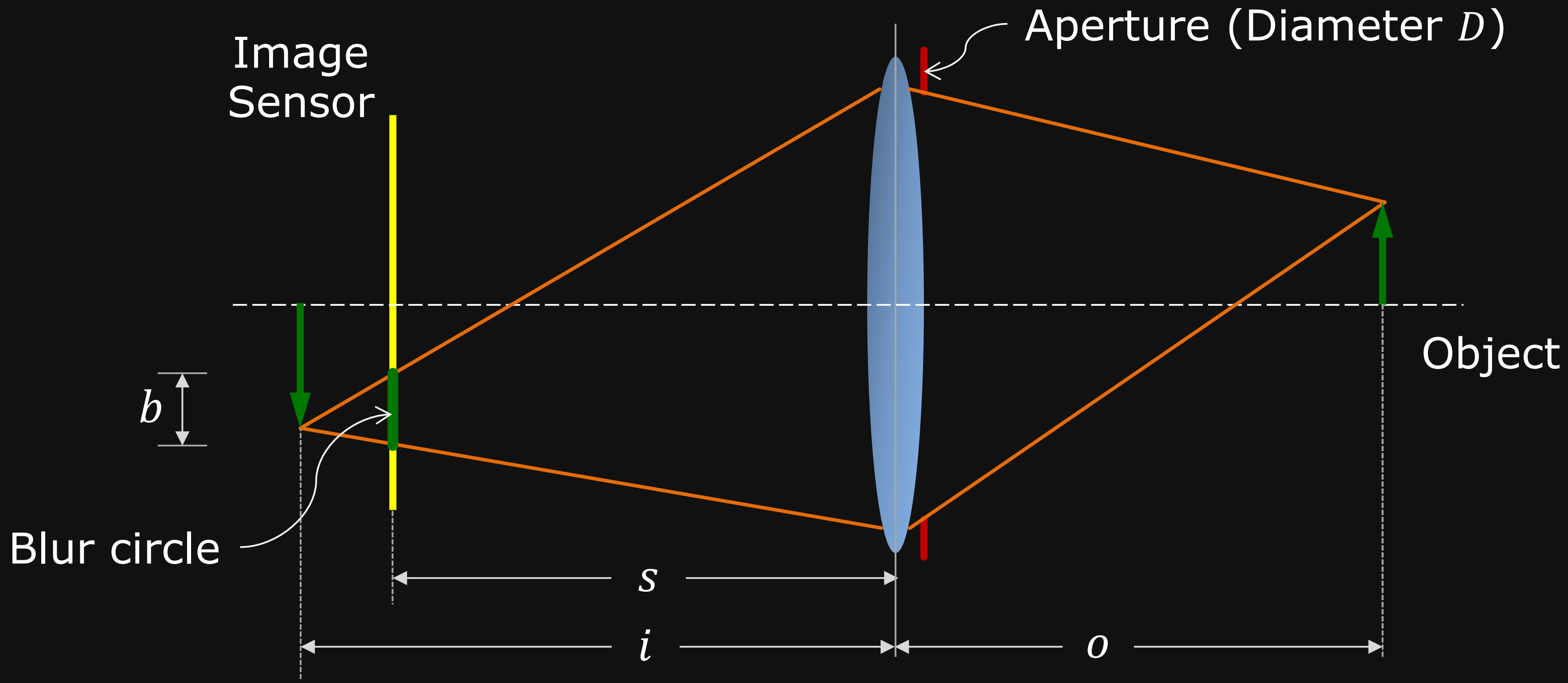
# Depth From Defocus (DFD)




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 (DFD)




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

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

$f/D$: F-Number

# Defocus as Convolution

Within a region where scene depth is constant…

$f_0(x,y)$

$*$

$h(x,y)$

$=$

$f(x,y)$

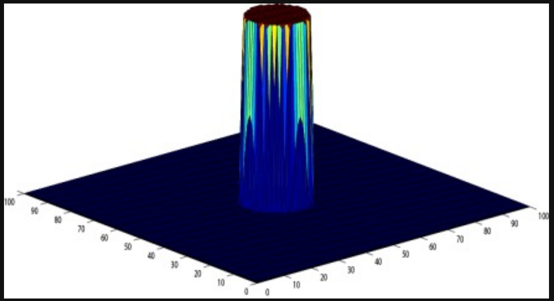
…Defocus is linear and shift invariant, and therefore can be expressed as a convolution.

