Image Formation

Image: Projection of 3D scene onto 2D plane. We need to understand the geometric and photometric relation between the scene and its image.

Topics:

(1) Pinhole and Perspective Projection

(2) Image formation using Lenses

(3) Lens Related Issues

(4) Wide Angle Cameras

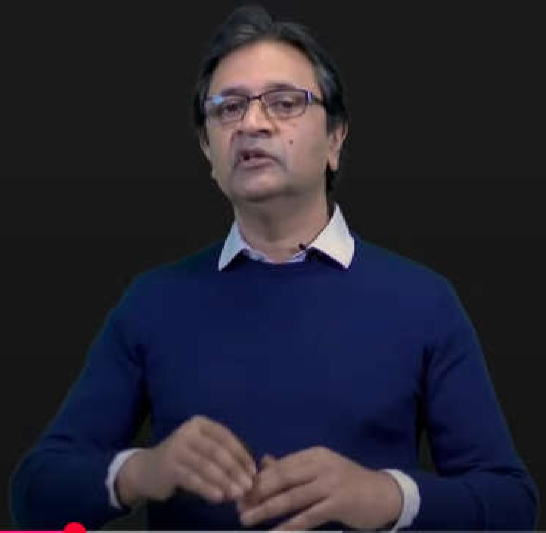
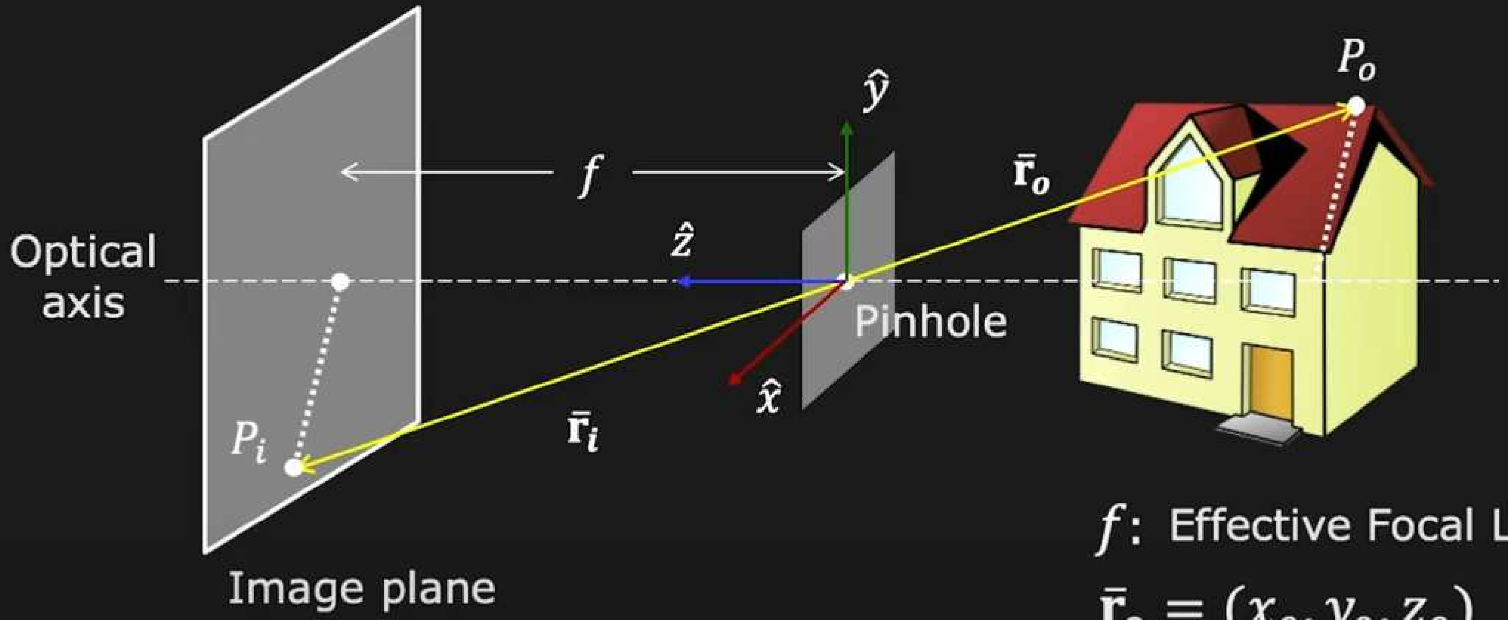


Image Formation

Is an image being formed on the screen?

Perspective Imaging with Pinhole



f: Effective Focal Length

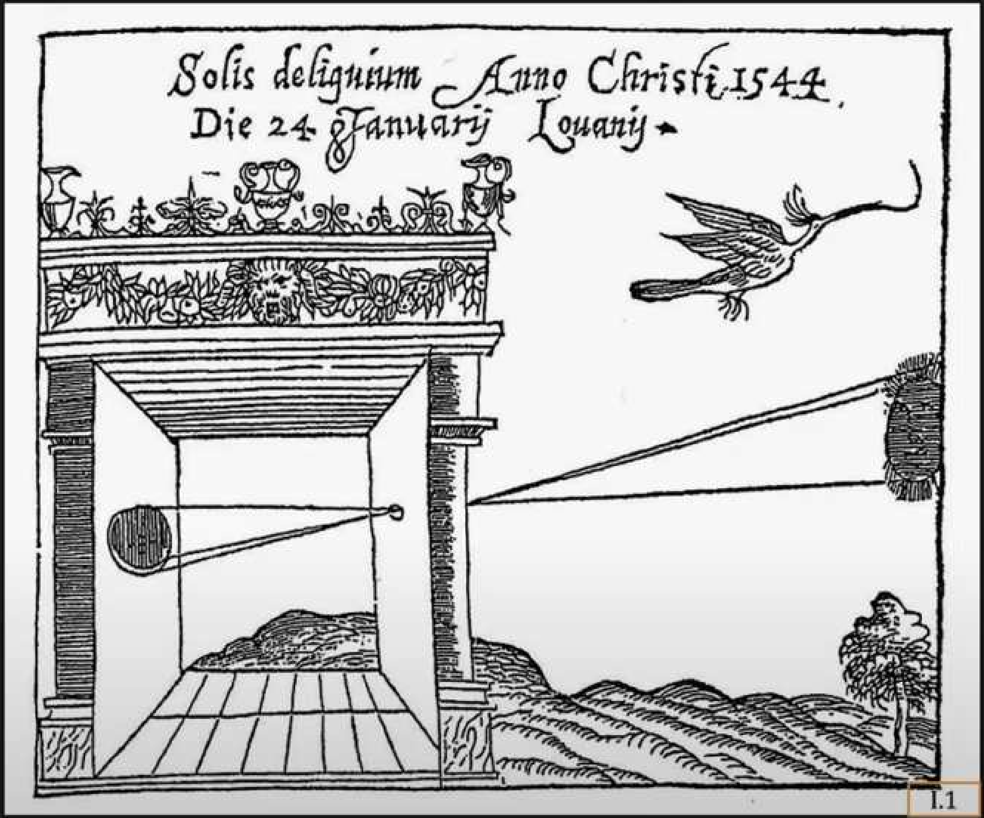
$$\bar{\mathbf{r}}_o = (x_o, y_o, z_o)$$

$$\bar{\mathbf{r}}_i = (x_i, y_i, f)$$

Using similar triangles:

$$\boxed{\frac{\bar{\mathbf{r}}_i}{f} = \frac{\bar{\mathbf{r}}_o}{z_o}} \rightarrow \boxed{\frac{x_i}{f} = \frac{x_o}{z_o}, \quad \frac{y_i}{f} = \frac{y_o}{z_o}}$$

Camera Obscura



"Dark Chamber"

Pinhole and Perspective Projection | Image Formation

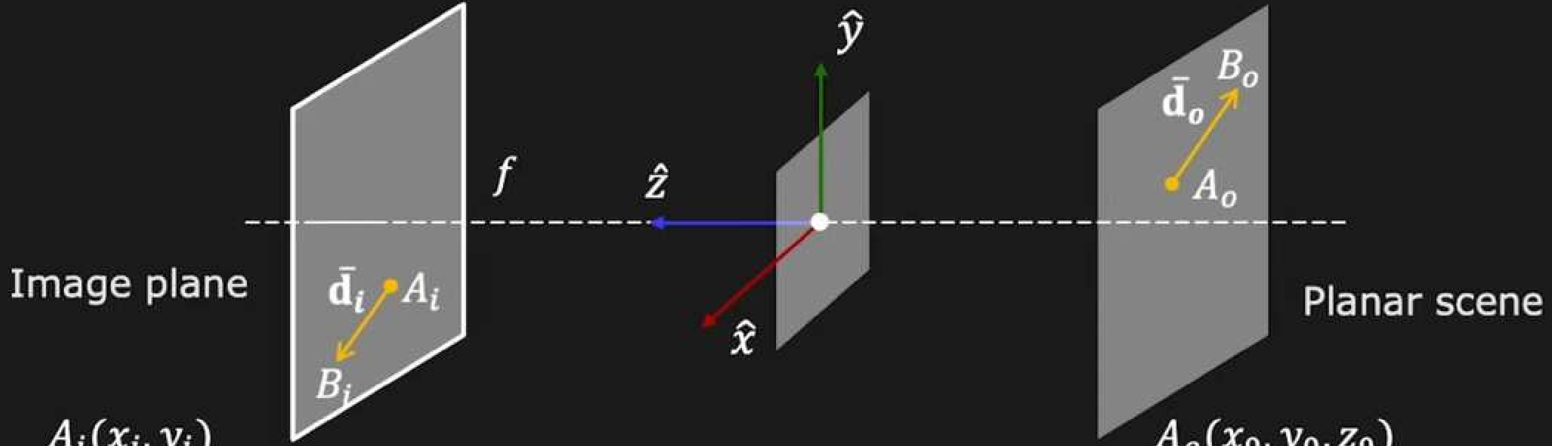
Watch later Share

Pinhole Eye of *Nautilus pompilius*

I.2

I.3

Image Magnification



$A_i(x_i, y_i)$

$B_i(x_i + \delta x_i, y_i + \delta y_i)$

$A_o(x_0, y_0, z_0)$

$B_o(x_0 + \delta x_o, y_0 + \delta y_o, z_0)$

Magnification: $$|m| = \frac{\|\bar{\mathbf{d}}_i\|}{\|\bar{\mathbf{d}}_o\|} = \sqrt{{\delta x_i}^2 + {\delta y_i}^2} \Big/ \sqrt{{\delta x_o}^2 + {\delta y_o}^2}$$

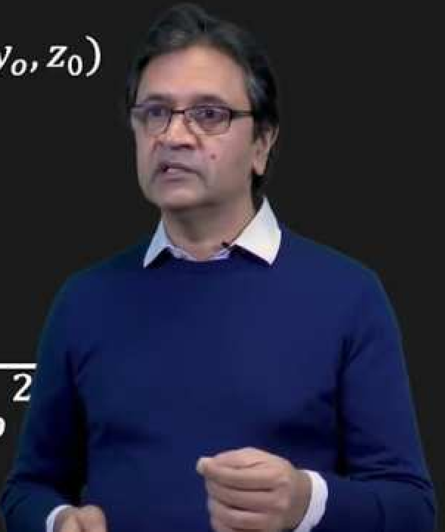
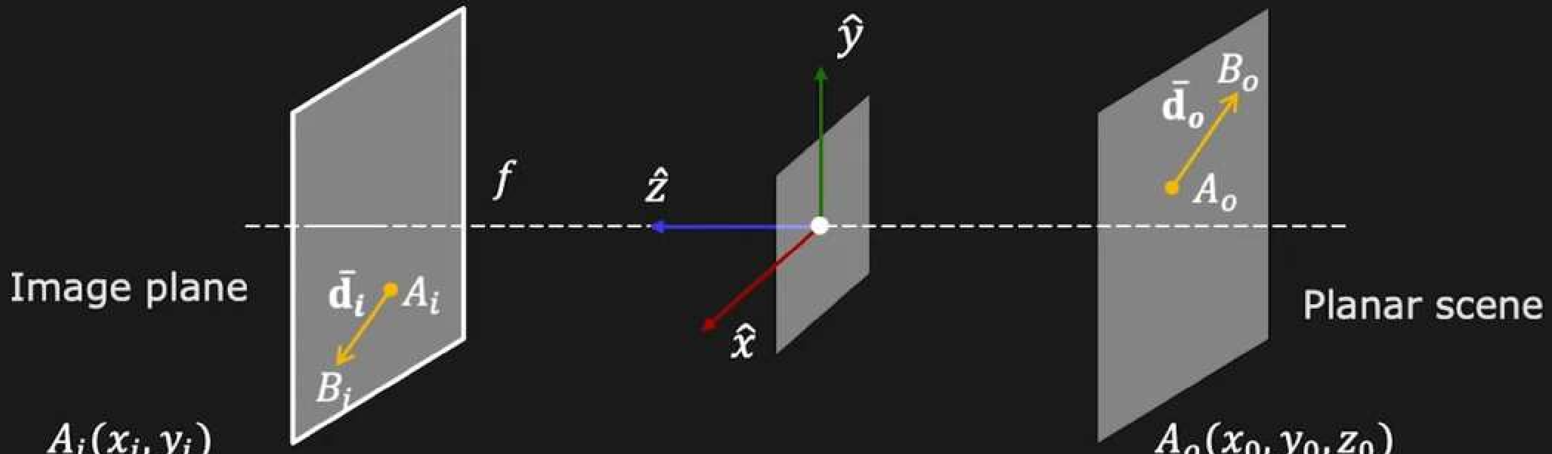


Image Magnification



$A_i(x_i, y_i)$

$B_i(x_i + \delta x_i, y_i + \delta y_i)$

$A_o(x_0, y_0, z_0)$

$B_o(x_0 + \delta x_o, y_0 + \delta y_o, z_0)$

From Perspective Projection:

$$\frac{x_i}{f} = \frac{x_o}{z_o} \quad \text{and} \quad \frac{y_i}{f} = \frac{y_o}{z_o} \quad \text{-------------------} \quad (A)$$

$$\frac{x_i + \delta x_i}{f} = \frac{x_o + \delta x_o}{z_o} \quad \text{and} \quad \frac{y_i + \delta y_i}{f} = \frac{y_o + \delta y_o}{z_o} \quad \text{---------} \quad (B)$$

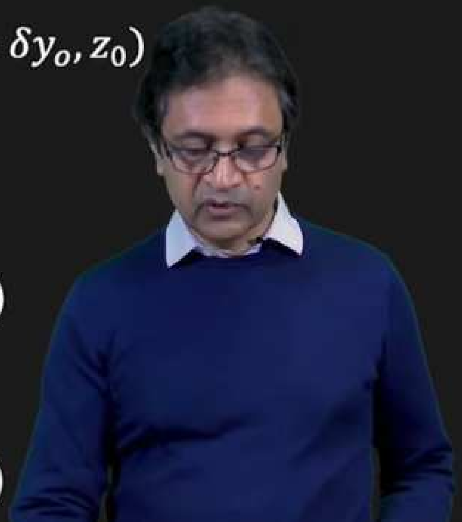


Image Magnification

From (A) and (B) we get:

$$\frac{\delta x_i}{f} = \frac{\delta x_o}{z_o} \quad \text{and} \quad \frac{\delta y_i}{f} = \frac{\delta y_o}{z_o}$$

Magnification:

$$|m| = \frac{\|\bar{\mathbf{d}}_i\|}{\|\bar{\mathbf{d}}_o\|} = \sqrt{\delta x_i{}^2 + \delta y_i{}^2} \Big/ \sqrt{\delta x_o{}^2 + \delta y_o{}^2} = \left|\frac{f}{z_o}\right|$$

