# Depth From Focus/Defocus

Introduction to Computational Photography:

EECS 395/495

Northwestern University

I.1

# Depth from Focus/Defocus

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

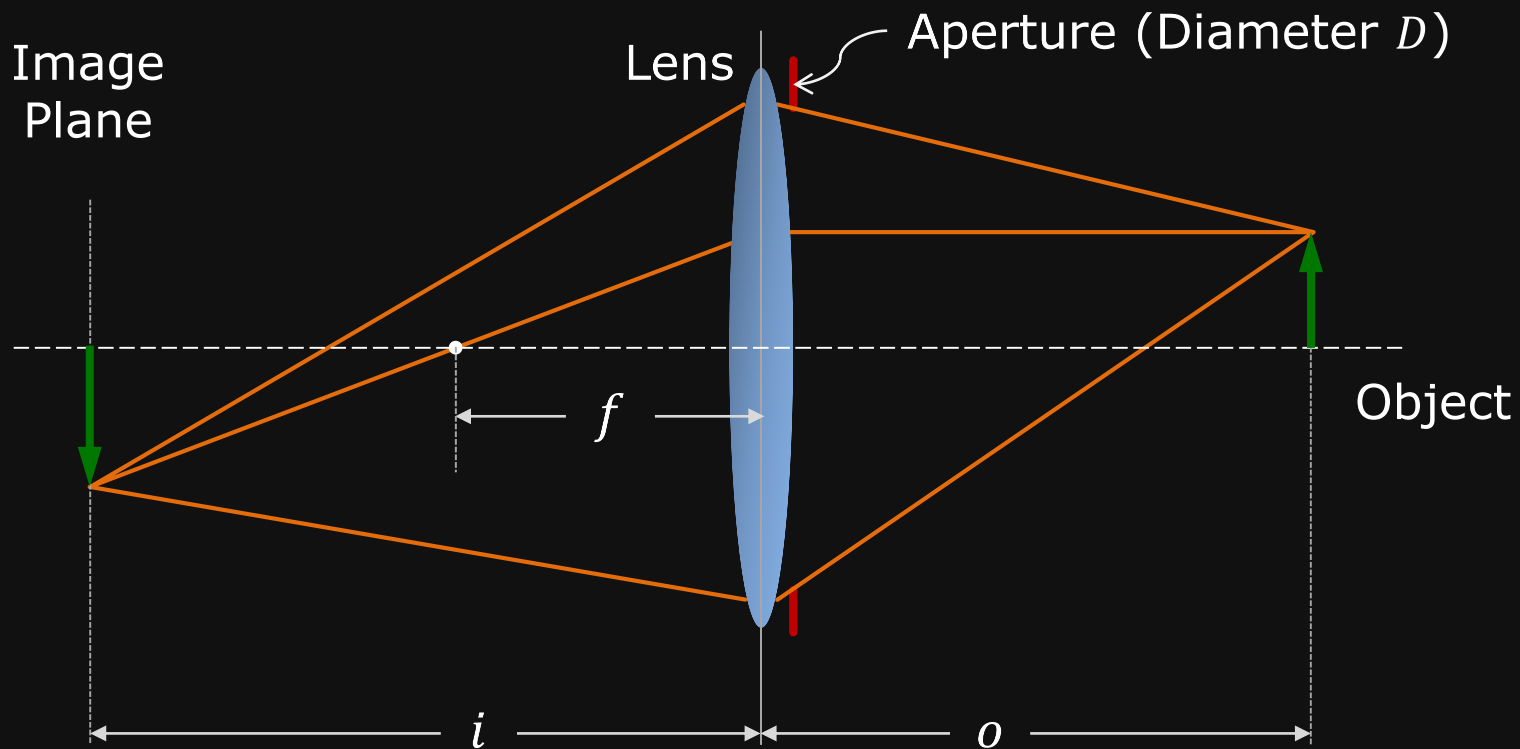
## Topics:

(1) Geometry of Defocus

(2) Depth from Focus

(3) Depth from Defocus

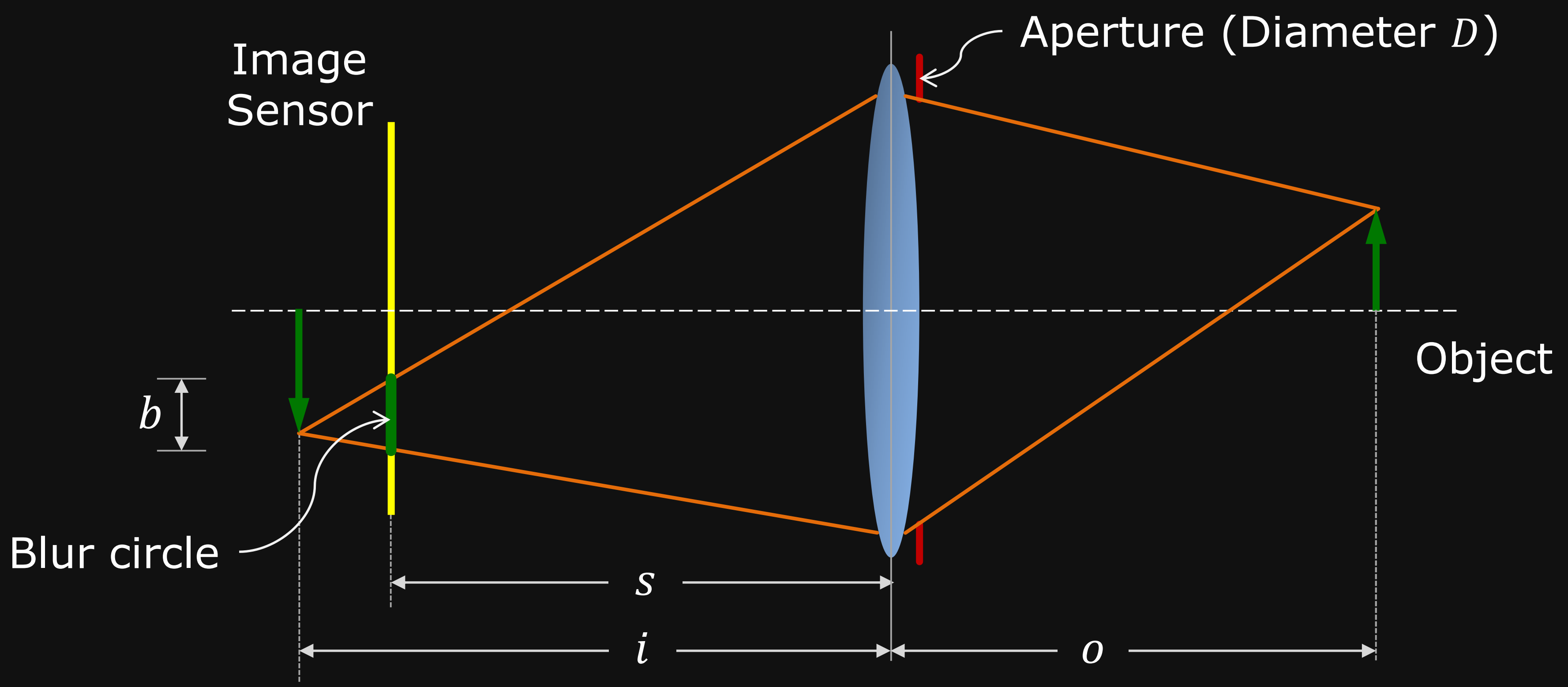
# Review: Gaussian Lens Law




Gaussian Lens Law: $$\boxed{\frac{1}{f} = \frac{1}{i} + \frac{1}{o}}$$

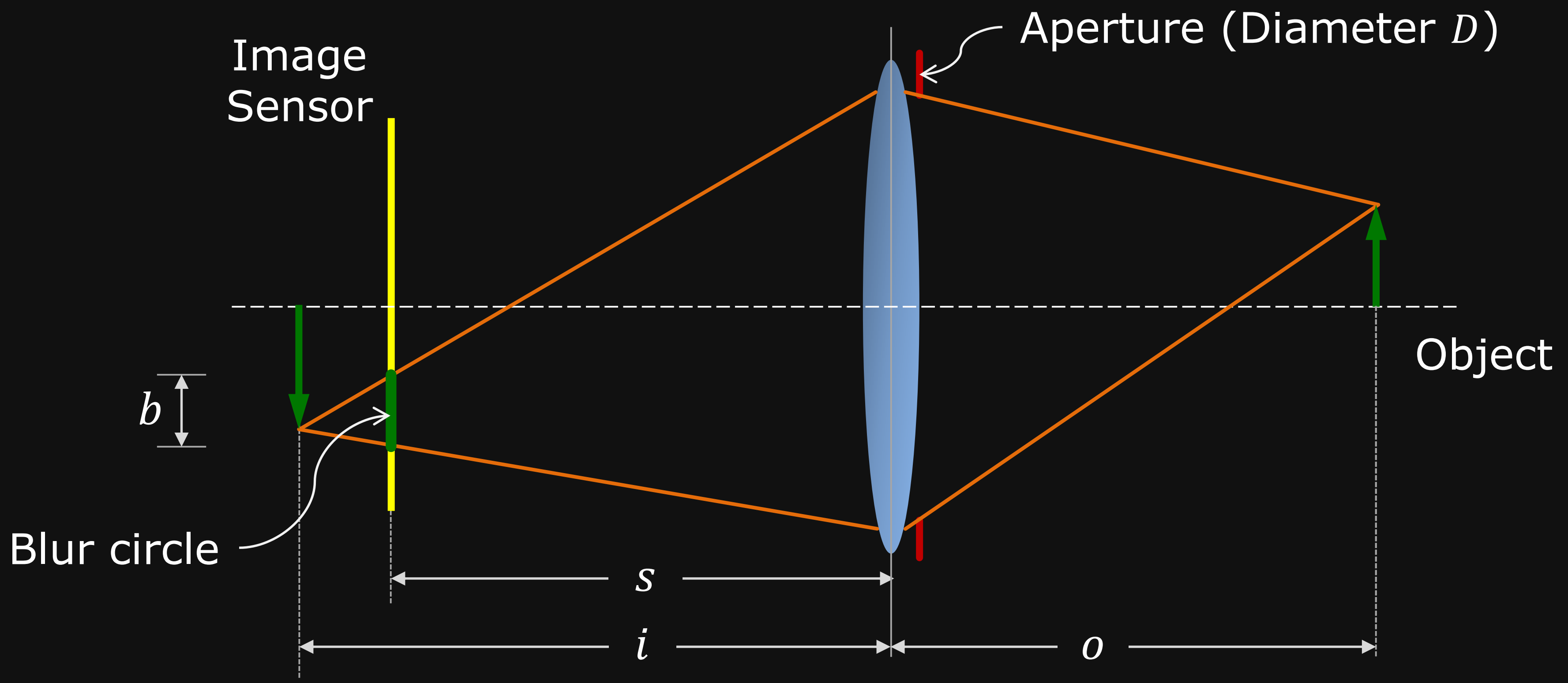
$f$: Focal Length

# Geometry of Image Defocus



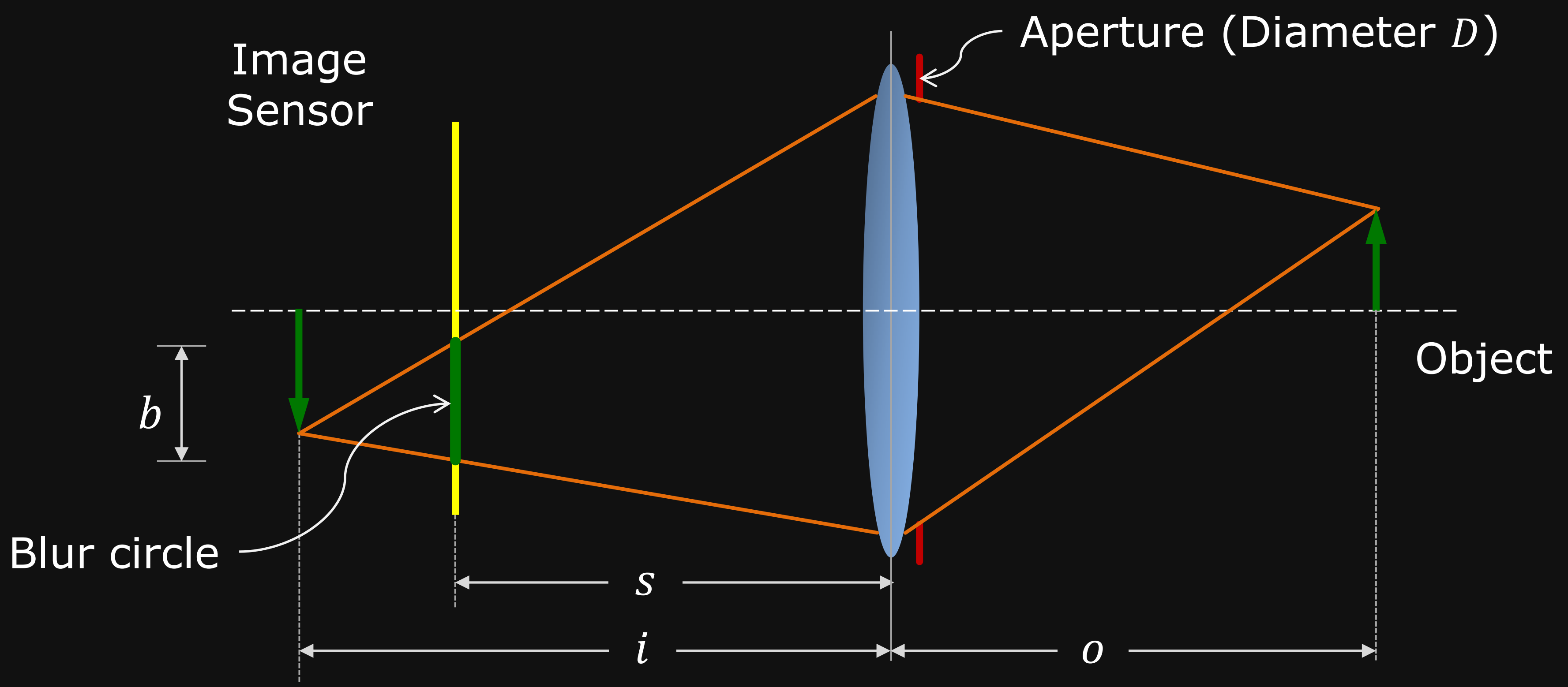

From Similar Triangles: $\frac{b}{D} = \left|\frac{i-s}{i}\right|$

# Image Defocus and Sensor Location




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

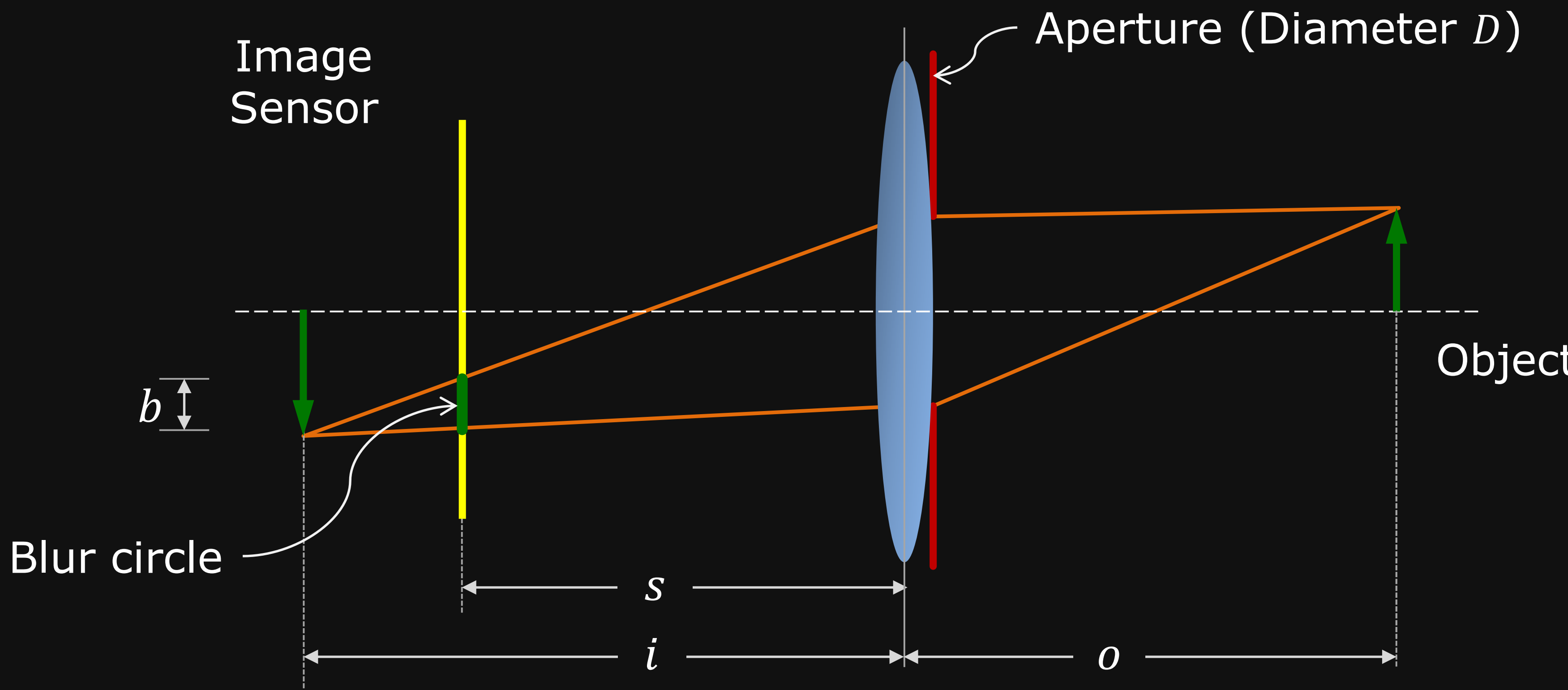
# Image Defocus and Sensor Location




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

Farther the Sensor from the Image Plane, Larger the Blur Circle $b$

# Image Defocus and Aperture Size




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

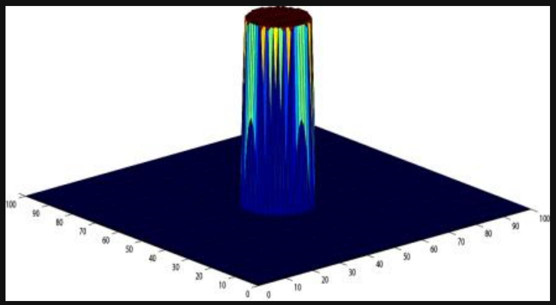
Smaller the Aperture Size $D$, Smaller the Blur Circle $b$

# Point Spread Function (PSF)

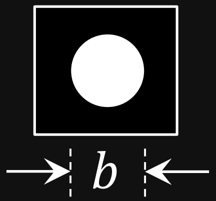
Point Spread Function (PSF): The response of a camera system to a point source (an impulse signal).