# Point Spread Function (PSF)

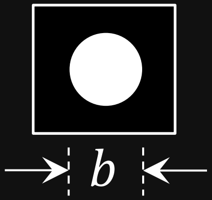
Point Spread Function (PSF): The response of a camera system to a point source (an impulse signal).

Pillbox (Disk) PSF:

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



$h(x,y)$

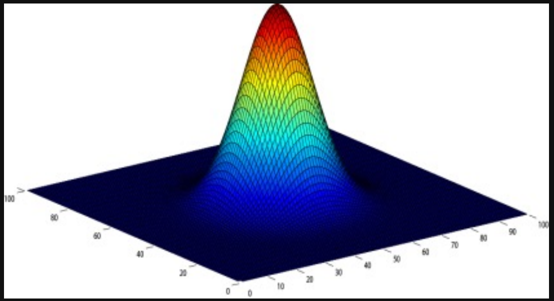


# Point Spread Function (PSF)

In practice, due to optical diffraction, lens aberration, and image sampling issues, 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)

# Depth from Defocus (DFD)

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

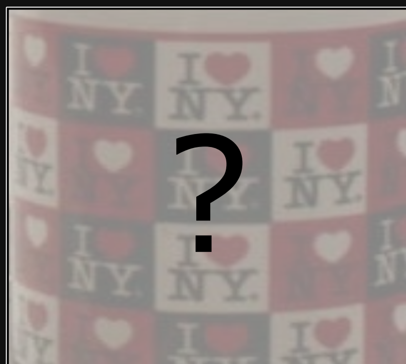

Captured image

Image Patches

PSFs

# Depth from Defocus

Can we estimate blur size from a single image?




$*$


$=$

Scene $f(x,y)$

Defocus PSF $h(x,y)$

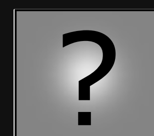
Image Patch $g(x,y)$

Impossible: One equation, two unknowns

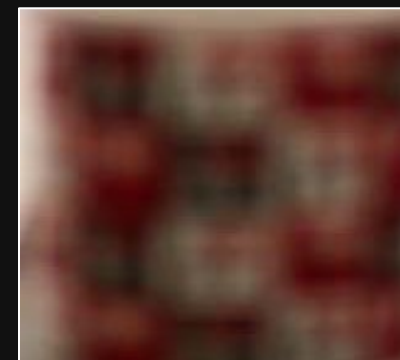
# Depth from Defocus

What if we have two images with different defocus?

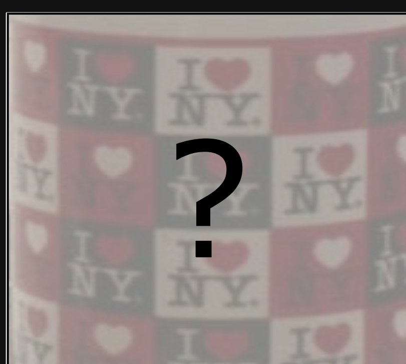

$*$




$=$

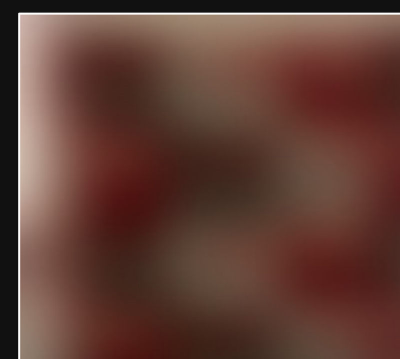


$f(x,y)$ $\quad h_{\sigma_1}(x,y)$ $\quad g_1(x,y)$




$*$


$=$



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

Two equations, three unknowns

# The Third Equation

If the two images were taken with different aperture sizes, then their blur sizes are related.