Image Magnification

$$\boxed{m = \frac{f}{z_o}}$$ Image size inversely proportional to depth

Image Magnification

$$\boxed{m = \frac{f}{z_o}}$$

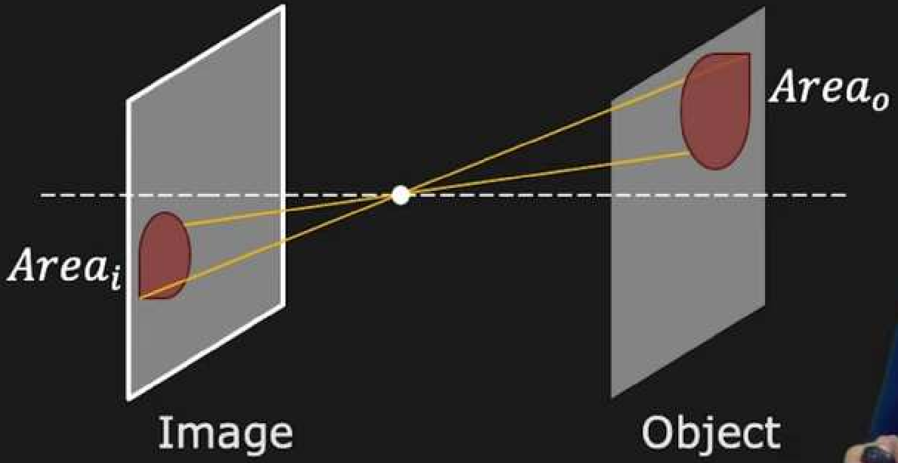
Image size inversely proportional to depth

Image Magnification

Remarks:

- m can be assumed to be constant if the range of scene depth Δz is much smaller than the average scene depth $\tilde{z}$

- $\dfrac{Area_i}{Area_o} = m^2$



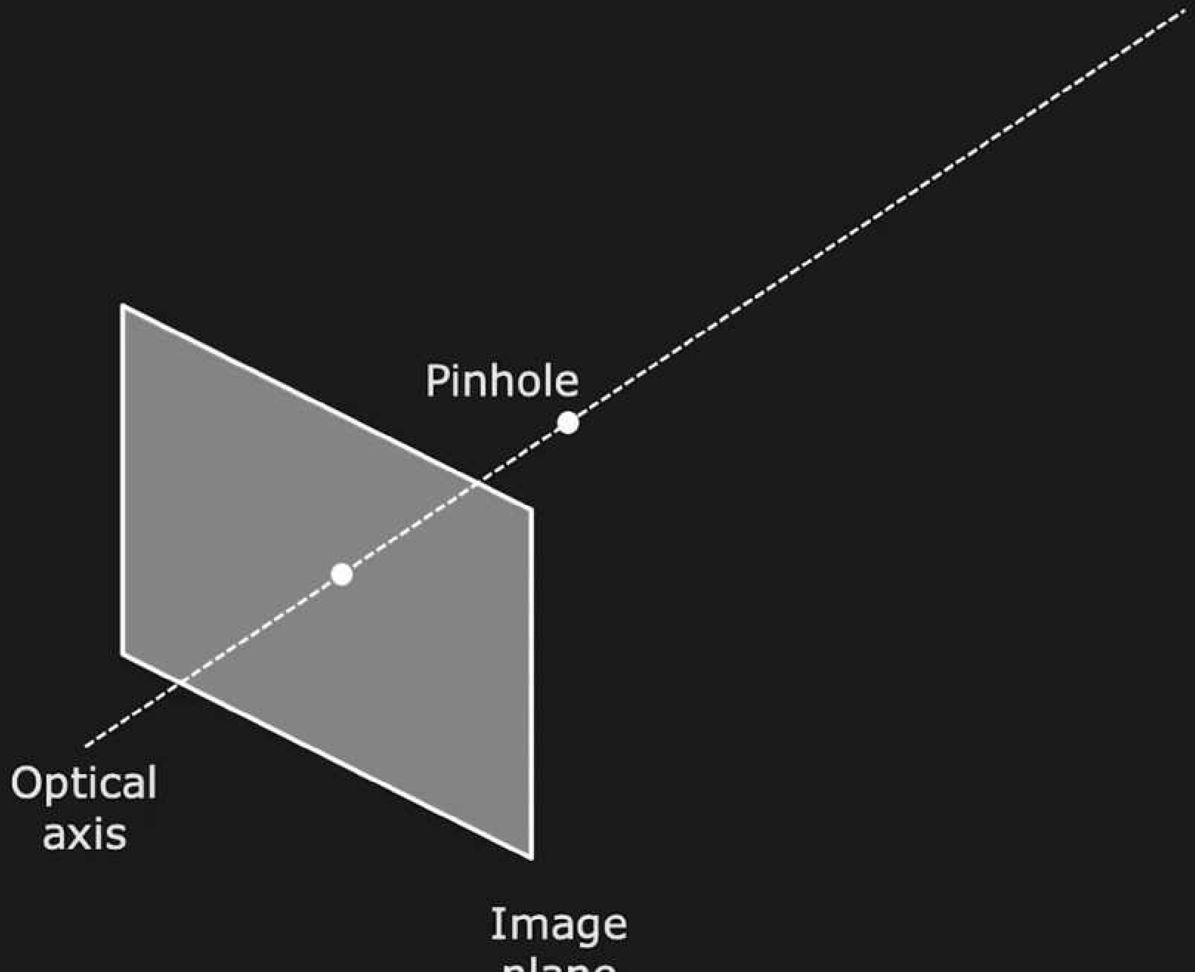
Vanishing Point

Parallel straight lines converge at a single image poi

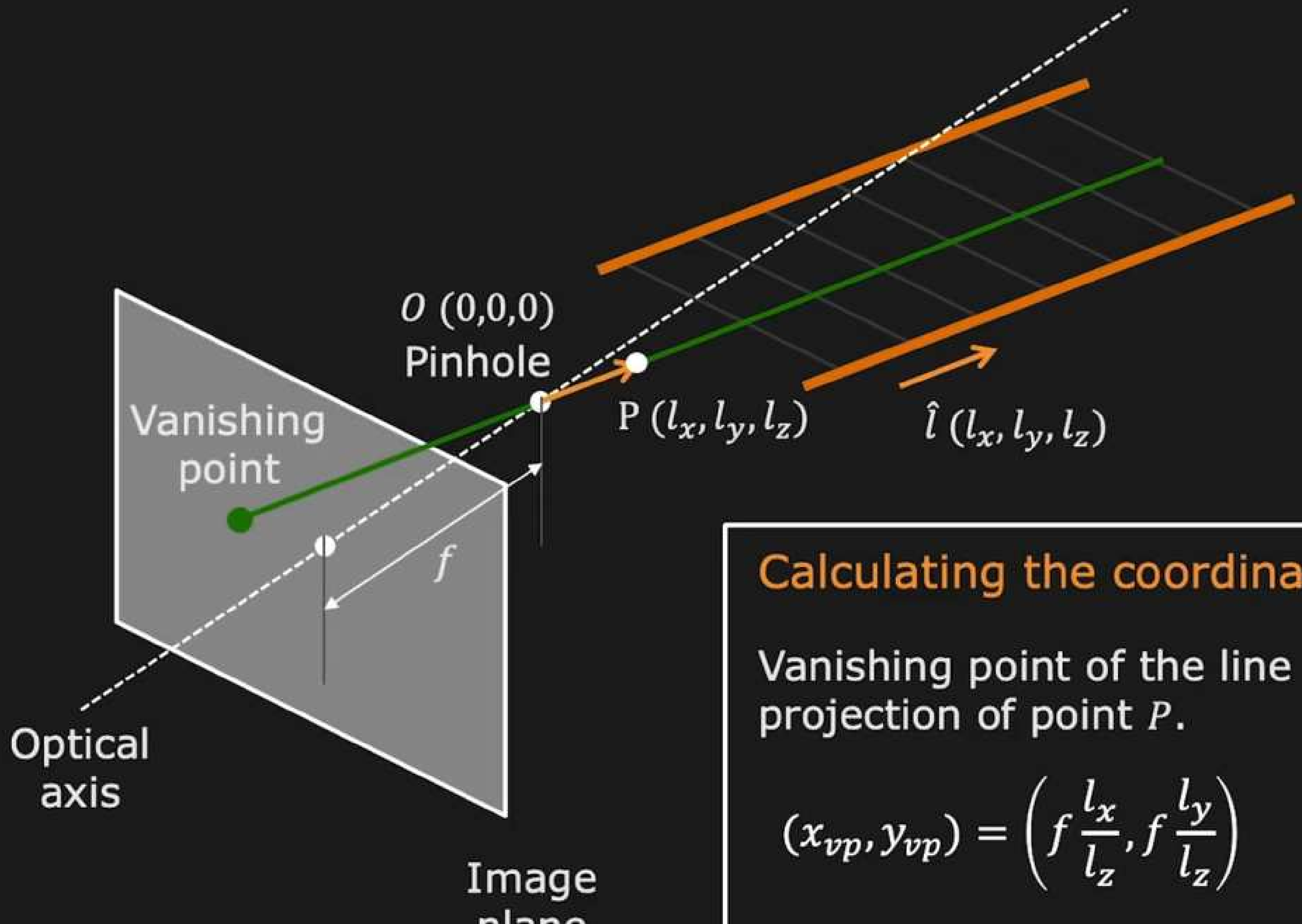
Vanishing Point

Location of Vanishing Point depends on the orientation of parallel straight lines.

Finding the Vanishing Point



Finding Vanishing Point



Calculating the coordinates:

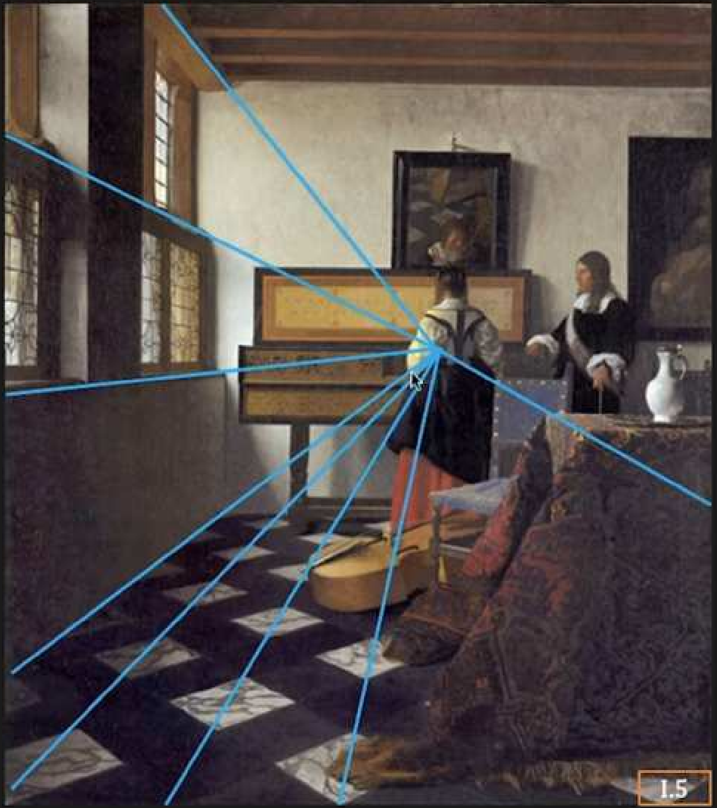
Vanishing point of the line is the projection of point P.

$$(x_{vp}, y_{vp}) = \left(f\frac{l_x}{l_z}, f\frac{l_y}{l_z}\right)$$

Use of Vanishing Point in Art

The Music Lesson, Johannes Vermeer, c. 1662-1664

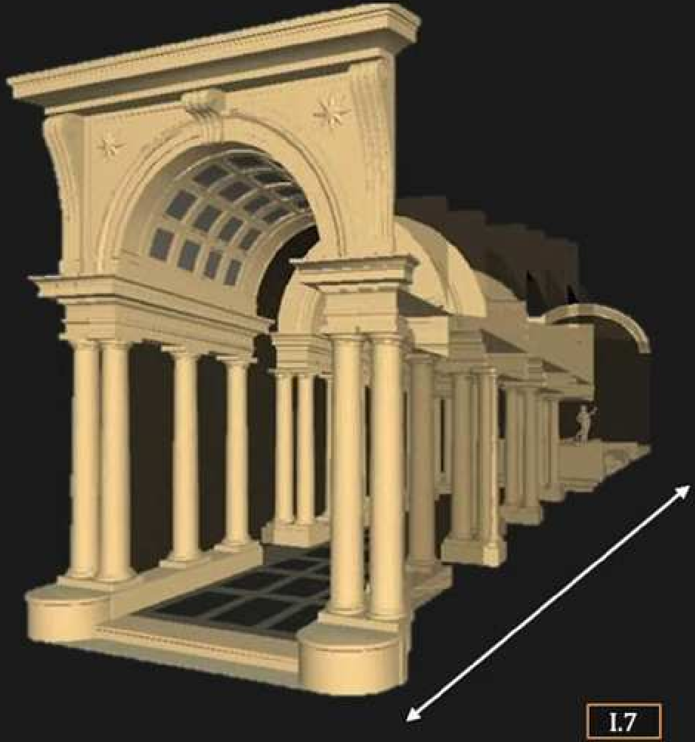
Use of Vanishing Point in Art



The Music Lesson, Johannes Vermeer, c. 1662-1664

False Perspective

Depth appears to be ~155 feet

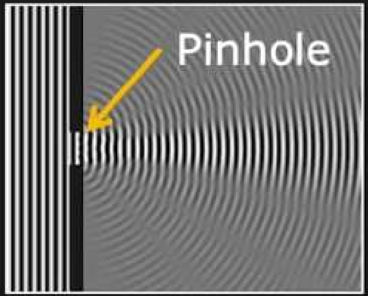


Depth is actually ~30 feet

Galleria Spada, Francesco Borromini, 1652

What is the Ideal Pinhole Size?

2 mm | 1 mm | 0.6 mm | **0.35 mm** | 0.15 mm | 0.07 mm

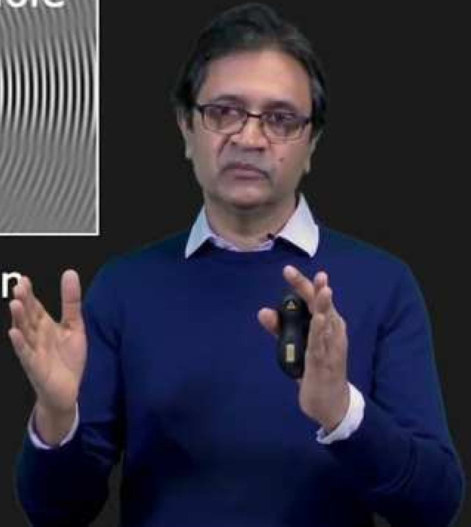


Diffraction

The pinhole must be tiny,
but if it's too tiny it will cause diffraction.