Pillbox (Disk) PSF:

$$h(x,y) = \begin{cases} 1, & x^2 + y^2 \leq b^2/4 \\ 0, & otherwise \end{cases}$$



$h(x,y)$



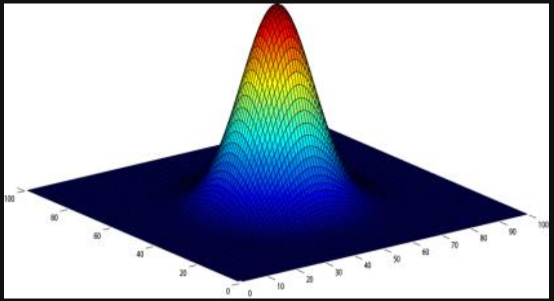

Blur circle

# 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:

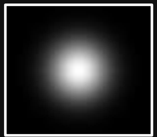
$$h(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



$h(x,y)$

Sigma $\sigma$ of Gaussian PSF
= Radius $b/2$ of Blur Circle

$$\boxed{\sigma = b/2}$$



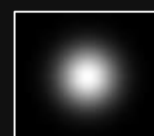
Blur function

# Defocus as Convolution

Within a region where scene depth is constant…


$f_0(x, y)$

$*$



$h(x, y)$

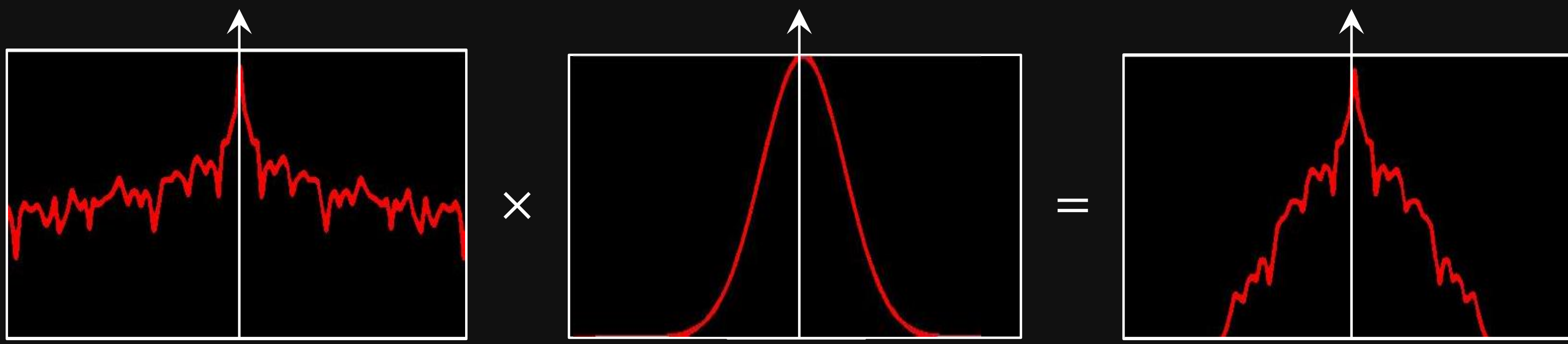
$=$

$f(x, y)$

…Defocus is Linear and Shift Invariant, and therefore can be formulated as Convolution.

# Defocus in Fourier Domain

Defocus can be represented as Product of Fourier Transforms



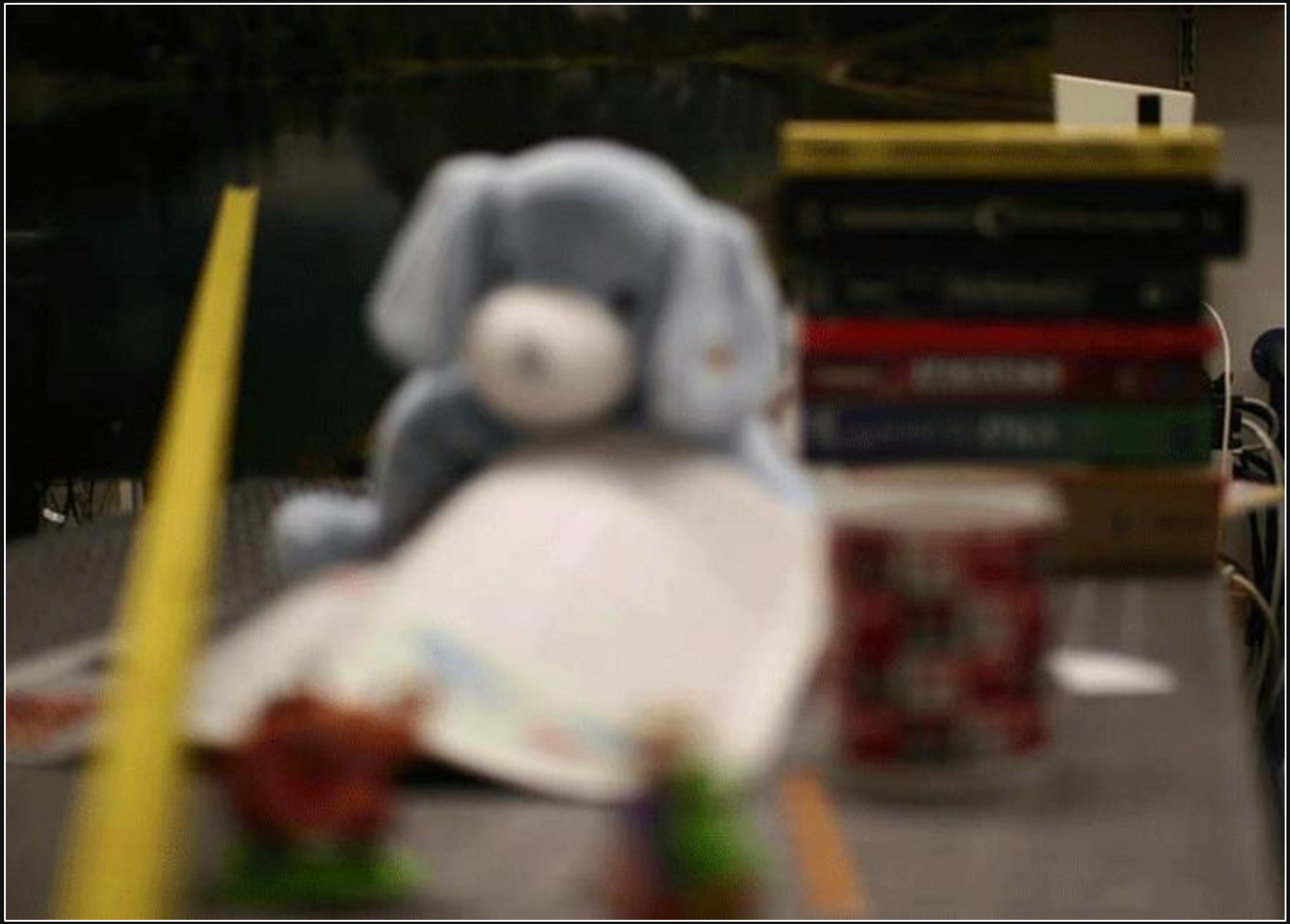

$F_0(u,v)$ $H(u,v)$ $F(u,v)$

(1D slice of Fourier Transform is shown)

Defocus is a Low-Pass Filter

# 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?

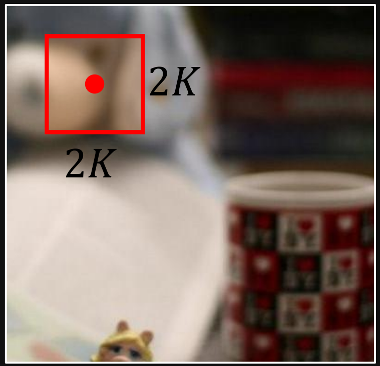
# Focus Measure

Use a High-Pass Filter to measure the amount of High Frequency Content within each small patch.