$D_1$


$D_2$


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

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

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

# Depth from Defocus

For each image patch we have:

Three unknowns

$f(x, y)$

$\sigma_1$

$\sigma_2$

Three equations:

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

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

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

Or, 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,

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

Sensitive to noise making the solution unstable

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

Solve the above to get $\sigma_1$ and $\sigma_2$.

[Pentland 1987]

# Reconstruction-Based DFD Algorithm

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

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

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

We can rewrite $E$ as a 2-variable function:

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

$(\sigma_1, f)$ can be found using: $\dfrac{\partial E}{\partial \sigma_1} = 0$ and $\dfrac{\partial E}{\partial f} = 0$

# Computing Depth From Defocus

Determine $\left(\sigma_1, f(x,y)\right)$ for each patch in the image.

Using $\left(\sigma_1, f(x,y)\right)$ compute the size of blur circle:

$$b_1 = 2\sigma_1$$

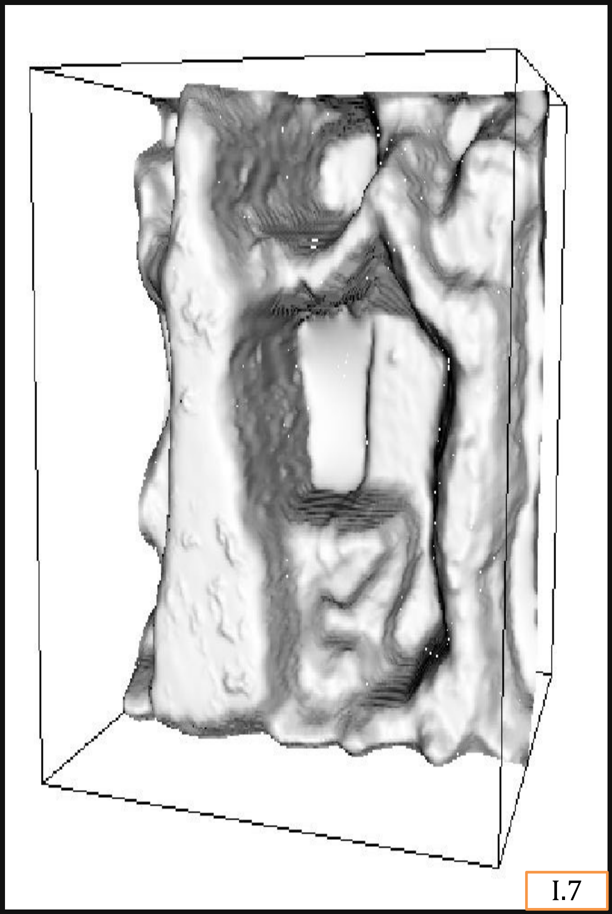
Object distance:

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

# Depth from Defocus: Result

Focused at
the far end

Focused at
the near end




Estimated
3D shape

# Capturing Defocused Images

## Method 1: Change Aperture

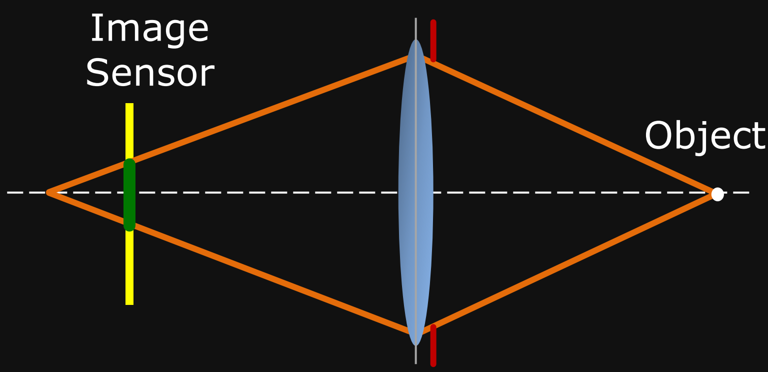



Image 1

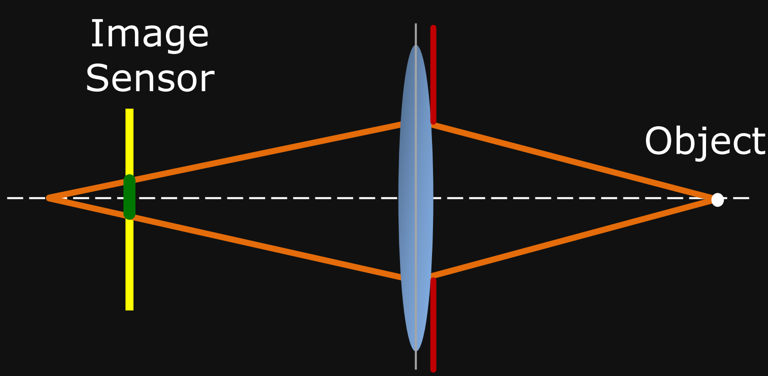



Image 2

## Method 2: Move Sensor

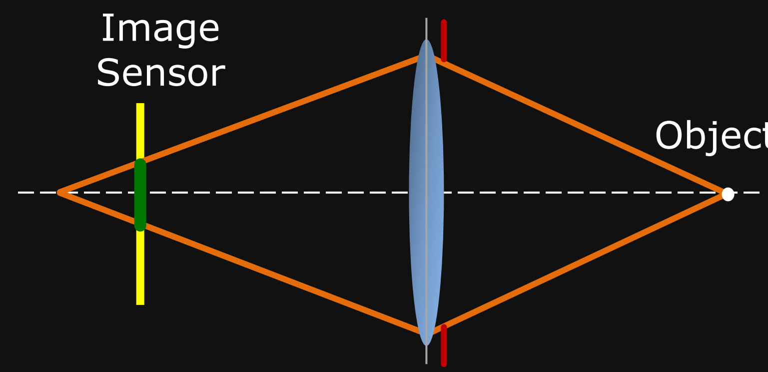



Image 1

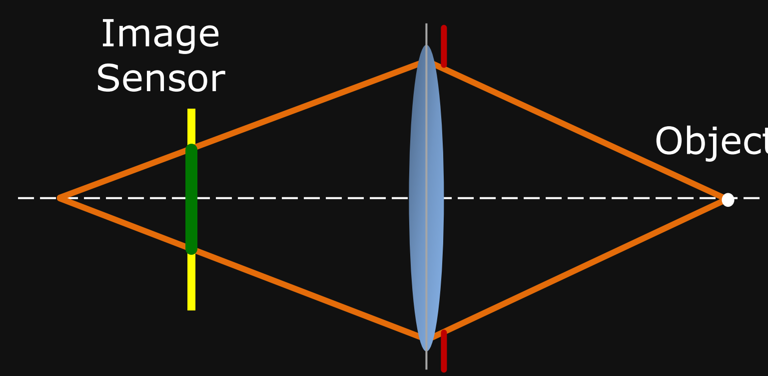
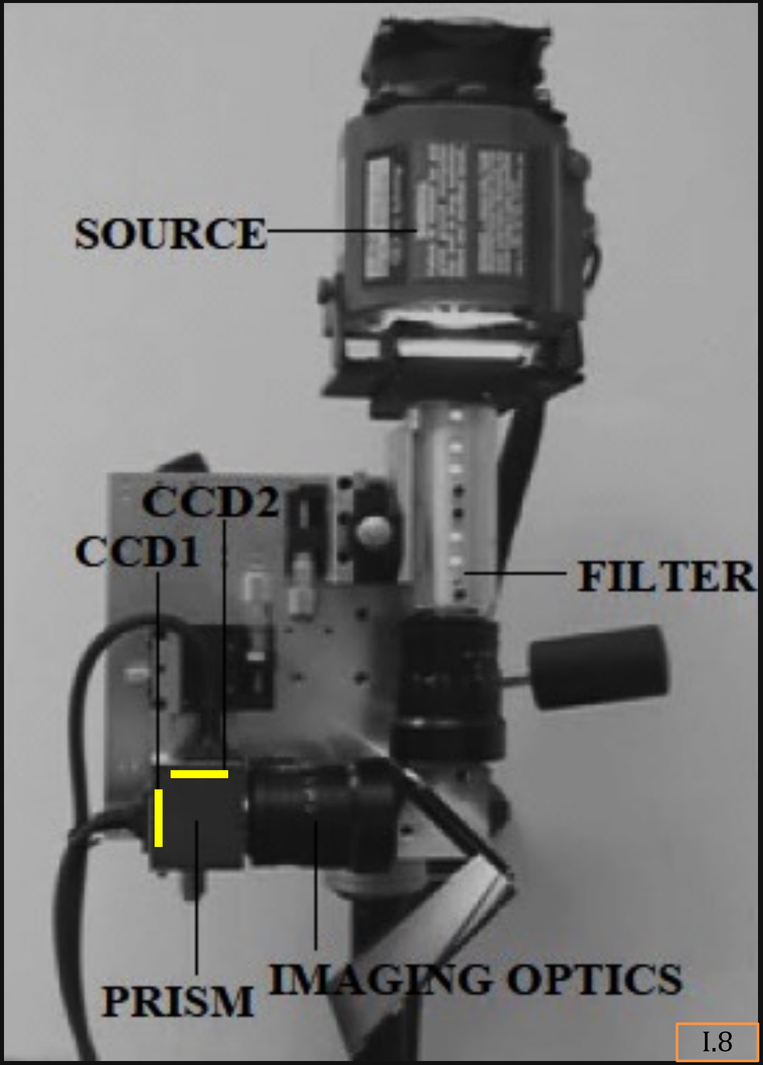