Ideal pinhole diameter: $\boxed{d \approx 2\sqrt{f\lambda}}$

f: effective focal length
λ: wavelength



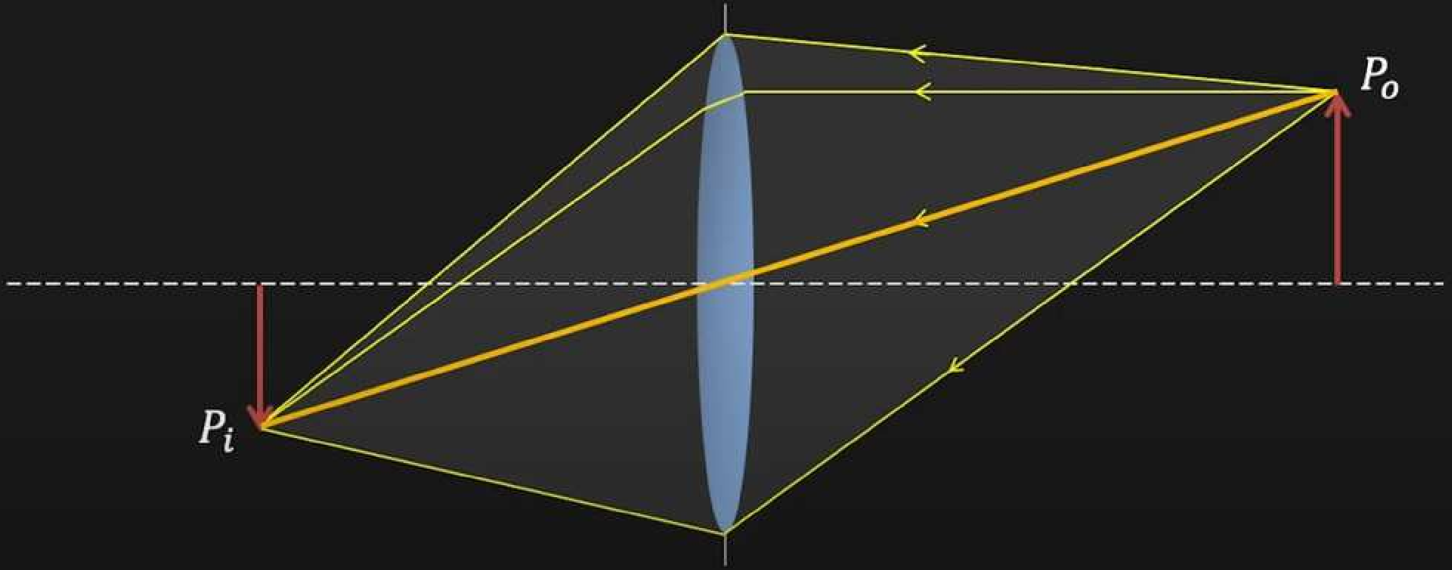
What about Exposure Time?

Pinholes pass less light and hence require long exposures to capture bright images.

$f = 73\,mm,\ d = 0.2\,mm,$
Exposure, $T = 12\,s$

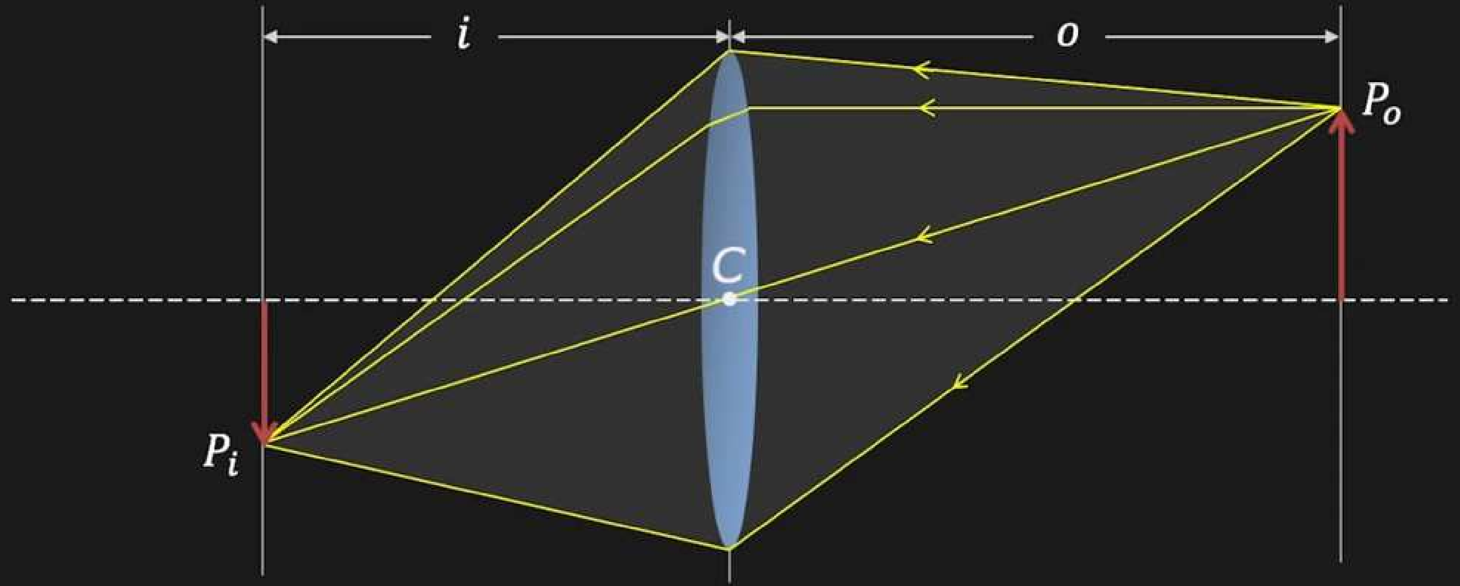
Lenses

Same projection as pinhole, but gather more light!



Focal length (f) determines the lens' bending power

Gaussian Lens (Thin Lens) Law



f: focal length

i: image distance

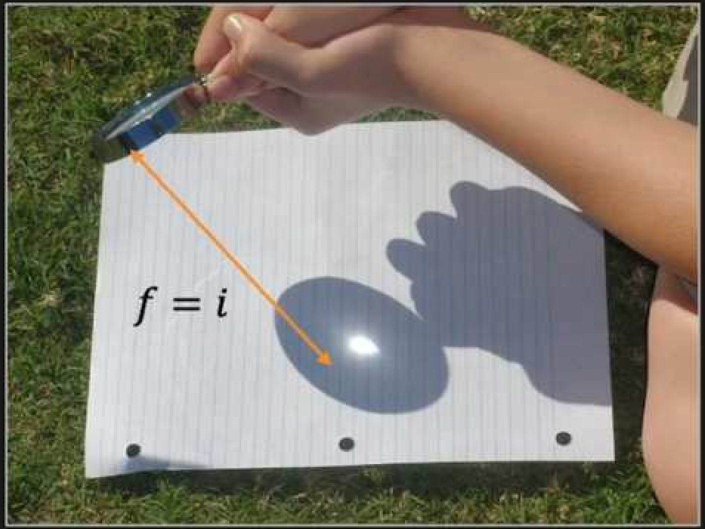
o: object distance

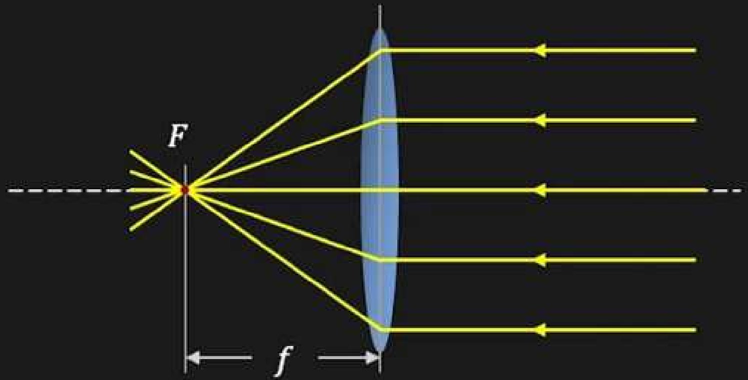
$$\boxed{\frac{1}{i}+\frac{1}{o}=\frac{1}{f}}$$

Example: If $f = 50mm$ & $o = 300mm$, then image distance $i = 60mm$

How to Find the Focal Length?

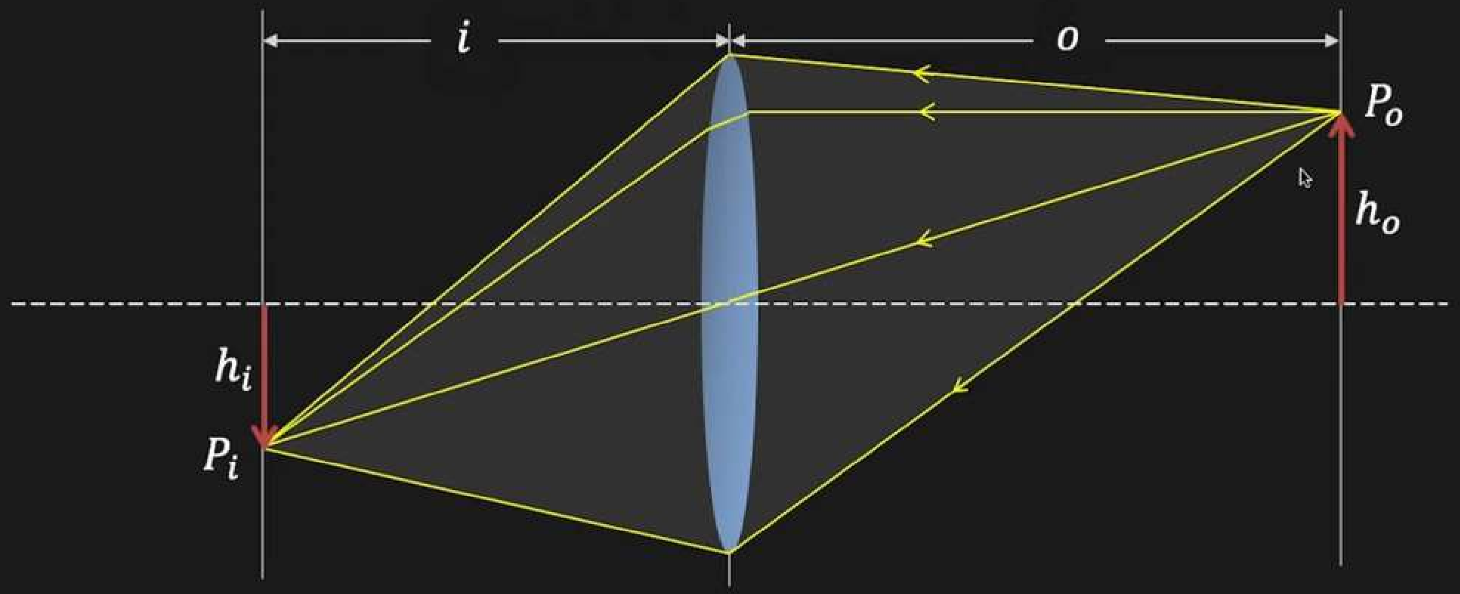
$$\frac{1}{i} + \frac{1}{o} = \frac{1}{f} \quad \Rightarrow \quad \text{If } o = \infty \text{, then } f = i$$





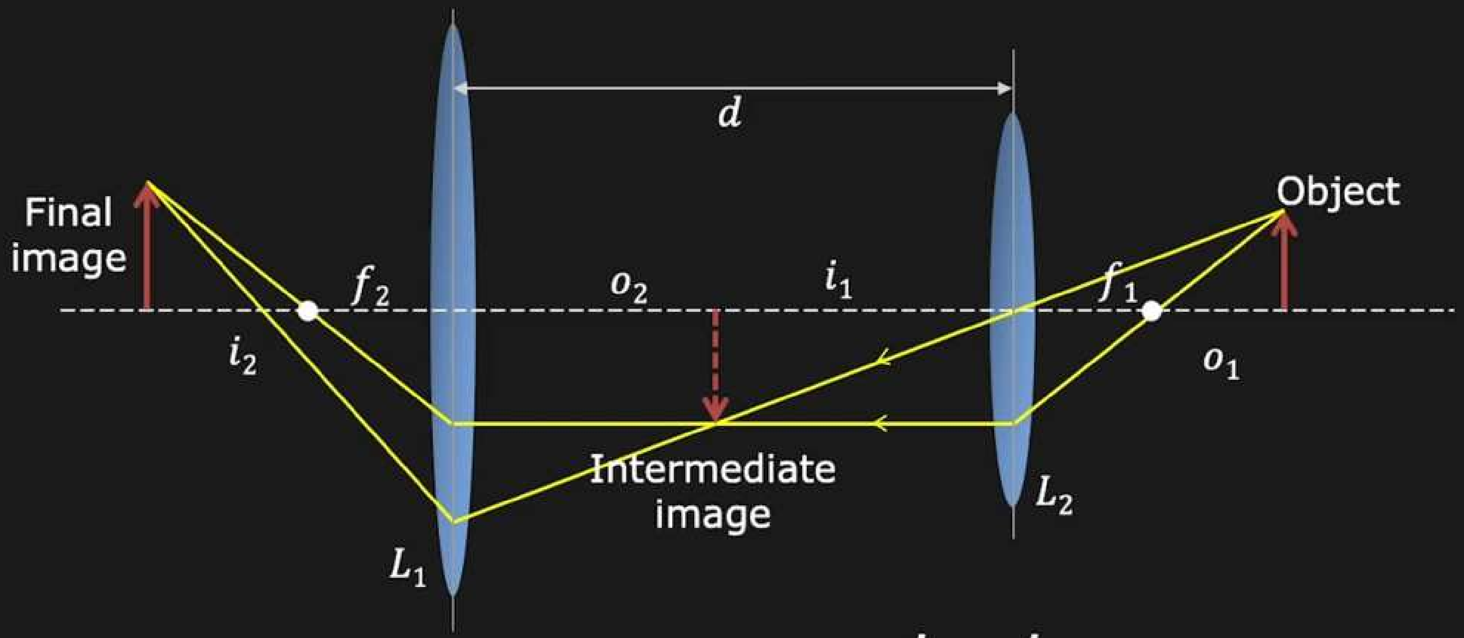
Focal length: Distance at which incoming rays that are parallel to the optical axis converge.

Image Magnification



Magnification: $m = \frac{h_i}{h_o}$

Two Lens System

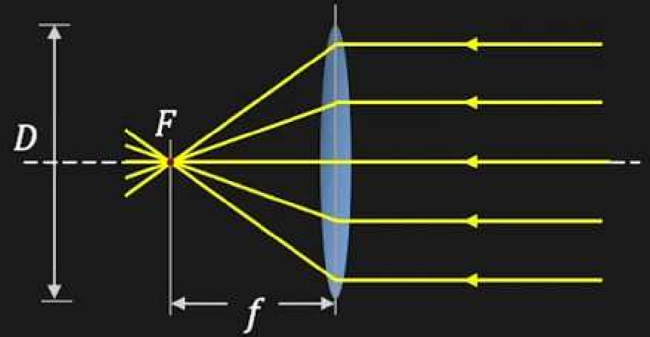


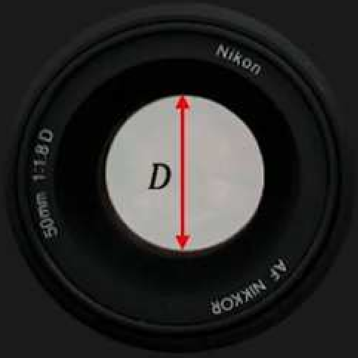
Magnification: $m = \dfrac{i_2}{o_2} \cdot \dfrac{i_1}{o_1}$

Zooming: Move lenses to change magnification

Aperture of Lens

Light receiving area of lens, indicated by lens diameter.





Aperture can be reduced/increased to control image brightness

f-number (f-stop, f-ratio) of Lens

Convenient to represent aperture as a fraction of focal length

Aperture: $D = {}^{f}/_{N}$

f-Number: $N = {}^{f}/_{D}$

where N is called the f-Number of lens.