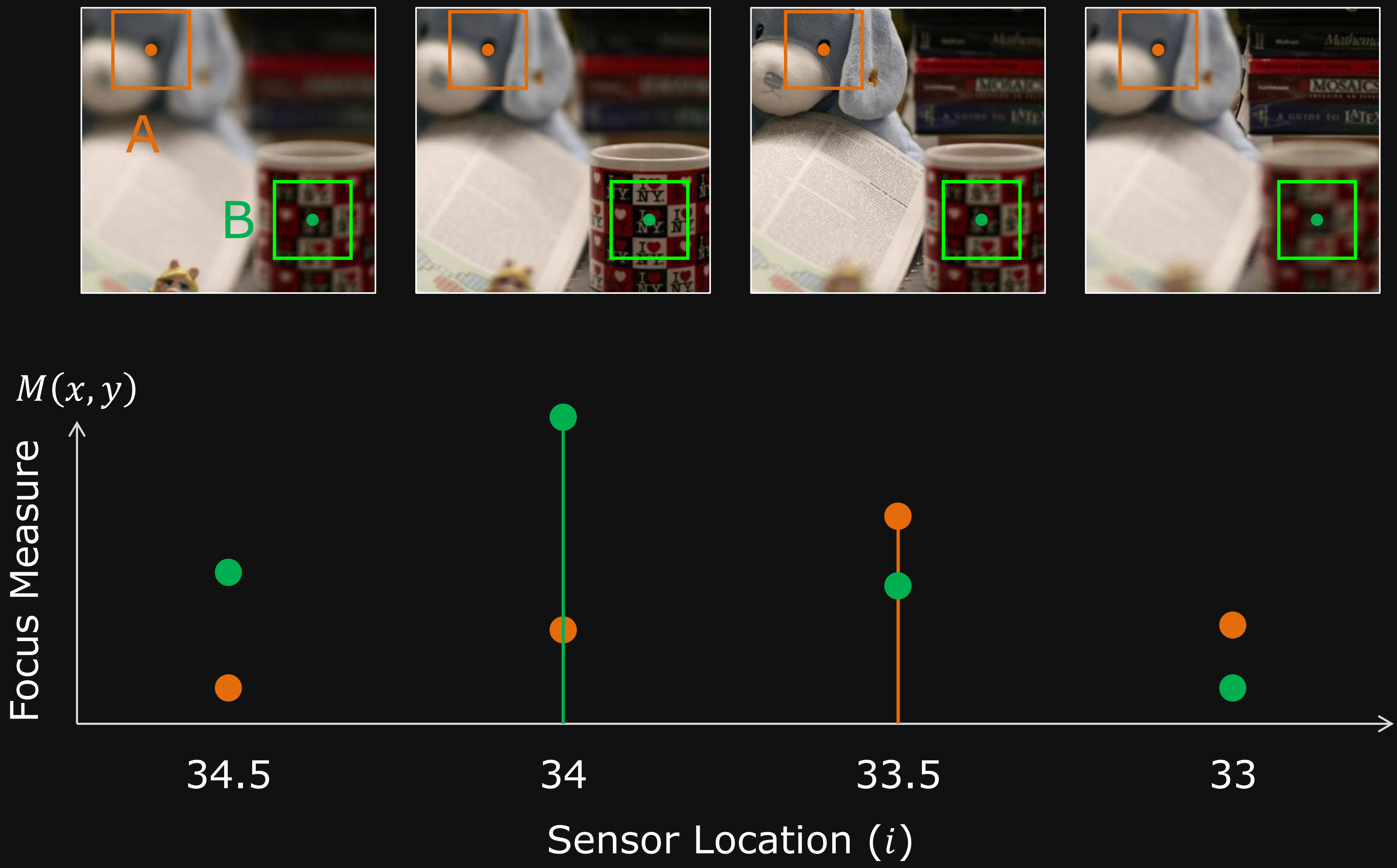
Use Laplacian as high pass filter: $\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$

Focus Measure: Sum of the Square of Laplacian responses within a small window.

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

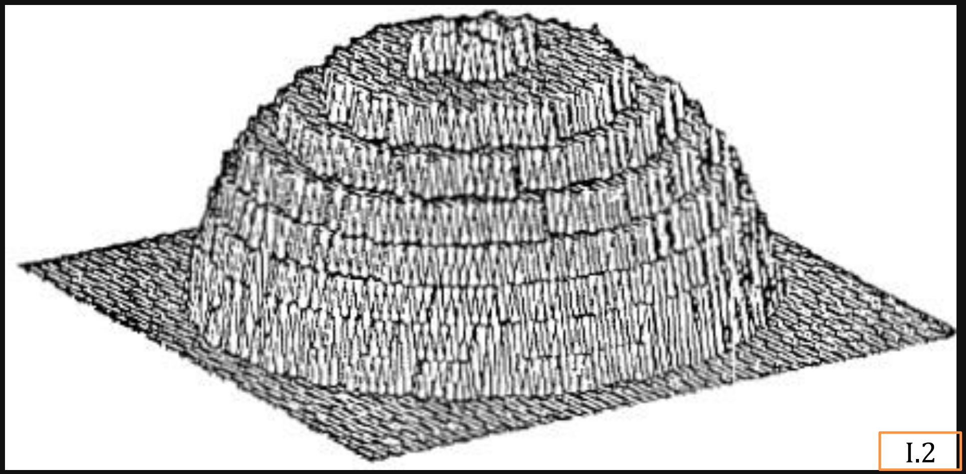


# Depth from Focus



# Depth from Focus: Result

Scene
(Metal ball with
rough surface)



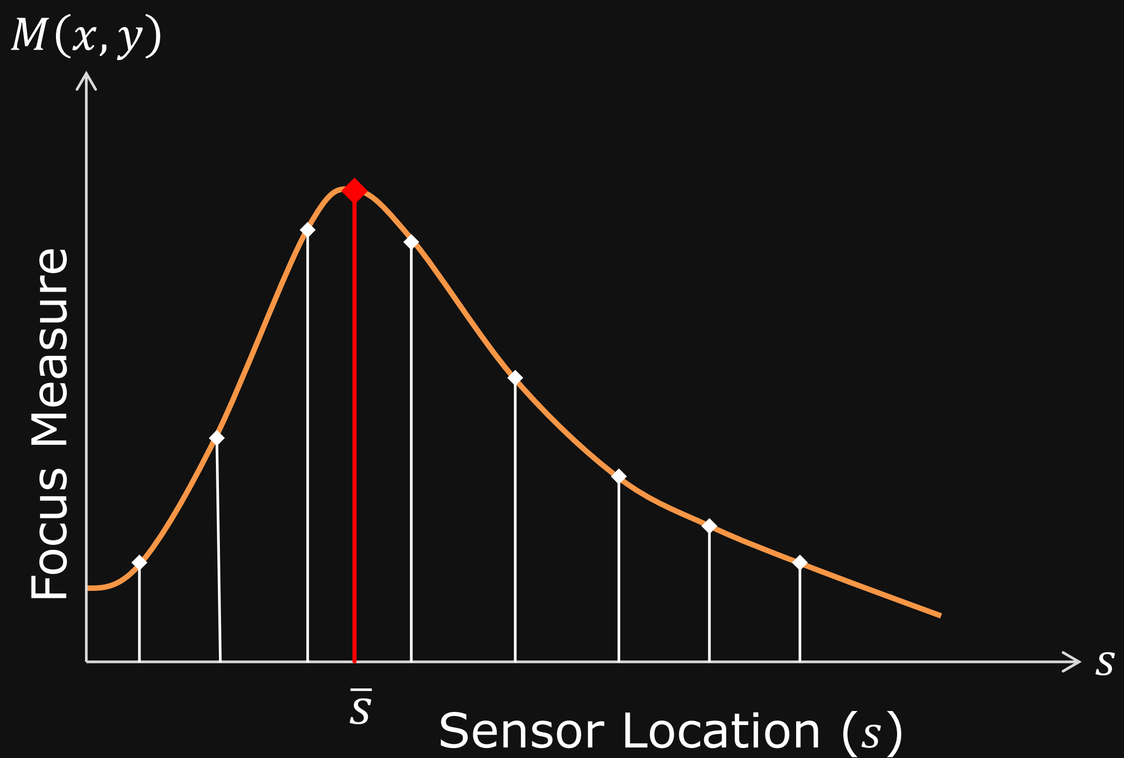

Shape obtained using
Depth from Focus

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

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

# Depth Estimation Using Gaussian Interpolation

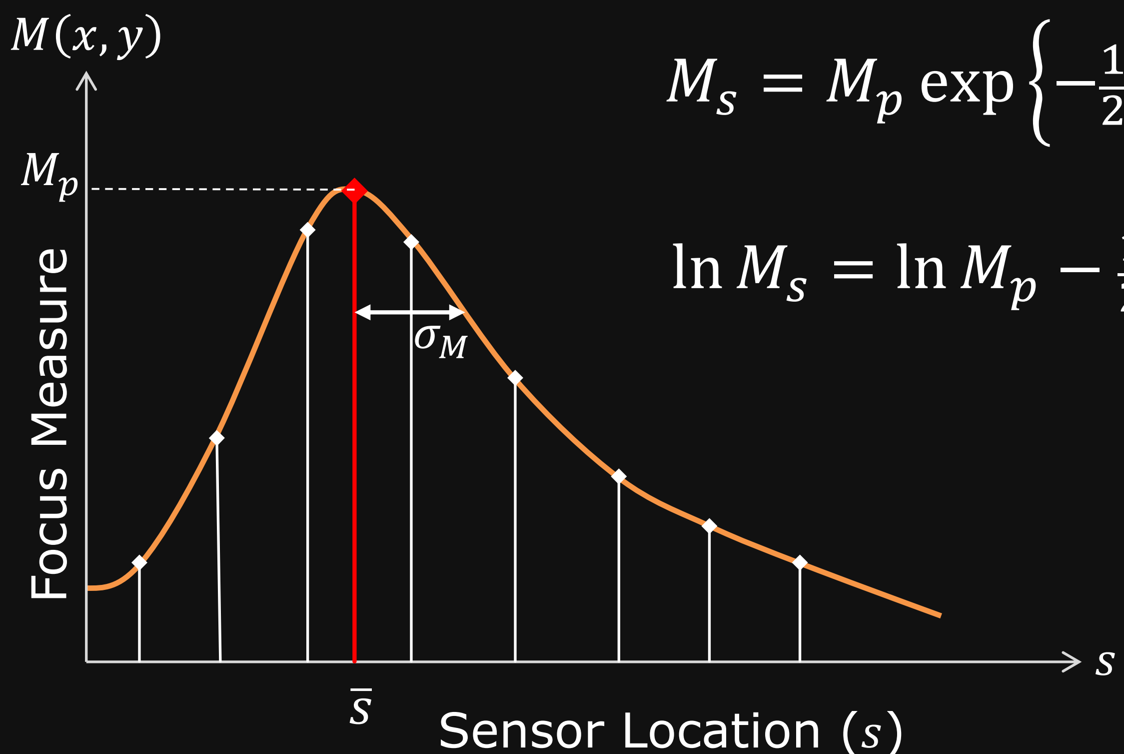
The Focus Measure Curve is a Gaussian-like function.




Mean of the Gaussian $\bar{s}$ may be used as the sensor location corresponding to the "best focus."

# Depth Estimation Using Gaussian Interpolation

The Focus Measure Curve is a Gaussian-like function.




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

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

How many samples do we need to estimate $\bar{s}$ ?