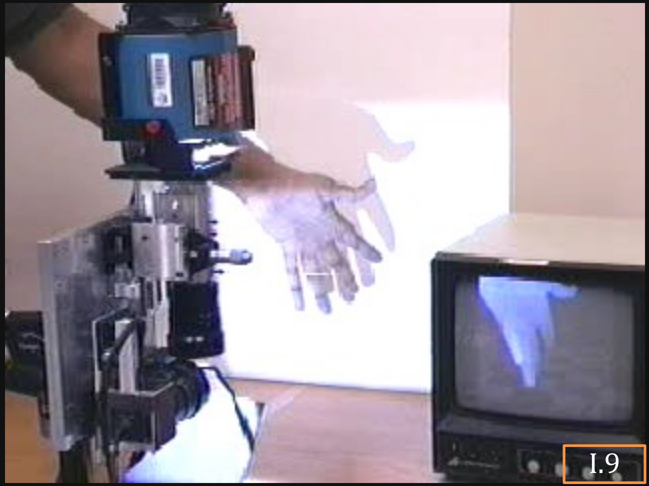

Image 2

# A Depth from Defocus (DFD) System




[Nayar 1996]

# A Depth from Defocus (DFD) System





Real-Time Depth from Defocus System
(uses a fine illumination pattern to ensure texture)

[Nayar 1996]

# References: Papers


[Favaro 2003] P. Favaro, A. Mennucci and S. Soatto, “Observing shape from defocused images”. IJCV, 2003.

[Nayar 1994] S. K. Nayar and Y. Nakagawa, “Shape from Focus,“ PAMI, 1994.

[Nayar 1996] S. K. Nayar, M. Watanabe, and M. Noguchi, “Real-time focus range sensor”. PAMI, 1996.

[Pentland 1987] A. Pentland, “A New Sense for Depth of Field”. PAMI, 1987.

[Subbarao 1994] M. Subbarao and G. Surya, “Depth from defocus: A spatial domain approach”. IJCV, 1994

# Image Credits

I.1 http://www.flickr.com/photos/tim_ellis/75690428/

I.2 Adapted from S. K. Nayar and Y. Nakagawa, “Shape from Focus,“ PAMI 1994.

I.3 Adapted from S. K. Nayar and Y. Nakagawa, “Shape from Focus,“ PAMI 1994.

I.4 http://www.cs.columbia.edu/CAVE/projects/shape_focus/

I.5 http://www.cs.columbia.edu/CAVE/projects/shape_focus/

I.6 http://www.cs.columbia.edu/CAVE/projects/shape_focus/

I.7 http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/FAVARO1/dfdtutorial.html.

I.8 http://www.cs.columbia.edu/CAVE/projects/depth_defocus/

I.9 http://www.cs.columbia.edu/CAVE/projects/depth_defocus/

I.10 http://www.cs.columbia.edu/CAVE/projects/depth_defocus/