Ex: A $50mm$ focal length, f/1.8 lens implies:

$N = 1.8$ $(D = 27.8mm)$ when aperture is fully open

$N = 1.8$

$N = 4$

$N = 8$

$N = 11$

Blur Circle (Defocus)

Focused Point

$$\frac{1}{i} + \frac{1}{o} = \frac{1}{f}$$

$$i = \frac{of}{o - f}$$

Defocused Point

$$\frac{1}{i'} + \frac{1}{o'} = \frac{1}{f}$$

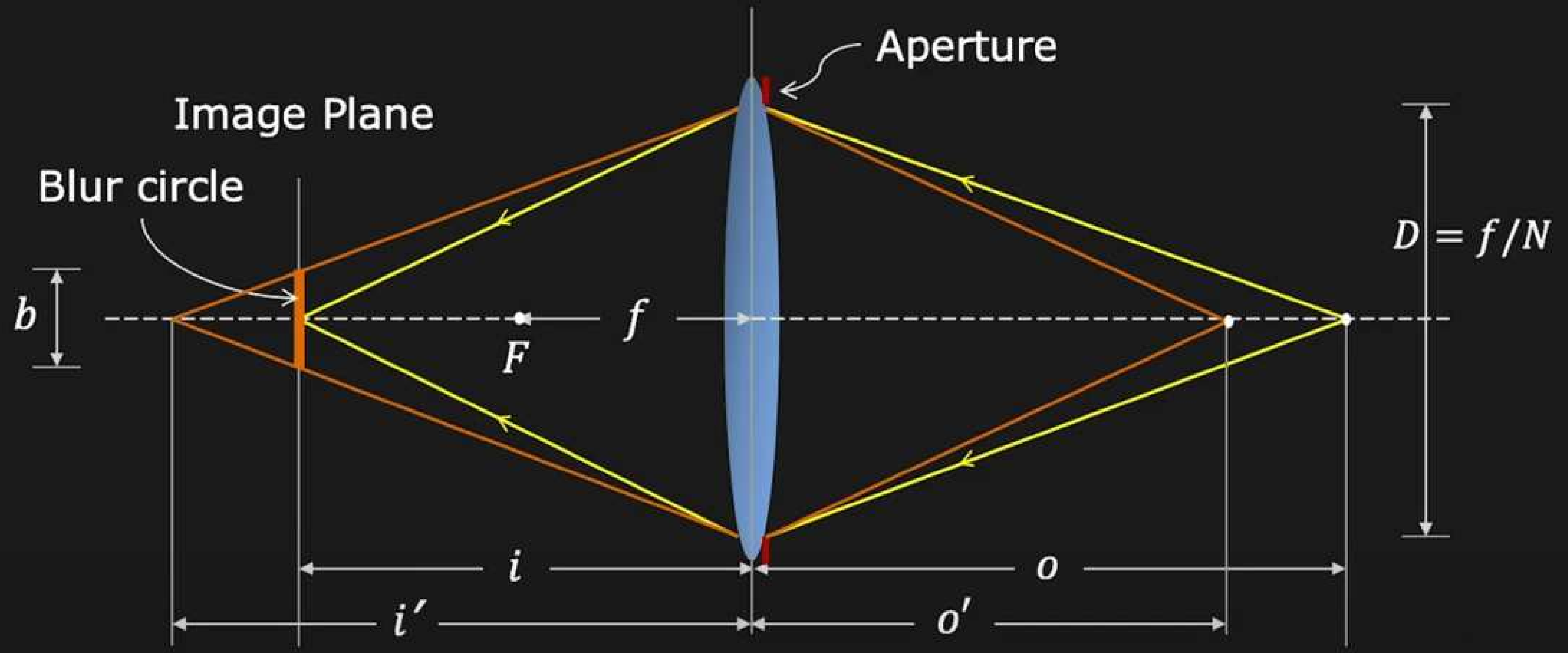
(Gaussian Lens Law)

$$i' = \frac{o'f}{o' - f}$$

Image Formation using Lenses | Image Formation

Watch later Share

Lens Defocus



From similar triangles:

$$\frac{b}{D} = \frac{|i' - i|}{i'}$$

Blur circle diameter:

$$\boxed{b = \frac{D}{i'}|i' - i|} \quad \boxed{b \propto D \propto \frac{1}{N}}$$

MORE VIDEOS

Blur Circle (Defocus)

Focused Point	Defocused Point	
$\frac{1}{i} + \frac{1}{o} = \frac{1}{f}$	$\frac{1}{i'} + \frac{1}{o'} = \frac{1}{f}$	(Gaussian Lens Law)
$i = \frac{of}{o - f}$	$i' = \frac{o'f}{o' - f}$	

Blur Circle (Defocus)

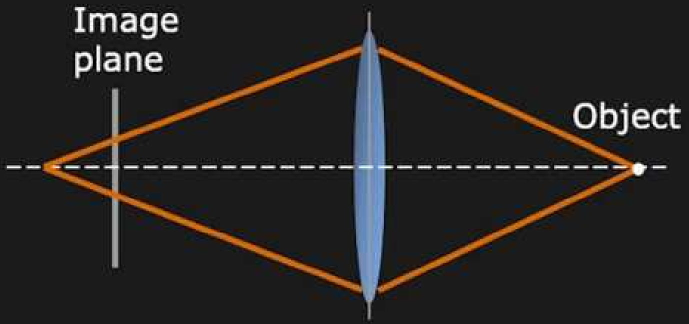
Focused Point	Defocused Point
$\frac{1}{i} + \frac{1}{o} = \frac{1}{f}$	$\frac{1}{i'} + \frac{1}{o'} = \frac{1}{f}$
$i = \frac{of}{o - f}$	$i' = \frac{o'f}{o' - f}$

(Gaussian Lens Law)

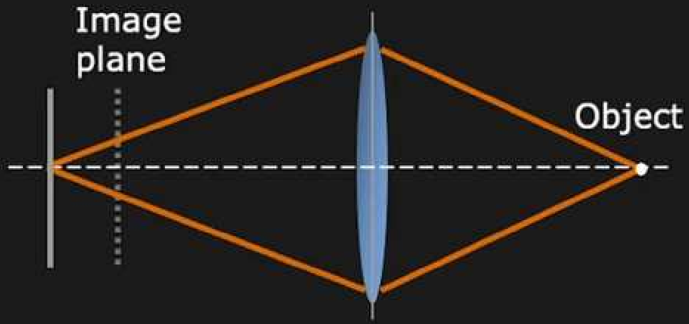
$$i' - i = \frac{f}{(o' - f)} \cdot \frac{f}{(o - f)} \cdot (o - o')$$

$$b = Df \left| \frac{(o - o')}{o'(o - f)} \right| \qquad \boxed{b = \frac{f^2}{N} \left| \frac{(o - o')}{o'(o - f)} \right|}$$

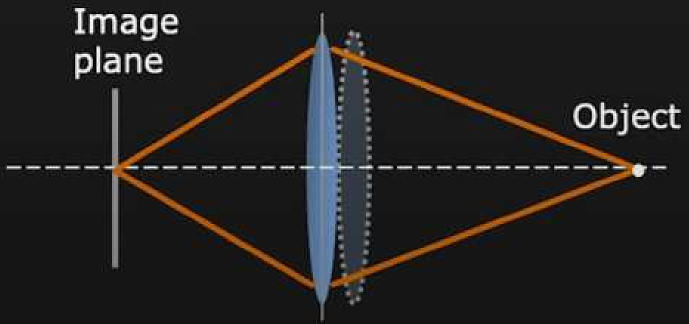
Focusing



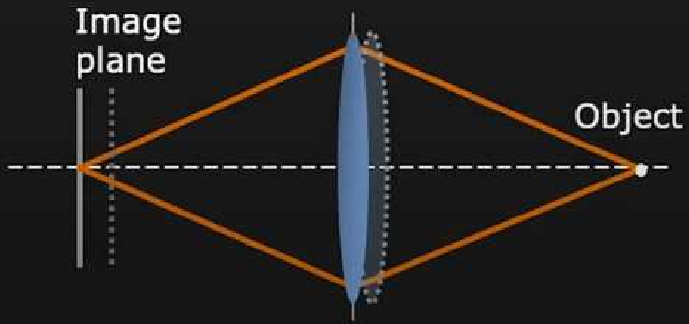
Defocused System



Move the image plane

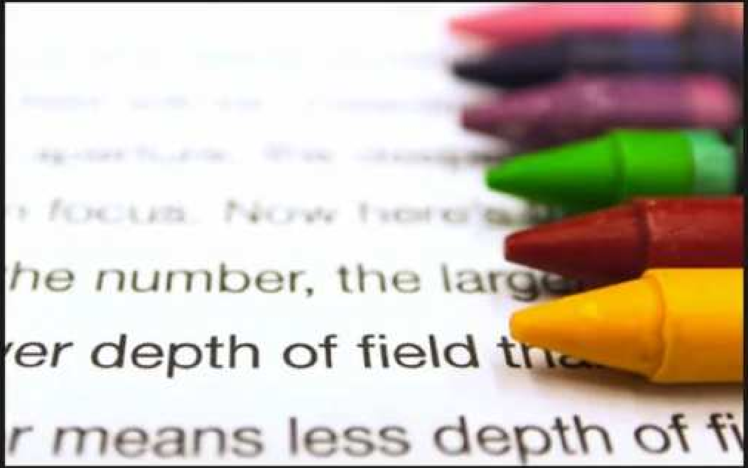


Move the lens



Move both lens and image plane

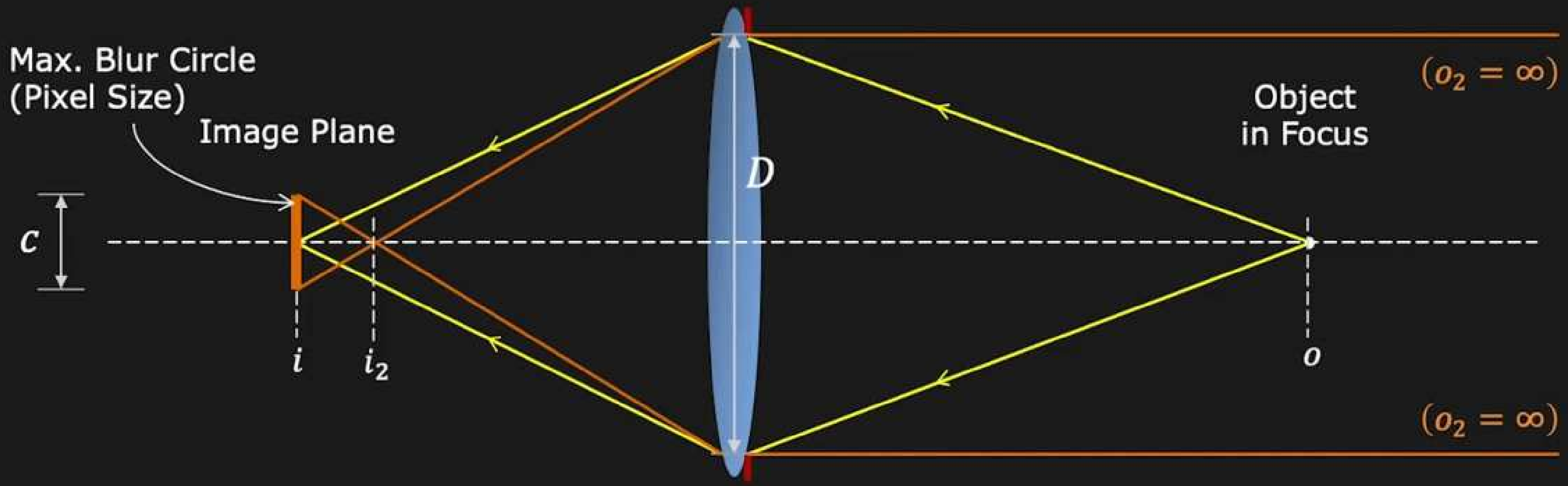
Depth of Field (DoF)



I.10

Range of object distances over which the image is "sufficiently well" focused, i.e., range over which blur b is less than pixel size.

Hyperfocal Distance



The closest distance $o = h$ the lens must be focused to keep objects at infinity $(o_2 = \infty)$ acceptably sharp (blur circle $\leq c$).

Hyperfocal Distance: $\boxed{h = \dfrac{f^2}{Nc} + f}$

Aperture Size: DOF vs. Brightness

Focal Length 50 mm, Focus = 1 m, Aperture D = 25 mm, f-Number N =

Aperture Size: DOF vs. Brightness

Focal Length 50 mm, Focus = 1m, Aperture D = 3.125mm, f-Number N 16

Aperture Size: DOF vs. Brightness

Large Aperture (Small f-Number)

- Bright Image or Short Exposure Time
- Shallow Depth of Field

Small Aperture (Large f-Number)