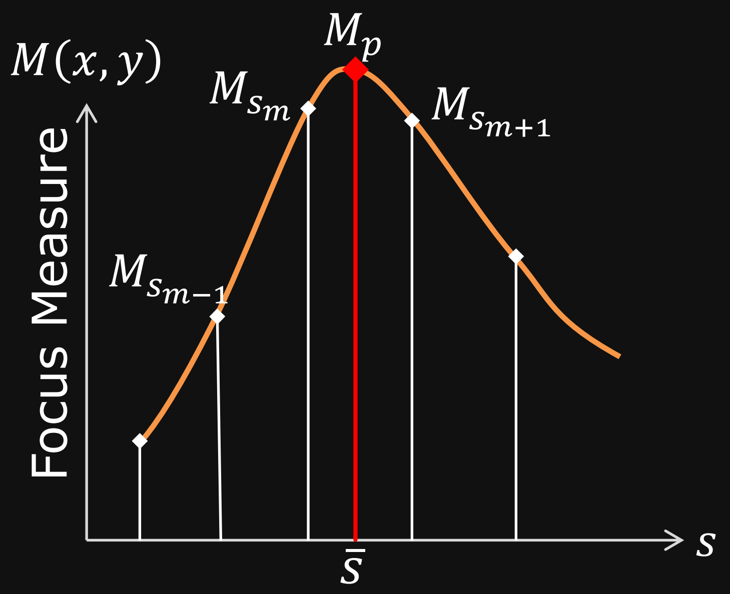
# Gaussian Interpolation

Only three samples are required to estimate $\bar{s}$ as there are only three unknowns ($\bar{s}$, $M_p$, $\sigma_M$)

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

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

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



# Depth Estimation Using Gaussian Interpolation

Solving for $\bar{s}$:

$$\bar{s} = \frac{\left(\ln M_{s_m} - \ln M_{s_{m+1}}\right)\left(s_m^2 - s_{m-1}^2\right)}{2\left(s_{m+1} - s_m\right)\left\{\left(\ln M_{s_m} - \ln M_{s_{m-1}}\right) + \left(\ln M_{s_m} - \ln M_{s_{m+1}}\right)\right\}}$$

$$- \frac{\left(\ln M_{s_m} - \ln M_{s_{m-1}}\right)\left(s_m^2 - s_{m+1}^2\right)}{2\left(s_{m+1} - s_m\right)\left\{\left(\ln M_{s_m} - \ln M_{s_{m-1}}\right) + \left(\ln M_{s_m} - \ln M_{s_{m+1}}\right)\right\}}$$

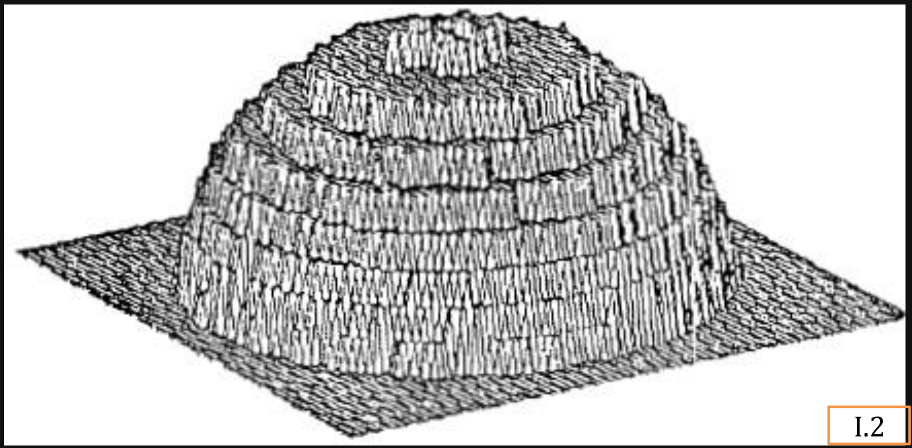
Finally, obtain scene depth using Gaussian Lens Law.

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

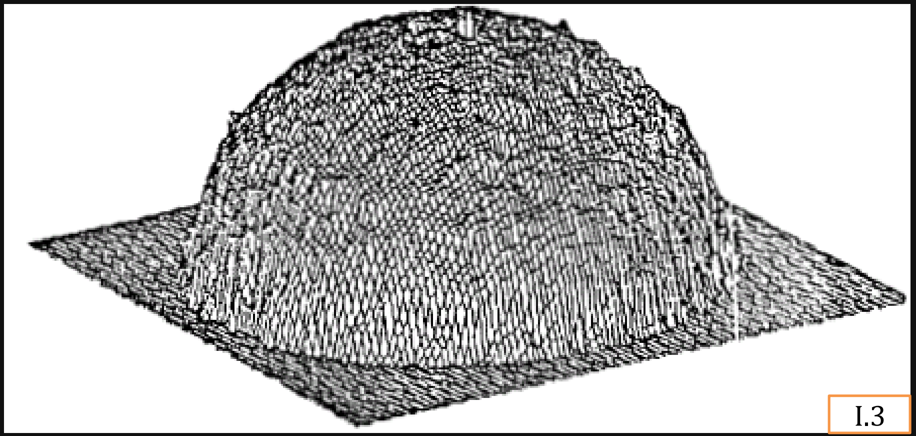
[Nayar 1994]

# Depth from Focus: Result

Scene
(Metal ball with
rough surface)

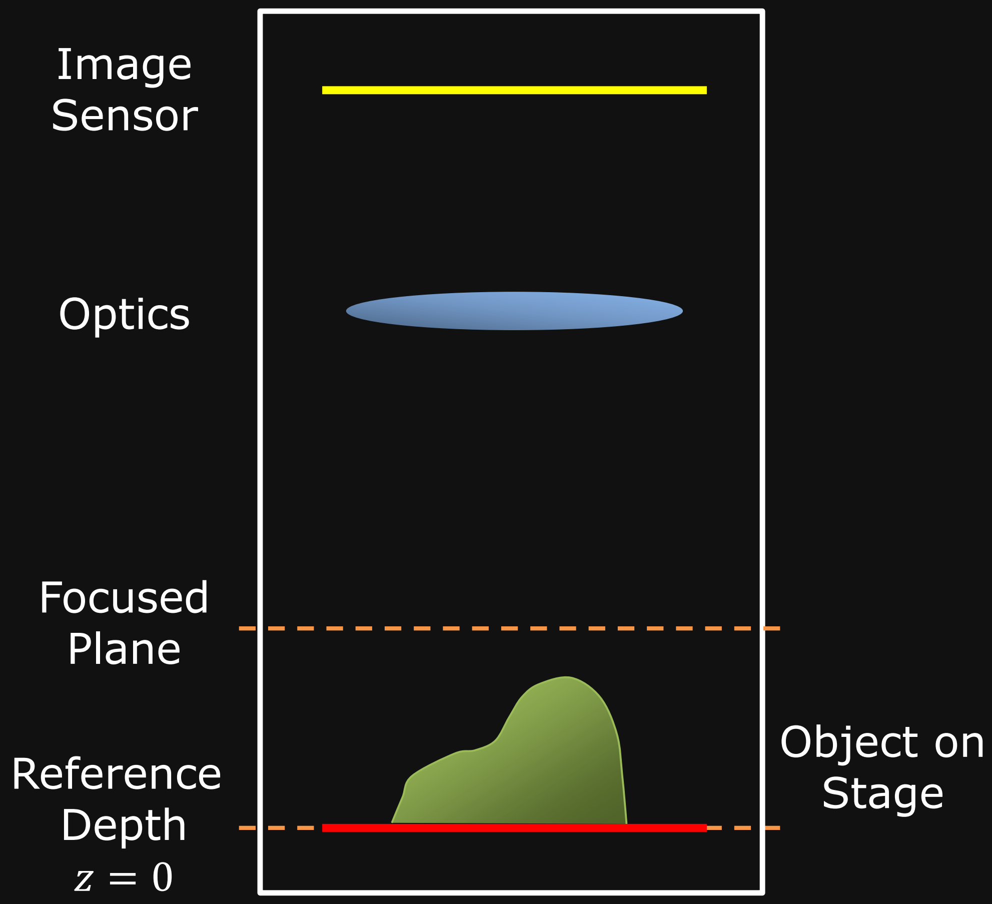



Depth without Gaussian Interpolation




Depth using Gaussian Interpolation

# A Depth from Focus (DFF) System

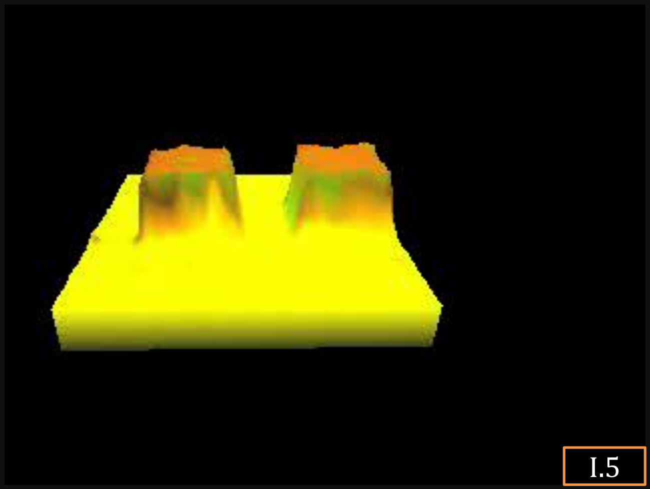



DFF using a Microscope

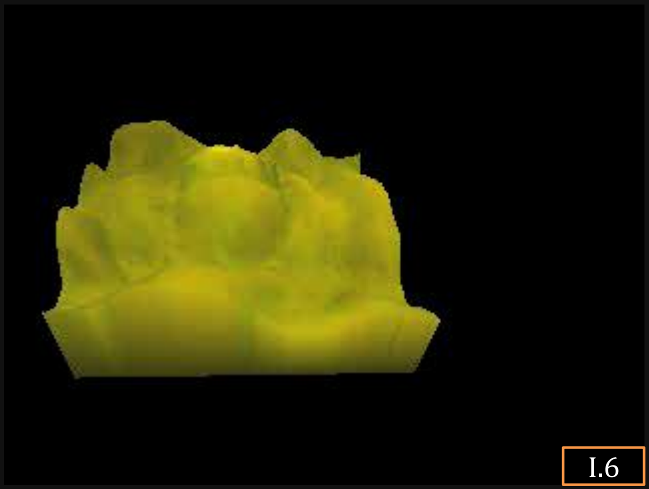

DFF System in action

[Nayar 1994]