- Dark Image or Long Exposure Time
- Large Depth of Field

Tissue Box Camera

Camera

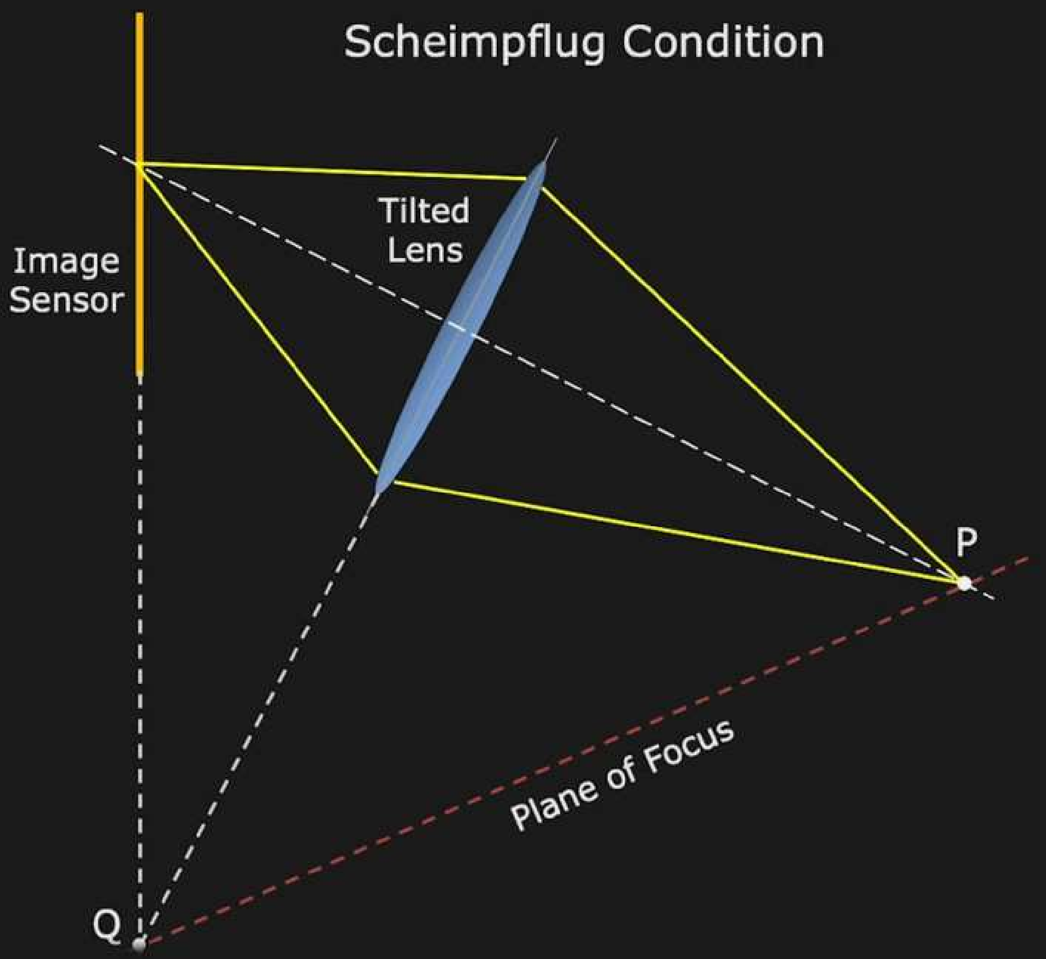
Image

Blocking the Lens

Camera

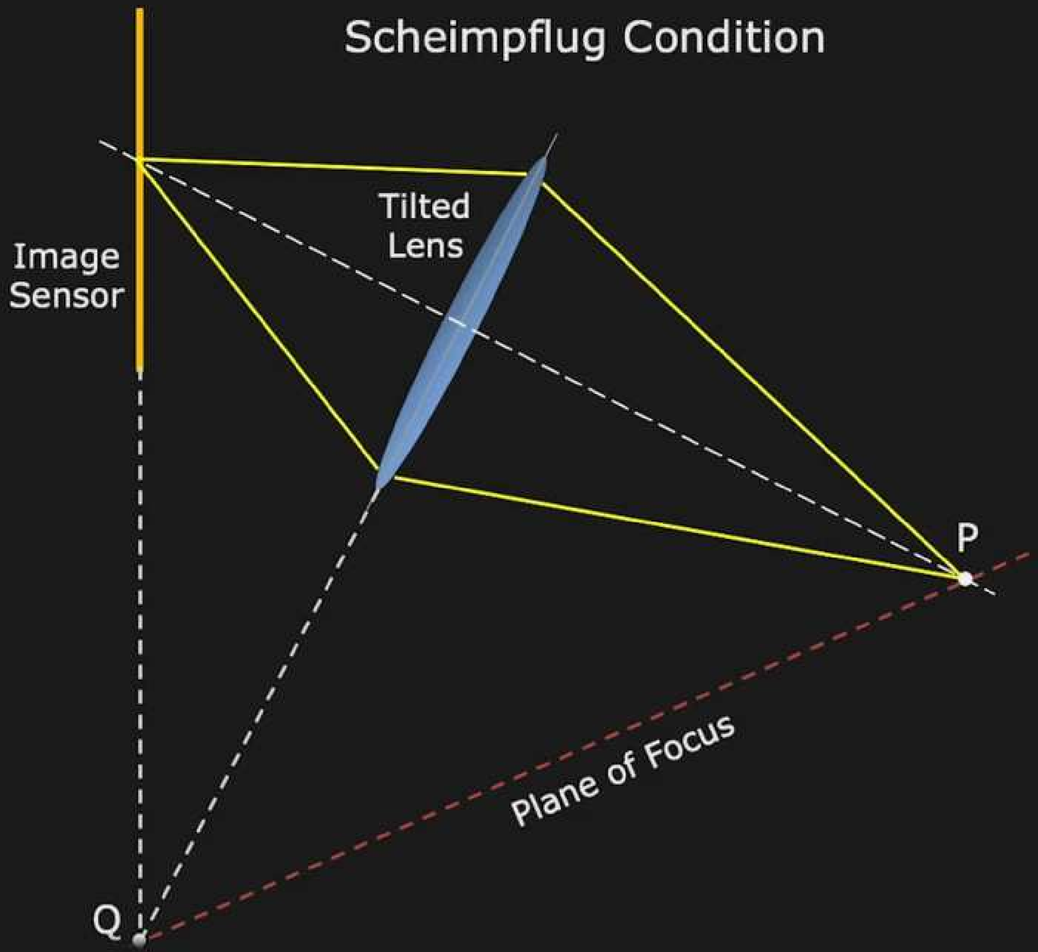
Image

Tilting the Lens



Tilting the Lens

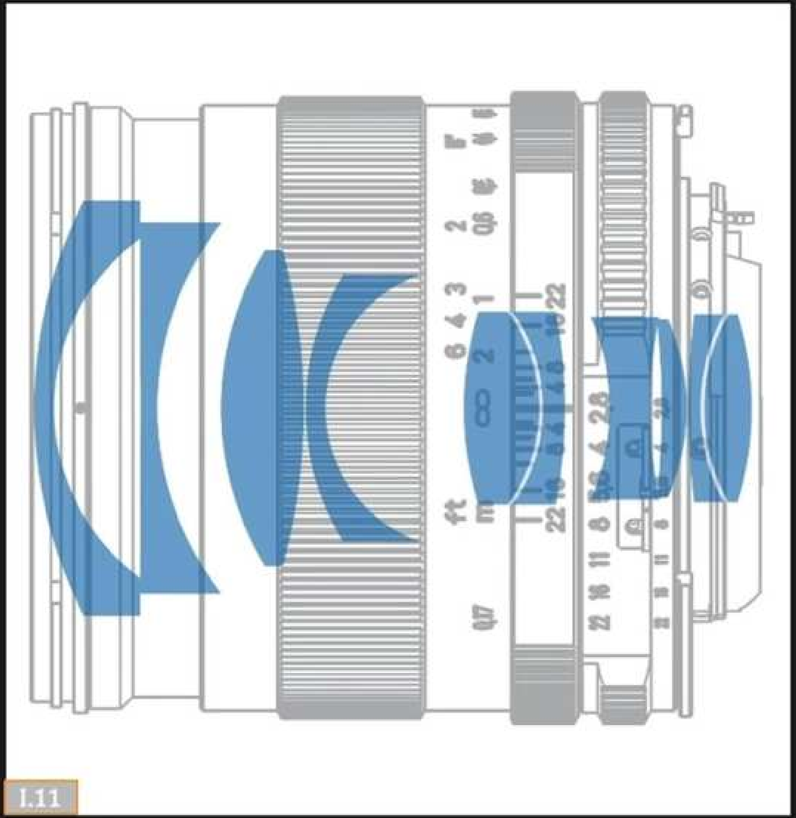
Scheimpflug Condition



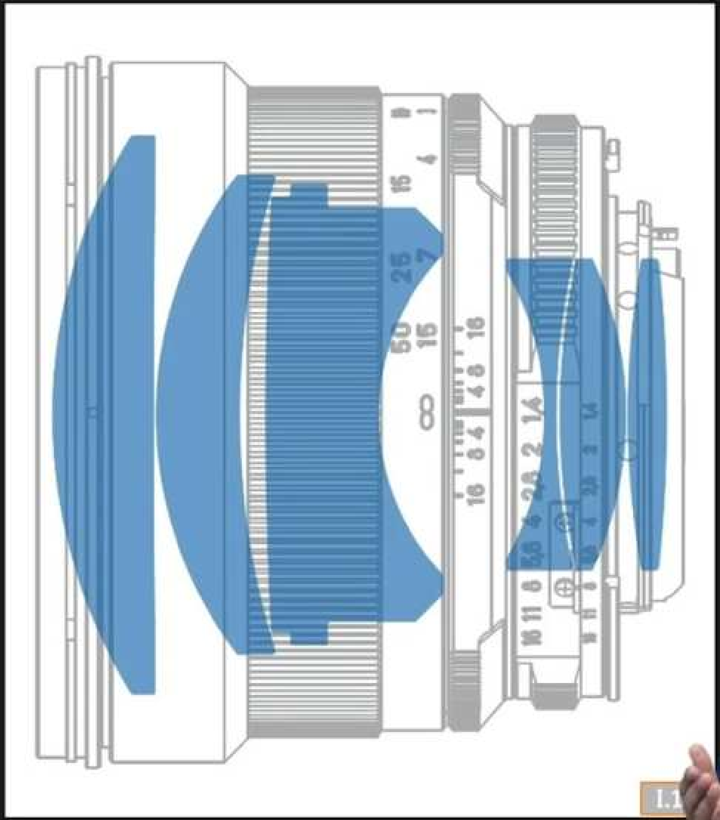
Tilt Camera

stenopeika.com

Compound Lenses

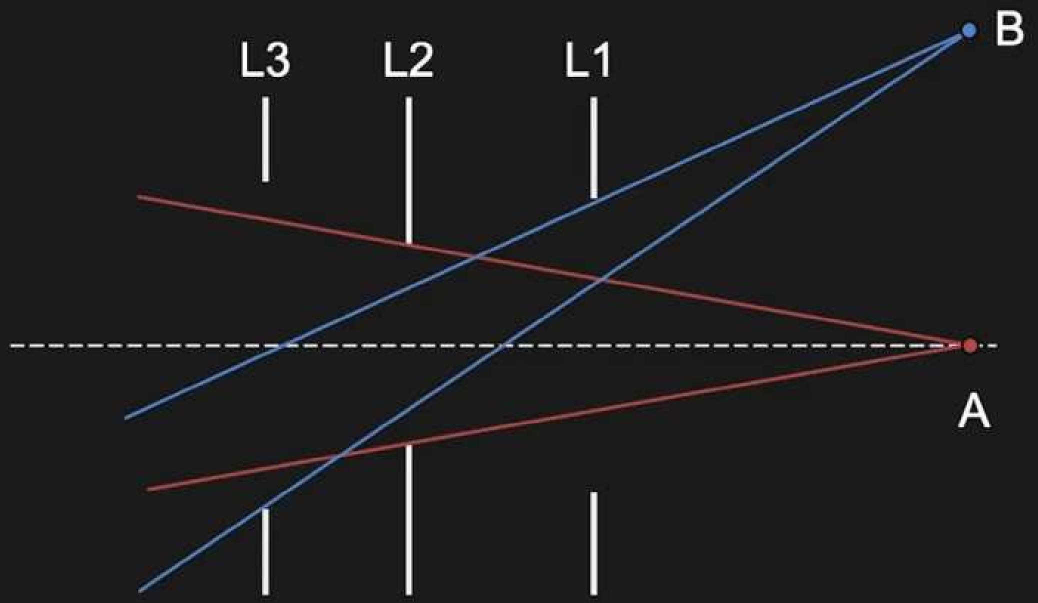


Zeiss 25mm F2.8



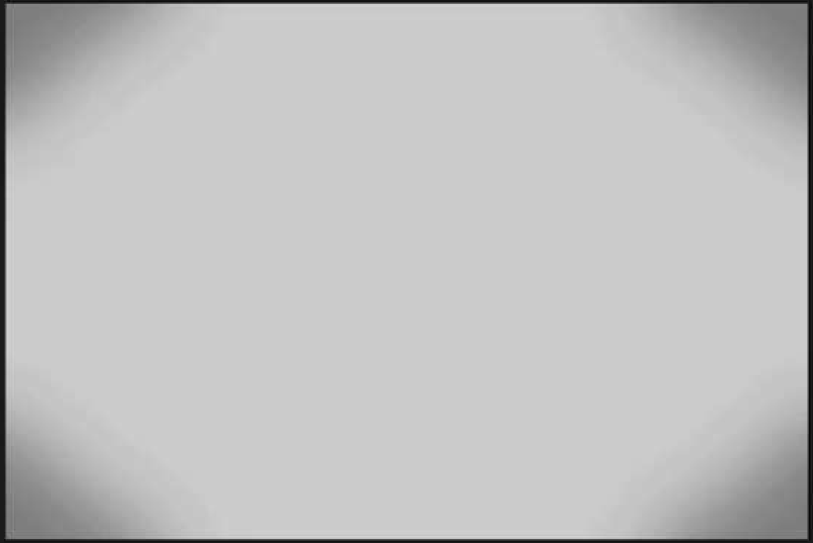
Zeiss 85mm F1.4

Vignetting



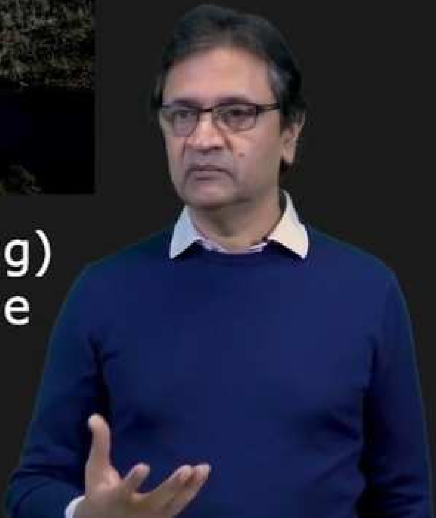
More light passes through L3 from point A than point B.
Results in a smooth fall-off in brightness from A to B.

Vignetting

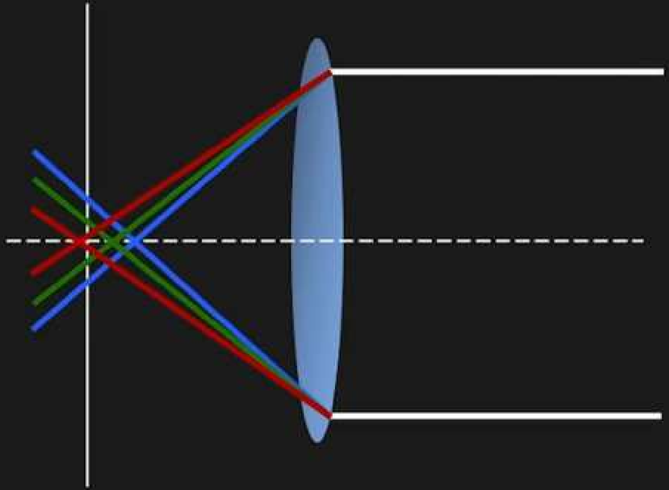


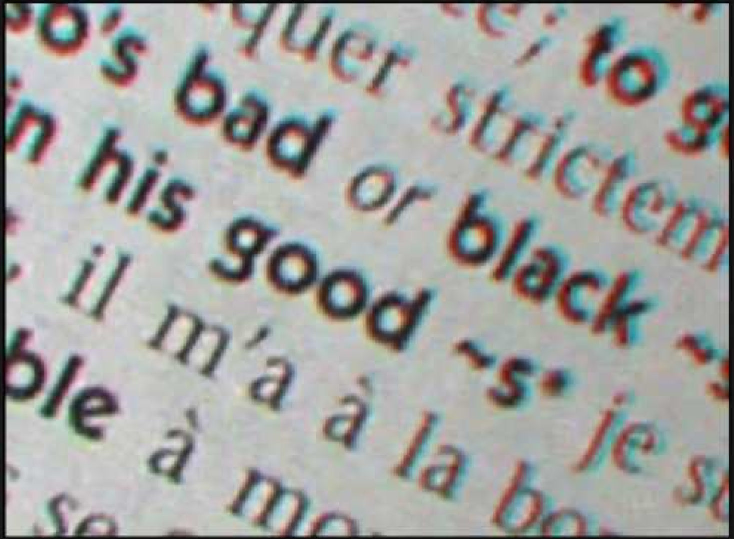
Brightness fall-off (Vignetting) in image of a White Wall

Brightness fall-off (Vignetting) in image of a Natural Scene



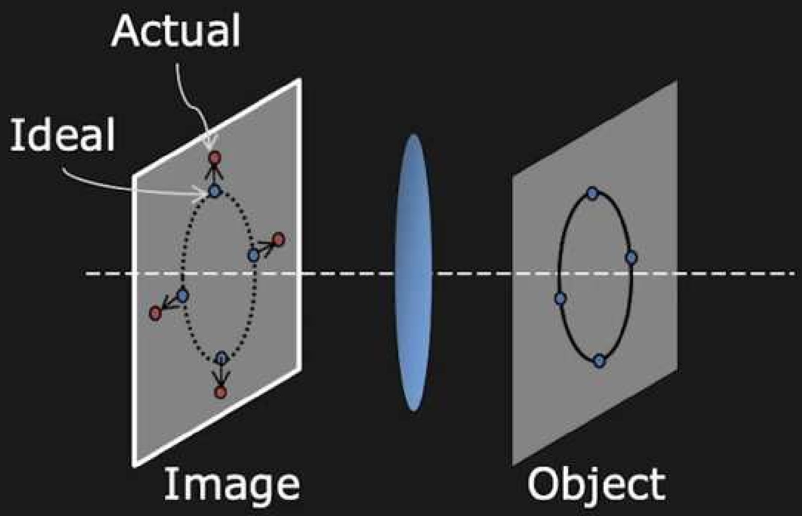
Chromatic Aberration



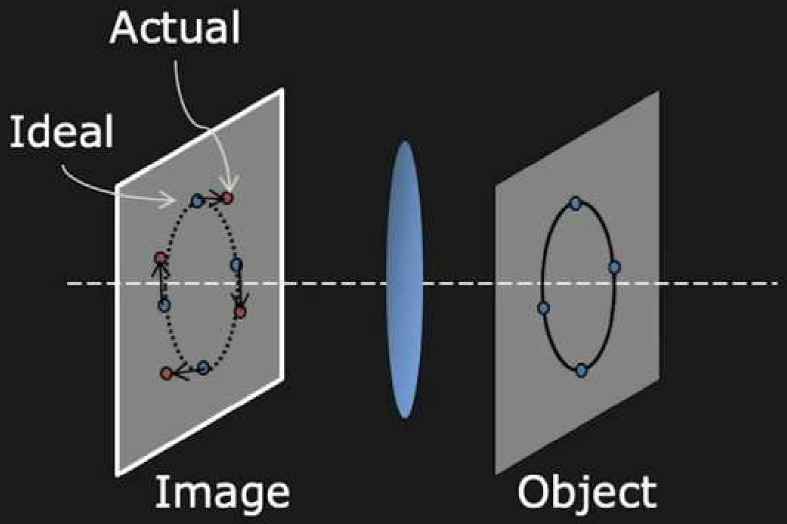


Refractive index (and hence focal length) of lens is different for different wavelengths.