# Depth from Focus System: Results



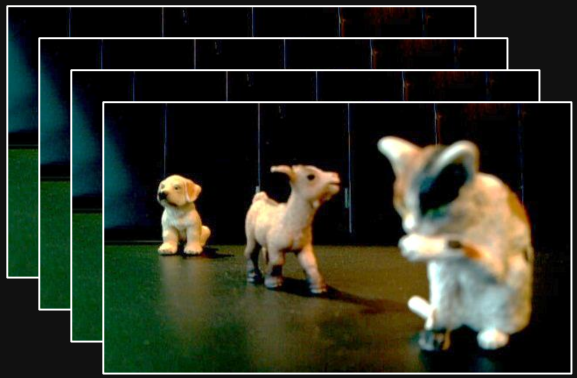

Silicon Wafer
(13 microns in height)




Leaf Structure
(30 microns in height)

[Nayar 1994]

# Homework 5: Depth from Focus



Focal Stack

Depth Map

All-Focus Image

<u>Fcam Programming</u>

1. Write a program to capture a focal stack

<u>MATLAB Programming</u>

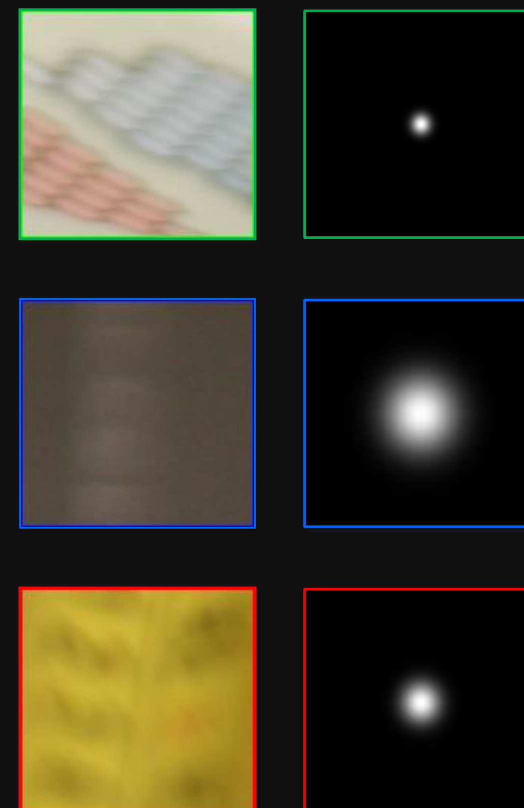
1. Calibrate the captured focal stack
2. Implement a focus measure and estimate depth
3. Compute an all-in-focus image

Due before class on Thursday 2/28

# Depth from Defocus (DFD)

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

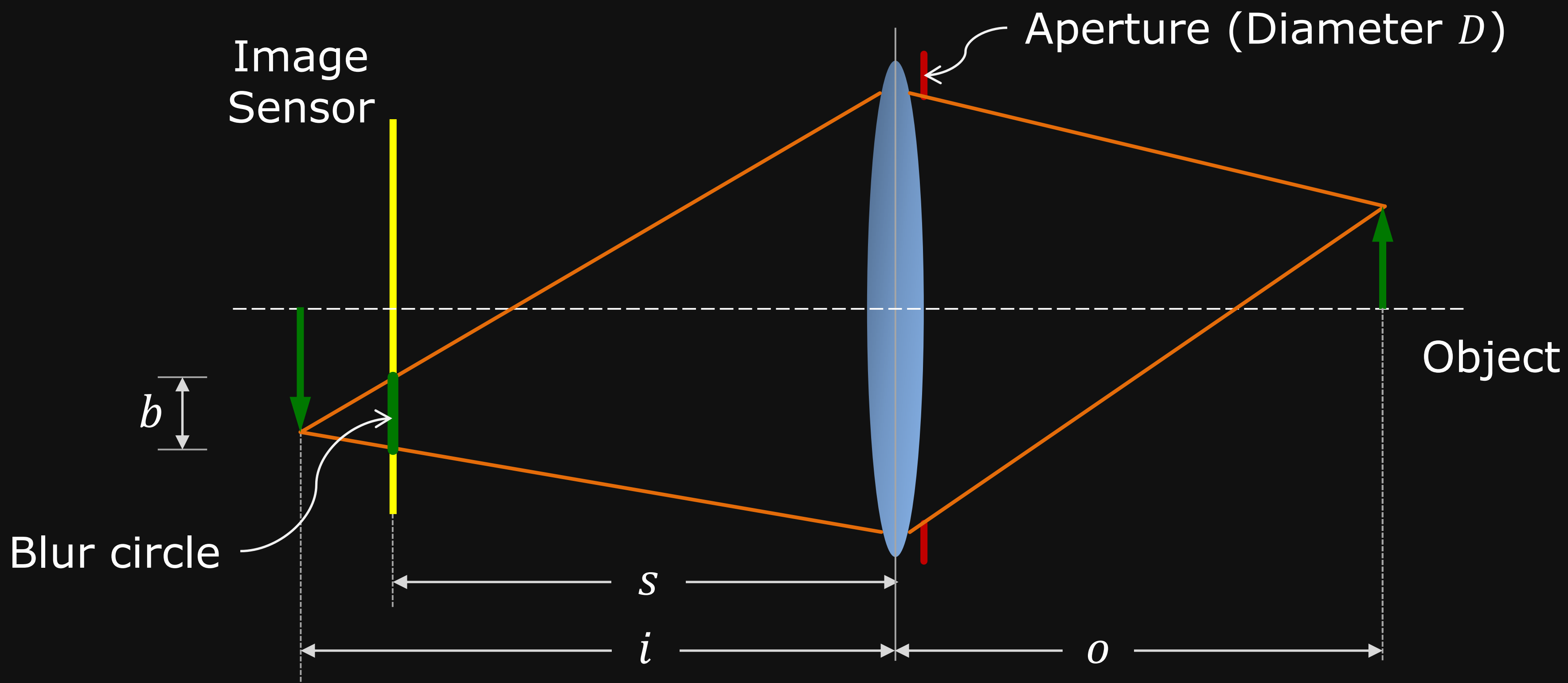
A Captured Image



Three Patches

PSFs

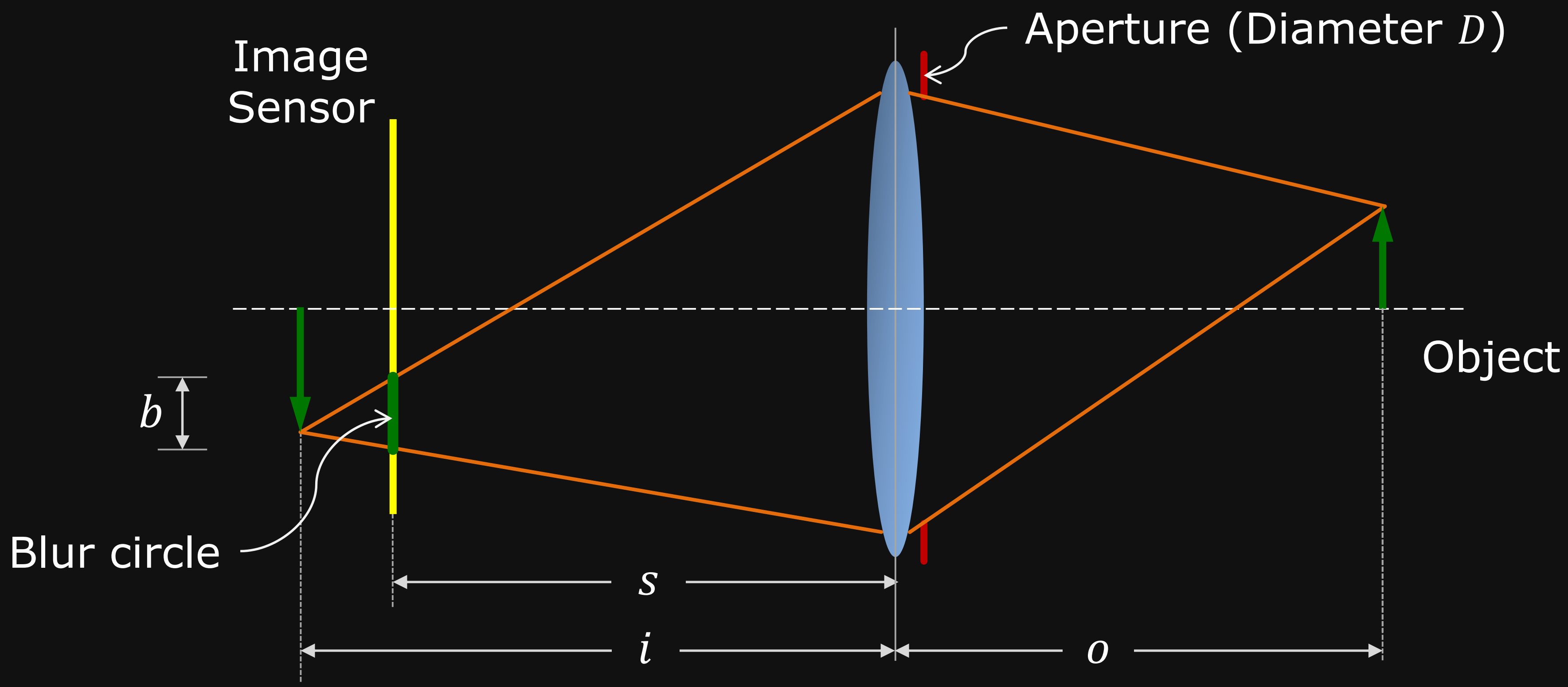
# Depth From Defocus




We know that: $$\frac{b}{D} = \frac{i-s}{i}$$ and $$\frac{1}{f} = \frac{1}{i} + \frac{1}{o}$$

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

# Depth From Defocus




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

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

$f/D$: F-Number

# Depth from Defocus: One Image?

Can we estimate blur size $b$ from a single image?