Geometric Distortion



Radial distortion



Tangential distortion

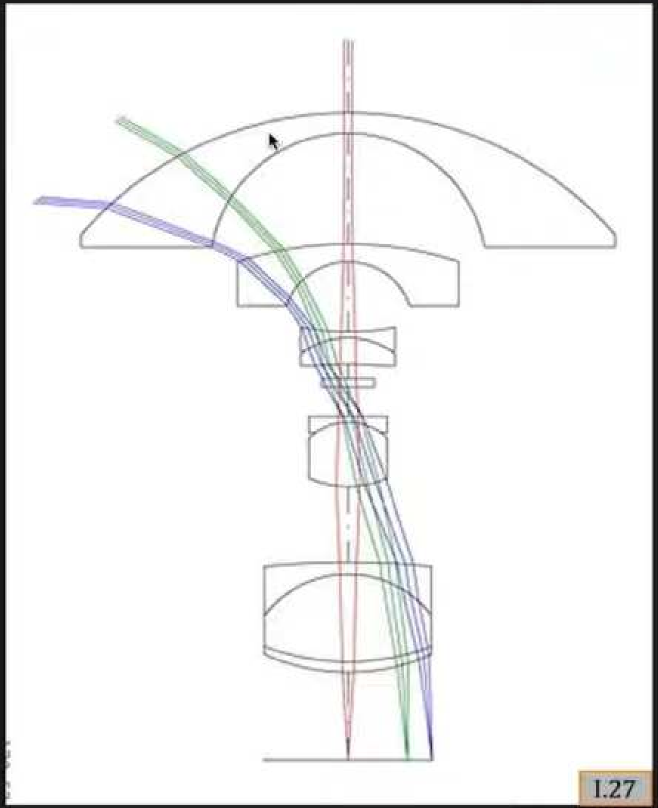
Due to lens imperfections

Geometric Distortion Correction

Radial (Barrel) distortion

Undistorted image

Fisheye Lens Camera



170^o Fisheye Lens

[Miyamoto 1

Fisheye Image

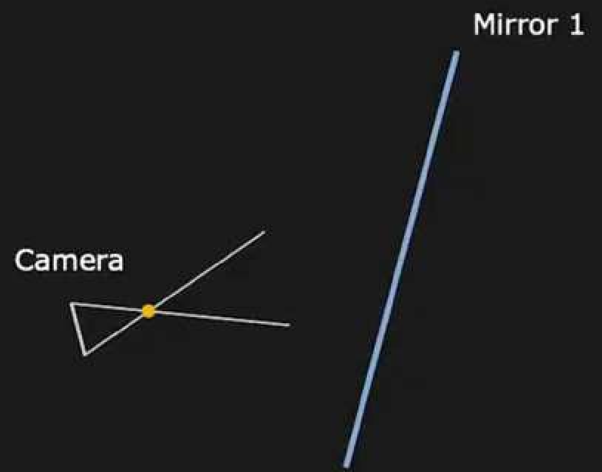
Fisheye Lens

Hemispherical Field of View

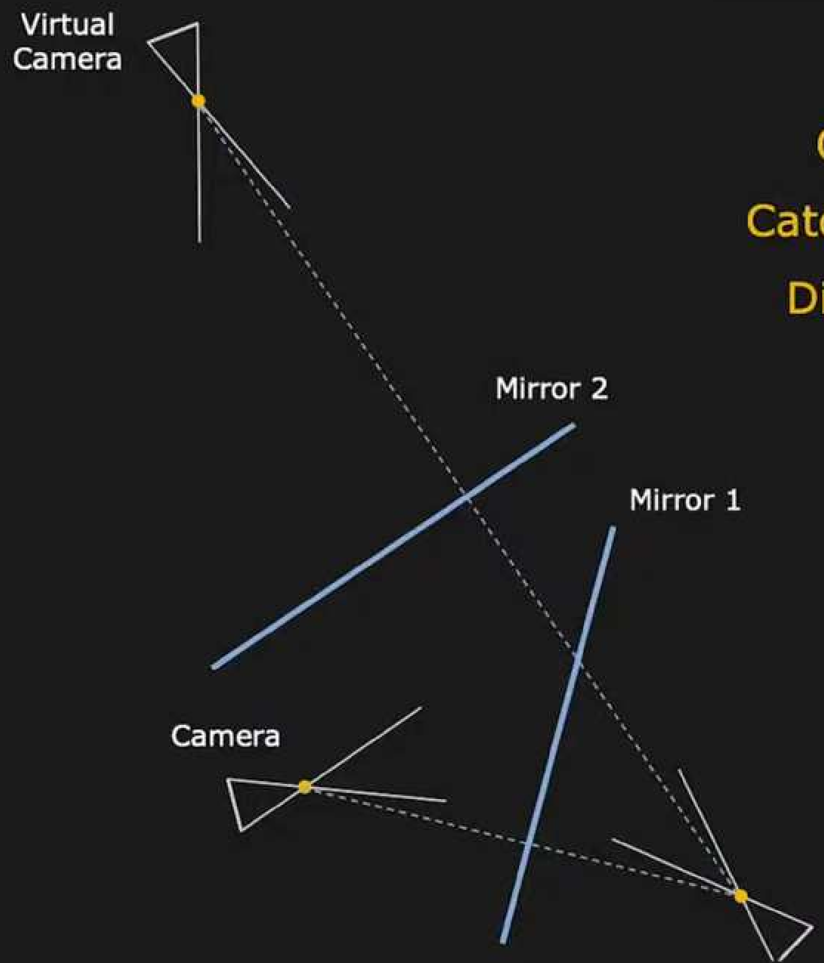
Capturing the Complete Sphere

Ricoh Theta

Planar Mirrors and Reflected Cameras



Planar Mirrors and Reflected Cameras

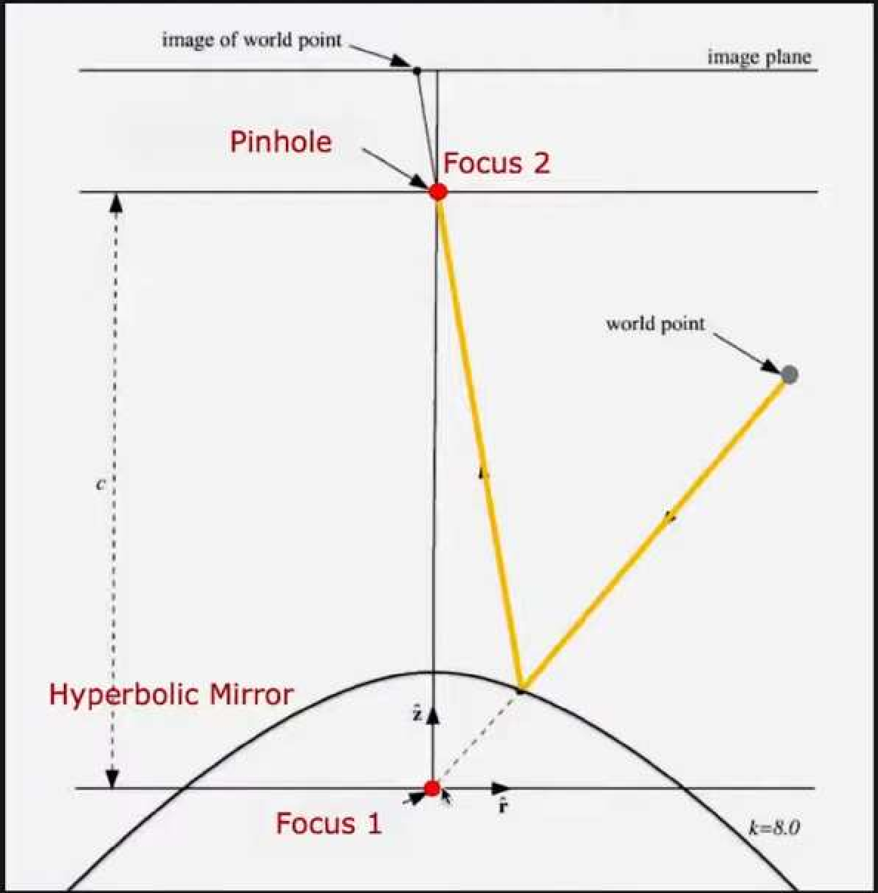


Catadioptrics =

Catoptrics (Mirrors) +

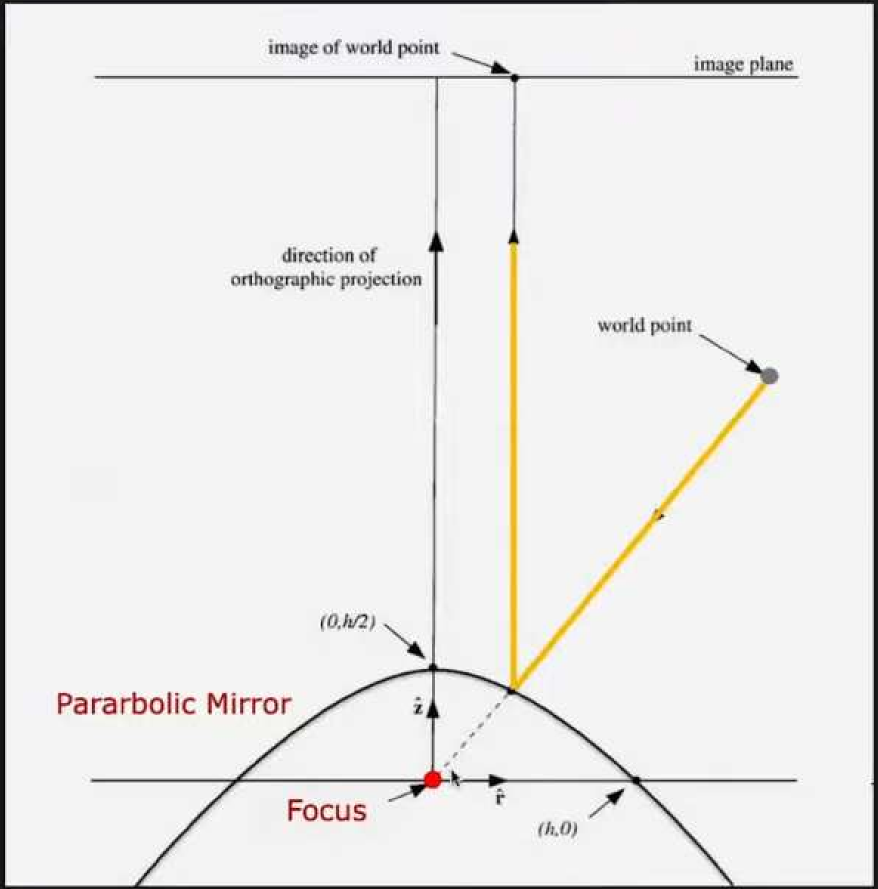
Dioptrics (Lenses)

Hyperbolic Mirror Camera



[Yagi 1999, Baker 1

Parabolic Mirror Camera



[Baker 1

Parabolic Mirror Image

[Nayar 1

Parabolic Mirror Image

[Nayar 19

Sony “Bloggie”

Kogeto “Dot” for iPhone

Concave Mirrors and Telescopes

Concave Mirror

21 ft

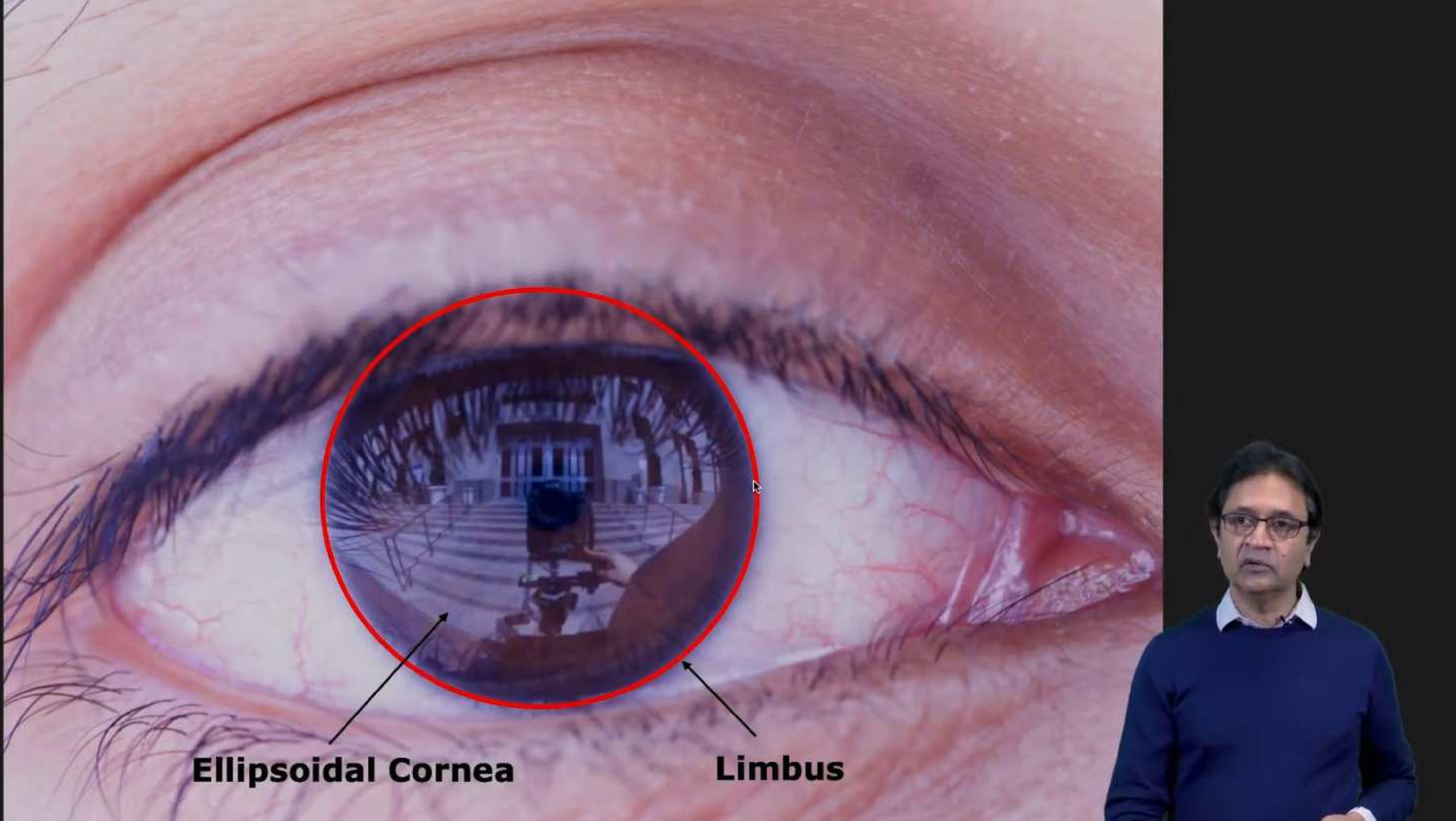
James Webb Space Telescope

Like Scallops?