$*$


$=$

Scene $f(x, y)$

Defocus PSF $h(x, y)$

Image Patch $g(x, y)$

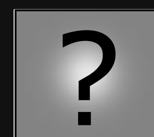
Impossible: One equation, Two unknowns

# Depth from Defocus: Two Images

What if we have two images with different defocus?

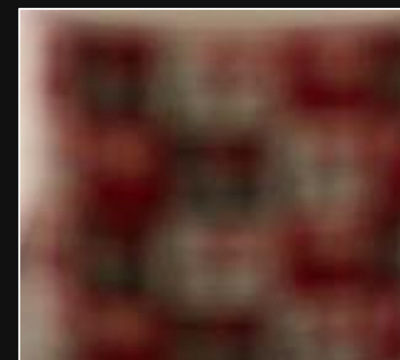

$f(x, y)$

$*$

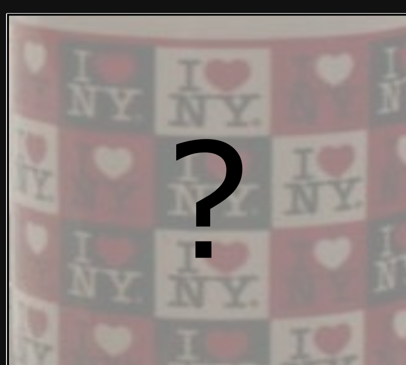



$h_{\sigma_1}(x, y)$

$=$

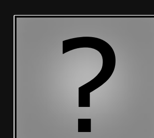


$g_1(x, y)$



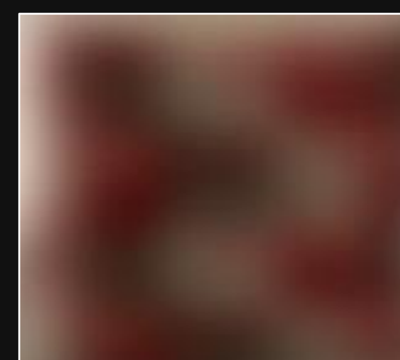

$f(x, y)$

$*$




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

$=$

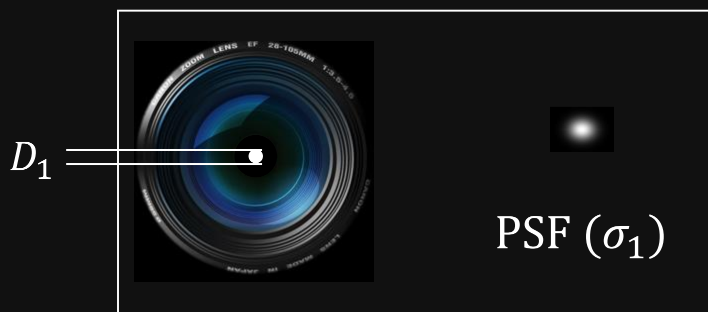


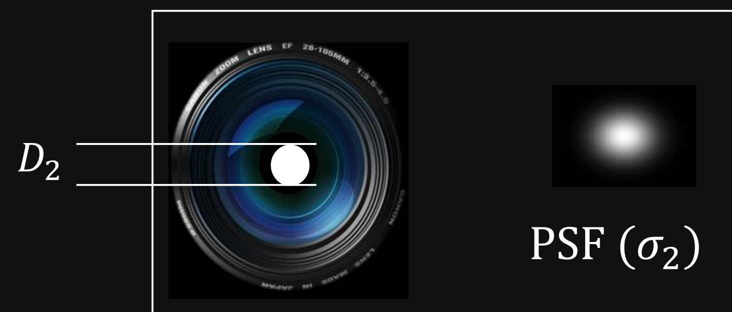
$g_2(x, y)$

Two equations, Three unknowns

# The Third Equation

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




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

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

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

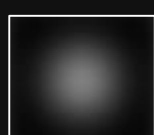
# Depth from Defocus

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 $F(u,v)$

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

$$\Rightarrow \frac{\|G_1(u,v)\|}{\|G_2(u,v)\|} = \frac{\left\|H_{\sigma_1}(u,v)\right\|}{\left\|H_{\sigma_2}(u,v)\right\|}$$

Substitute for $H_\sigma(u,v)$

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

Take natural logarithm on both sides and rearrange:

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

Sensitive to noise

[Pentland 1987]

# A Naïve DFD Algorithm

Estimating robust $(\sigma_1^2 - \sigma_2^2)$:

$$\sigma_1^2 - \sigma_2^2 = \frac{1}{N} \sum_{(u_i, v_j)} \frac{\ln\|G_2(u_i, v_j)\| - \ln\|G_1(u_i, v_j)\|}{2\pi^2(u_i{}^2 + v_j{}^2)} \qquad \text{------ (A)}$$

Use only frequencies $(u_i, v_j)$ such that: $u_i > 0$, $v_\mathrm{j} > 0$,
$G_1(u_i, v_j) \gg 0$, $G_2(u_i, v_j) \gg 0$

We also know that: $\sigma_1 / \sigma_2 = D_1 / D_2$ ------ (B)

Solve (A) and (B) get $\sigma_1$ and $\sigma_2$.

Size of blur circle: $b_1 = 2\sigma_1$

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

[Pentland 1987]

# Reconstruction-Based DFD Algorithm

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

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

We need to find $(\sigma_1, \sigma_2, f(x,y))$ that minimize the Reconstruction Error:

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

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

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

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

# Computing Absolute Depth

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

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

Size of blur circle: $b_1 = 2\sigma_1$

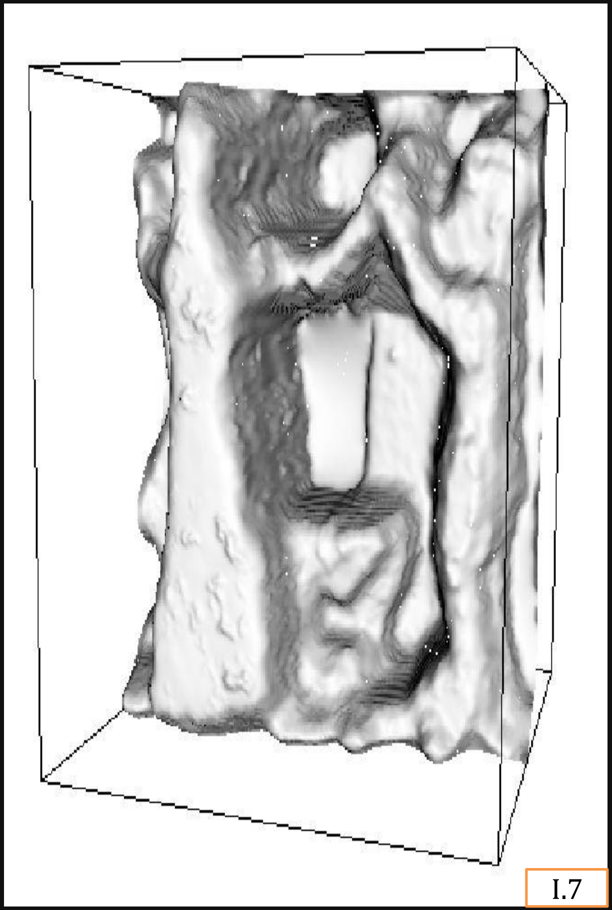
Object distance:

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

# Depth from Defocus: Result

Focused at the far end

Focused at the near end




Estimated 3D shape

# Capturing Defocused Images

## Method 1: Change Aperture

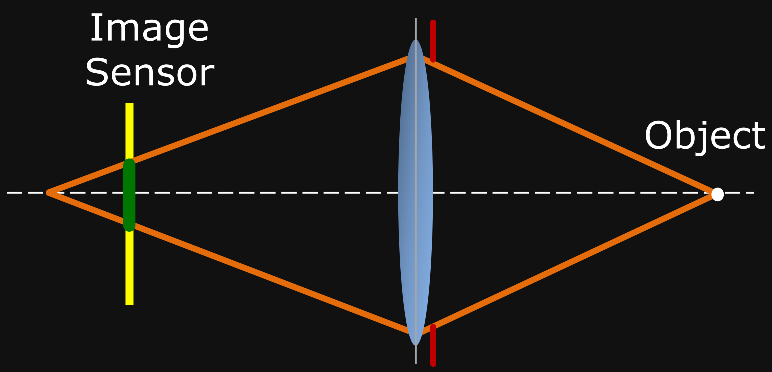



Image 1

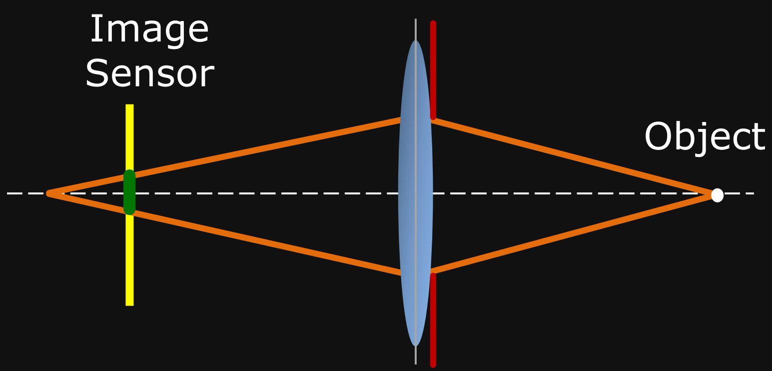



Image 2

## Method 2: Move Sensor

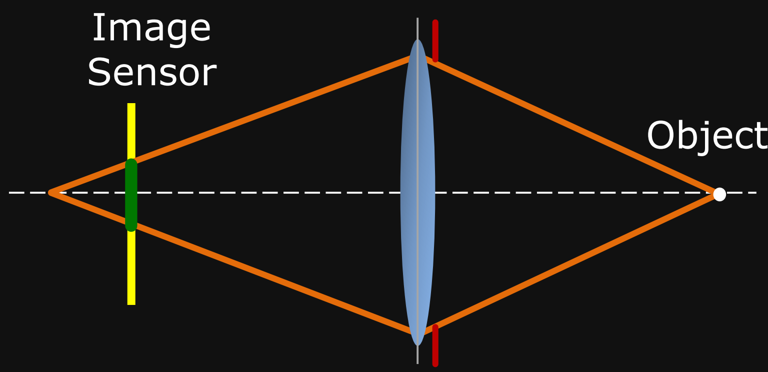



Image 1

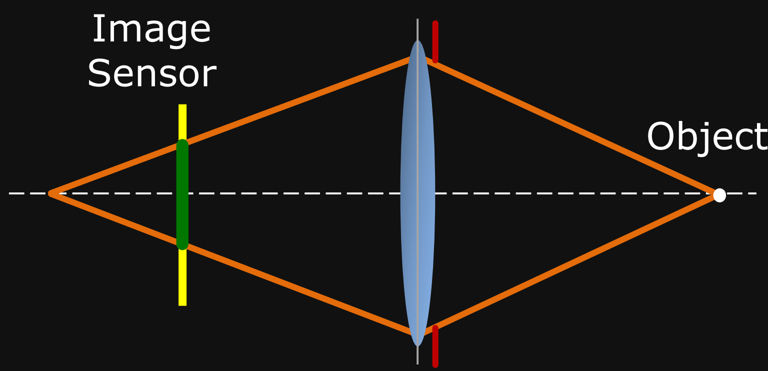
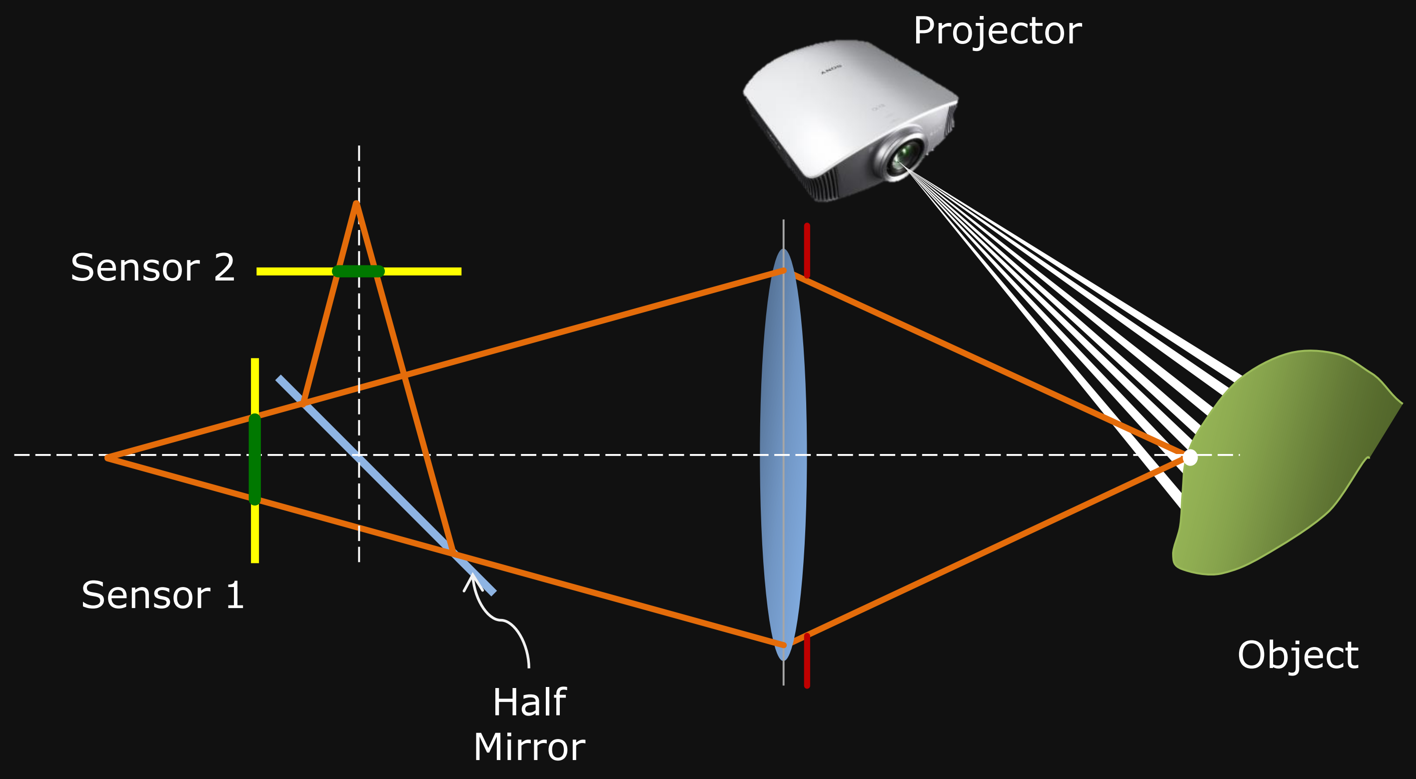


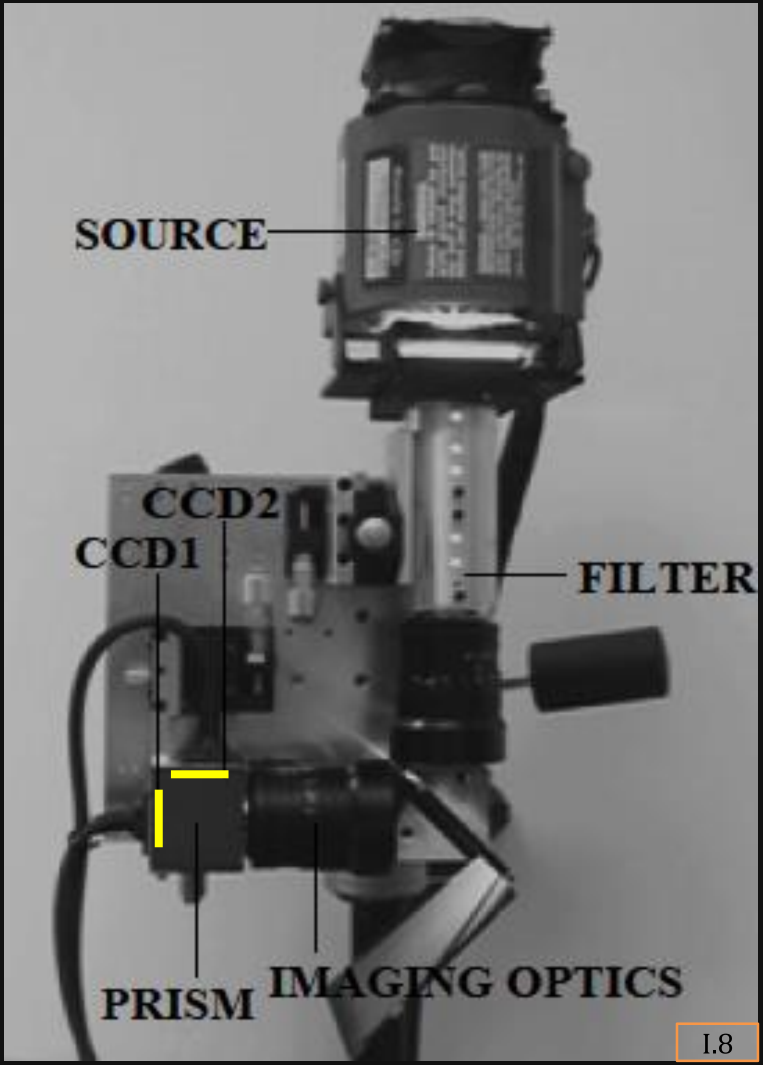

Image 2

# Implementing a DFD System




Two-Sensor DFD System with Active Illumination

# A Depth from Defocus (DFD) System




[Nayar 1996]

# A Depth from Defocus (DFD) System



Real time Depth from Defocus System

[Nayar 1996]

# References: Papers


[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

[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.

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

# Image Credits

I.1 http://s3.media.squarespace.com/production/425896/5919186/wp-content/uploads/2009/03/shutterstation-dof-93.jpg

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/