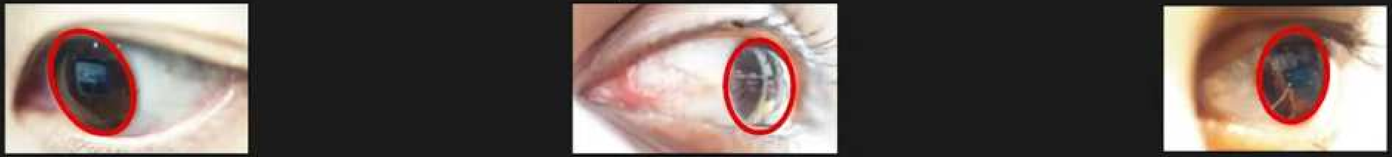
Scallop Eyes

Telescopic Eyes with Parabolic Mirrors

The World in an Eye


Ellipsoidal Cornea
Limbus

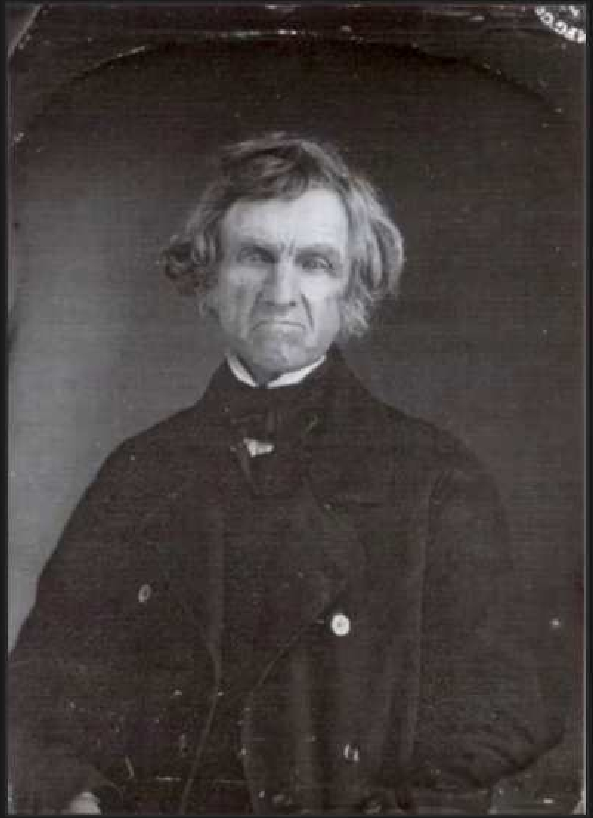
Eye Images



Environment Images

Retinal Images

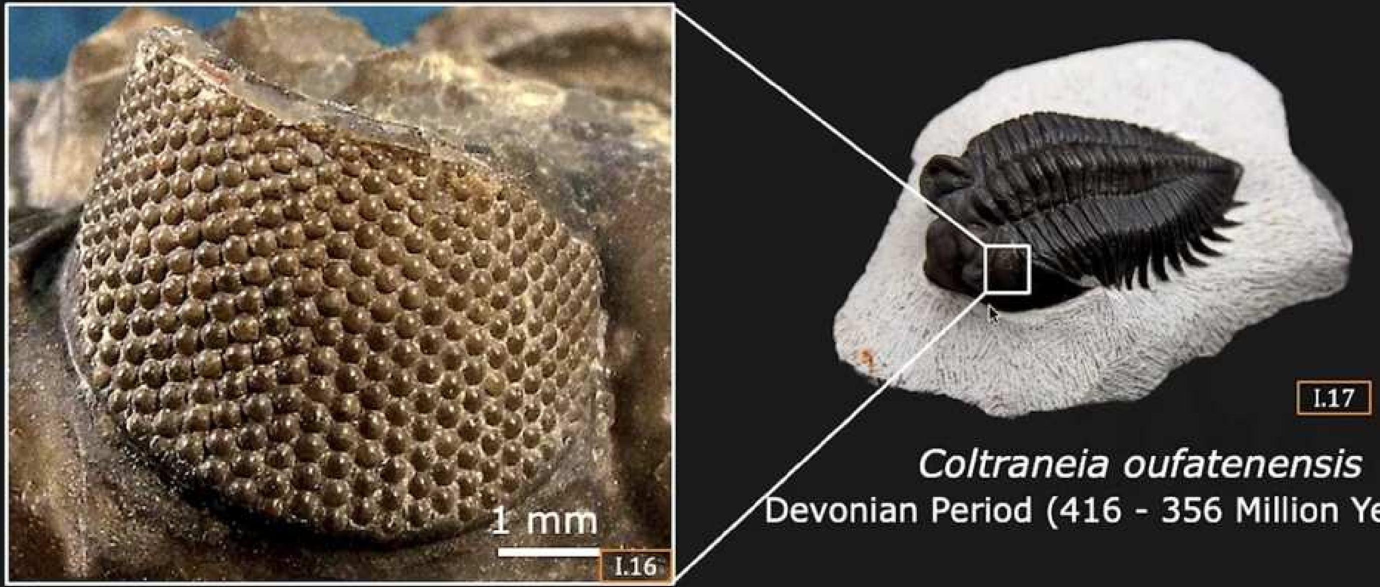
Going Back in Time



"Grumpy Grandpa", c. 1845

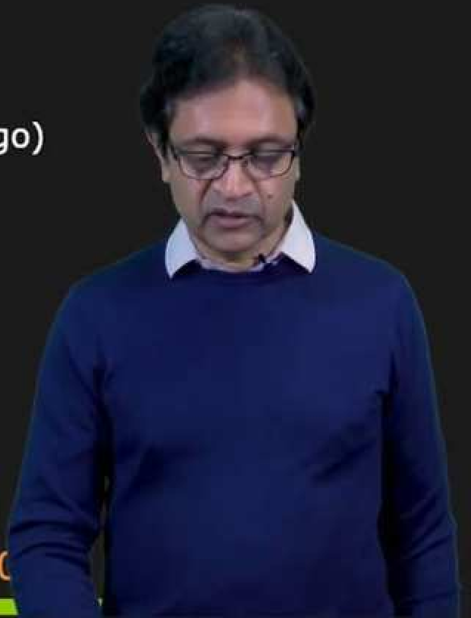
[Courtesy: Joe Bauman, Daguerreian Socie

Fossilized Eye of Trilobite



Coltraneia oufatenensis
Devonian Period (416 - 356 Million Years Ago)

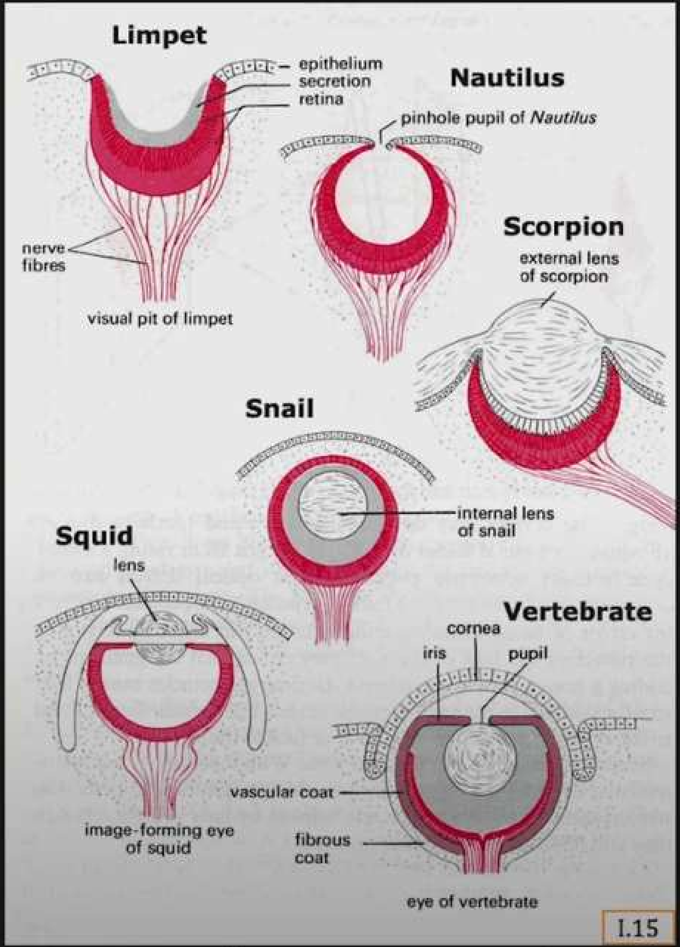
Earliest kind of eye preserved as a fossil. The facets are the corneal lenses made from transparent Calcite ($CaCO_3$).



[Clarkson 20

Animal Eyes | Image Formation

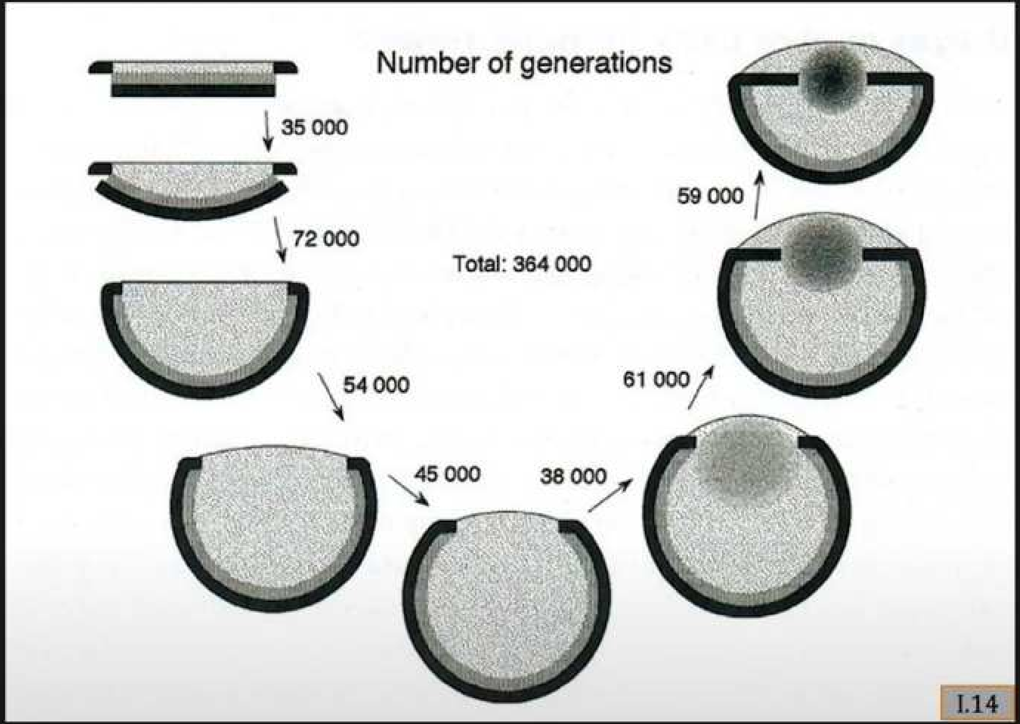
Primitive Eyes



Animal Eyes | Image Formation

Watch later Share

Evolution of Eye: A Simulation



A patch of light sensitive epithelium can be gradually turned into a perfectly focused camera-type eye, if there is a continuous natural selection for improved spatial vision. This simulation reveals that the complete evolution can be accomplished in about 400,000 generations. First a pigment cup eye evolves, and then a lens.

EYE AND BRAIN

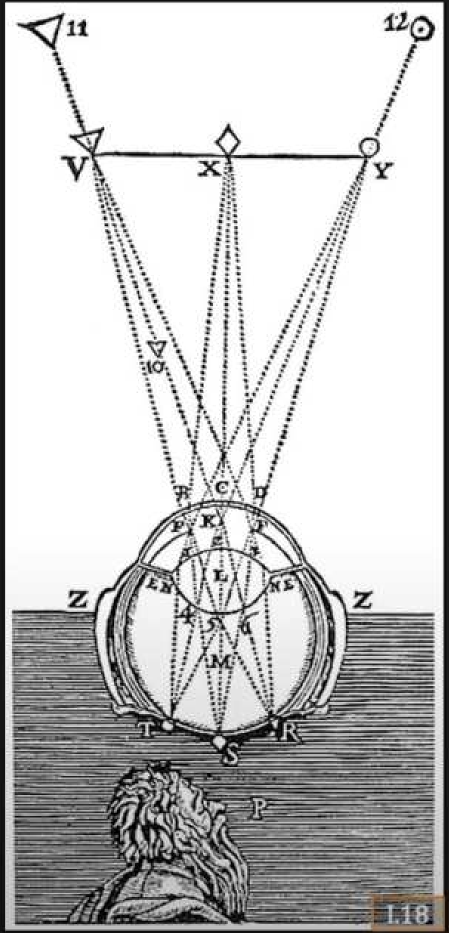
[Nilsson 199-]

2:57 / 15:05 • Evolution of Eye: A Simulation

Animal Eyes | Image Formation

Watch later Share

Image Formation in the Eye

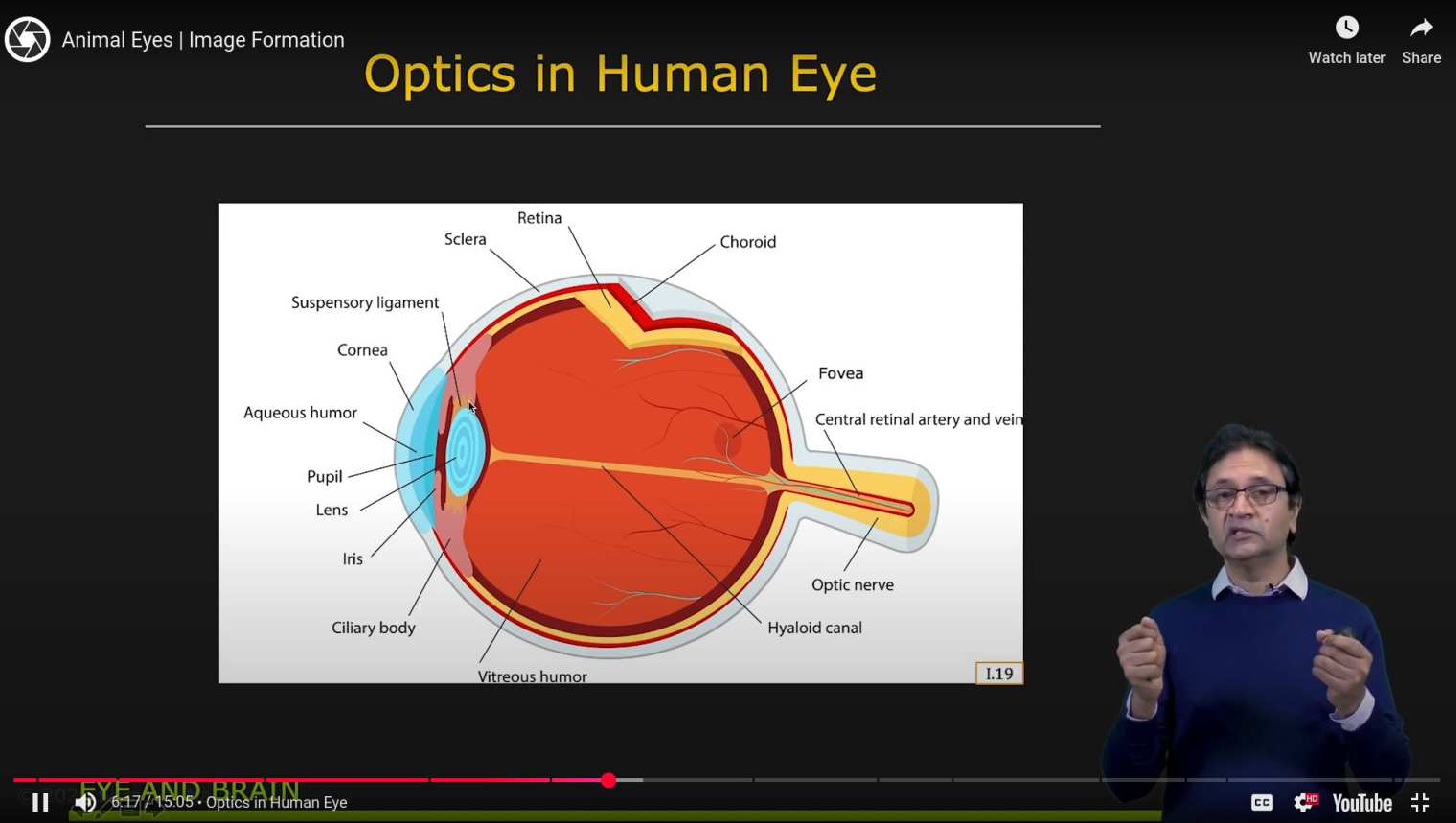


Descartes removed the eye of an ox, scraped its back to make it transparent, and then observed on it from a darkened room "not perhaps without wonder and pleasure" the inverted image of the scene.

[Descartes 1

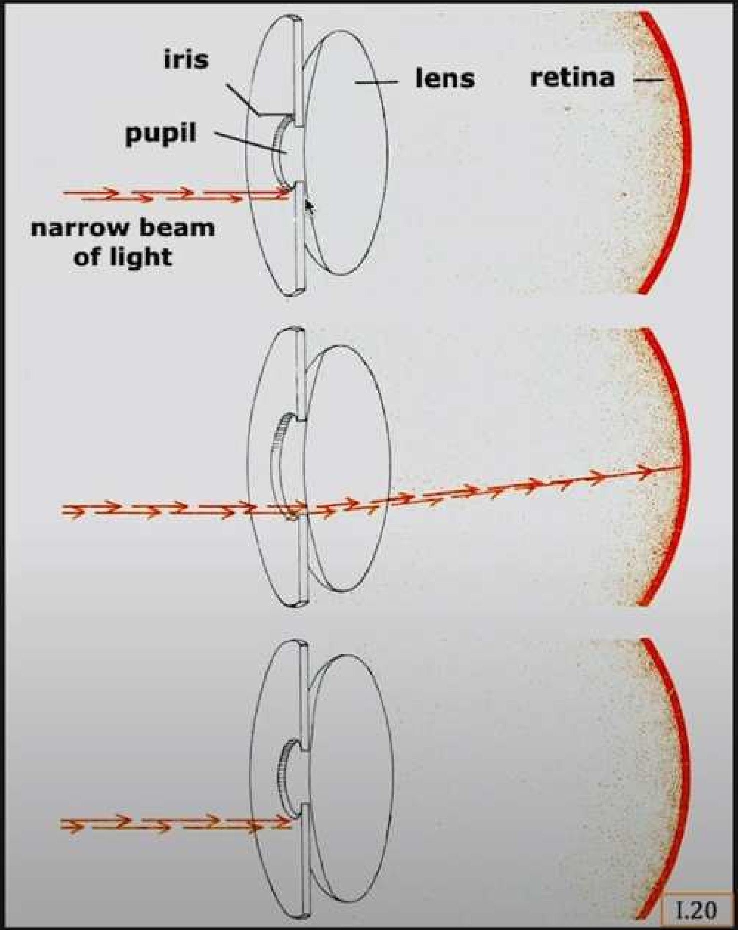
EYE AND BRAIN

4:33 / 15:05 • Image Formation in the Eye



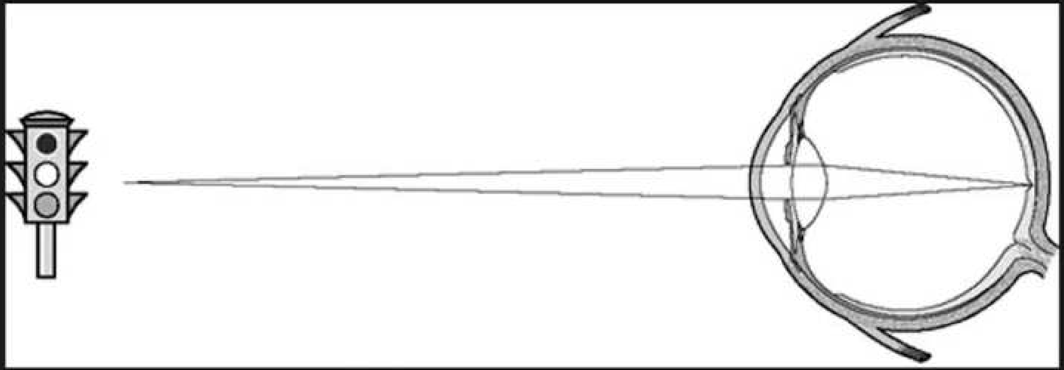
Animal Eyes | Image Formation
Optics in Human Eye
Retina
Sclera
Choroid
Suspensory ligament
Cornea
Fovea
Aqueous humor
Central retinal artery and vein
Pupil
Lens
Iris
Optic nerve
Ciliary body
Hyaloid canal
Vitreous humor
I.19
Watch later
Share
EYE AND BRAIN
6:17 / 15:05 • Optics in Human Eye

Human Eye: Iris Control System

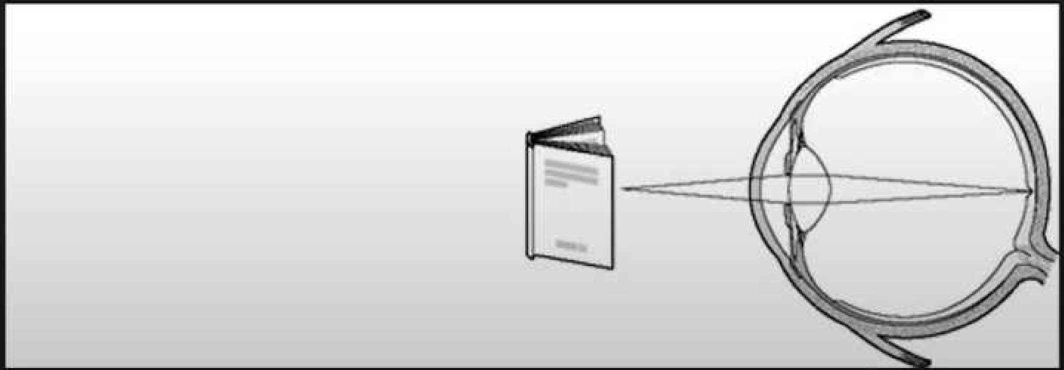


Making the iris oscillate with a narrow beam of light. When the iris opens up, strong light reaches the retina, which causes the retina to close. When it closes, no light is received by the retina and the iris opens again. The frequency and amplitude of this oscillation of the iris reveals the response of its control system.

Accommodation (Focusing) in the Eye

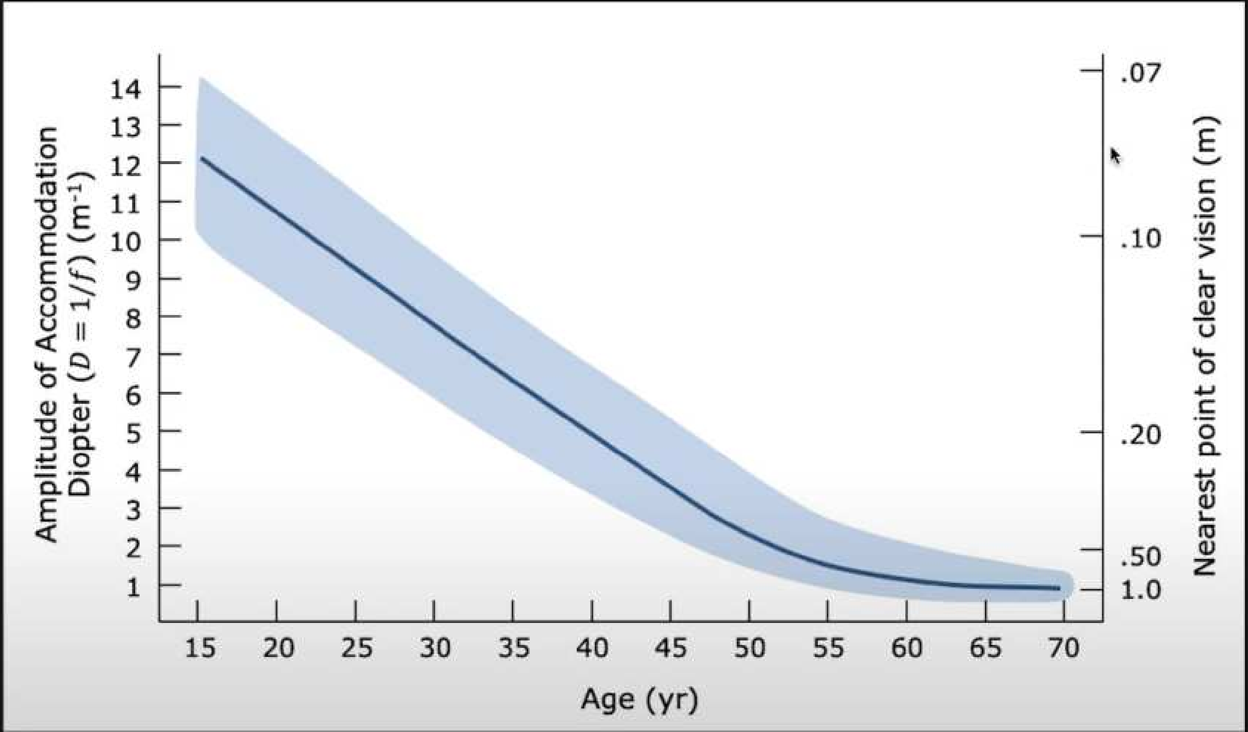


Focusing on distant objects: Lens is relaxed



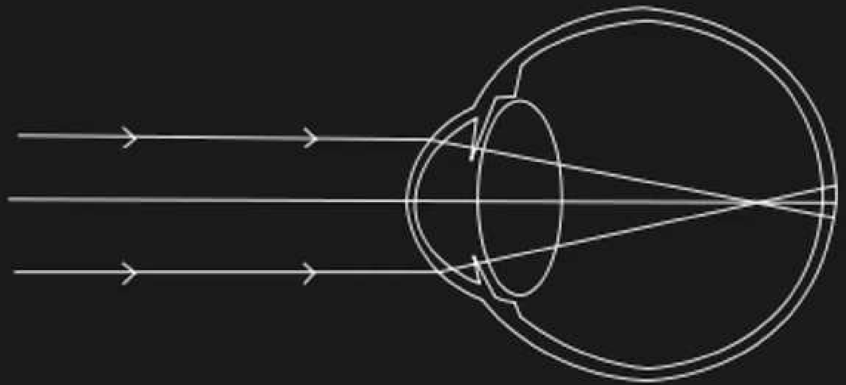
Focusing on nearby objects: Lens is squished

Change in Accommodation with Age

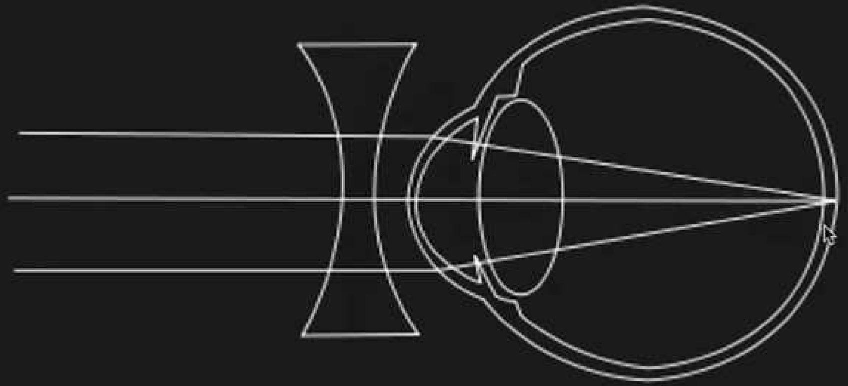


Accommodation decreases with age

Myopia (Near-Sightedness)

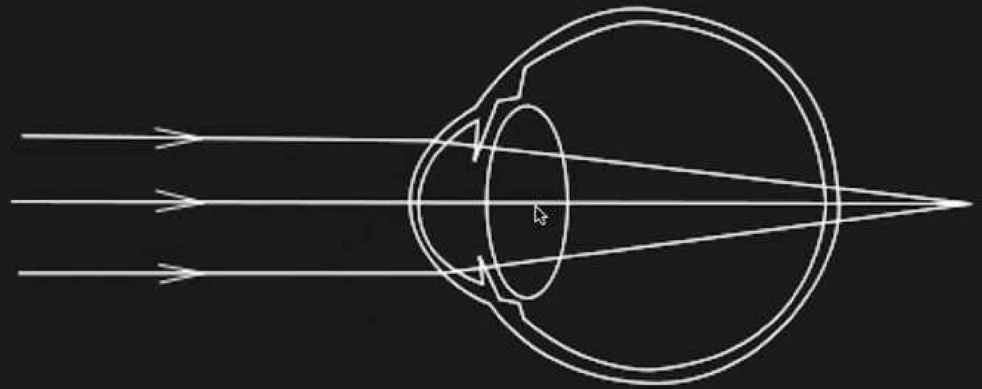


Inability to focus on objects far away

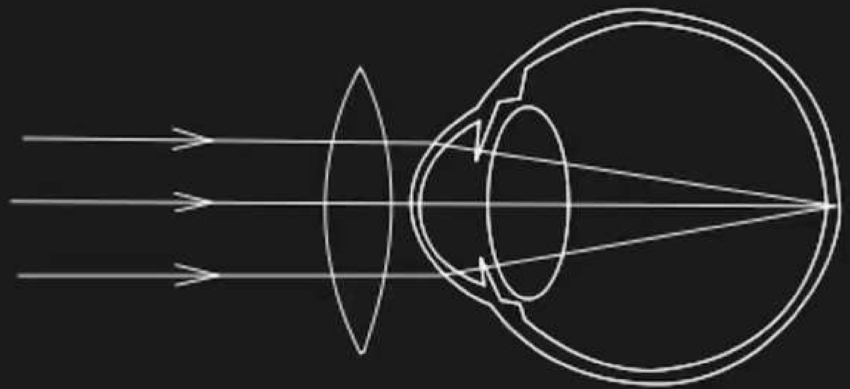


Can be fixed by using a diverging (concave) lens

Hyperopia (Far-Sightedness)

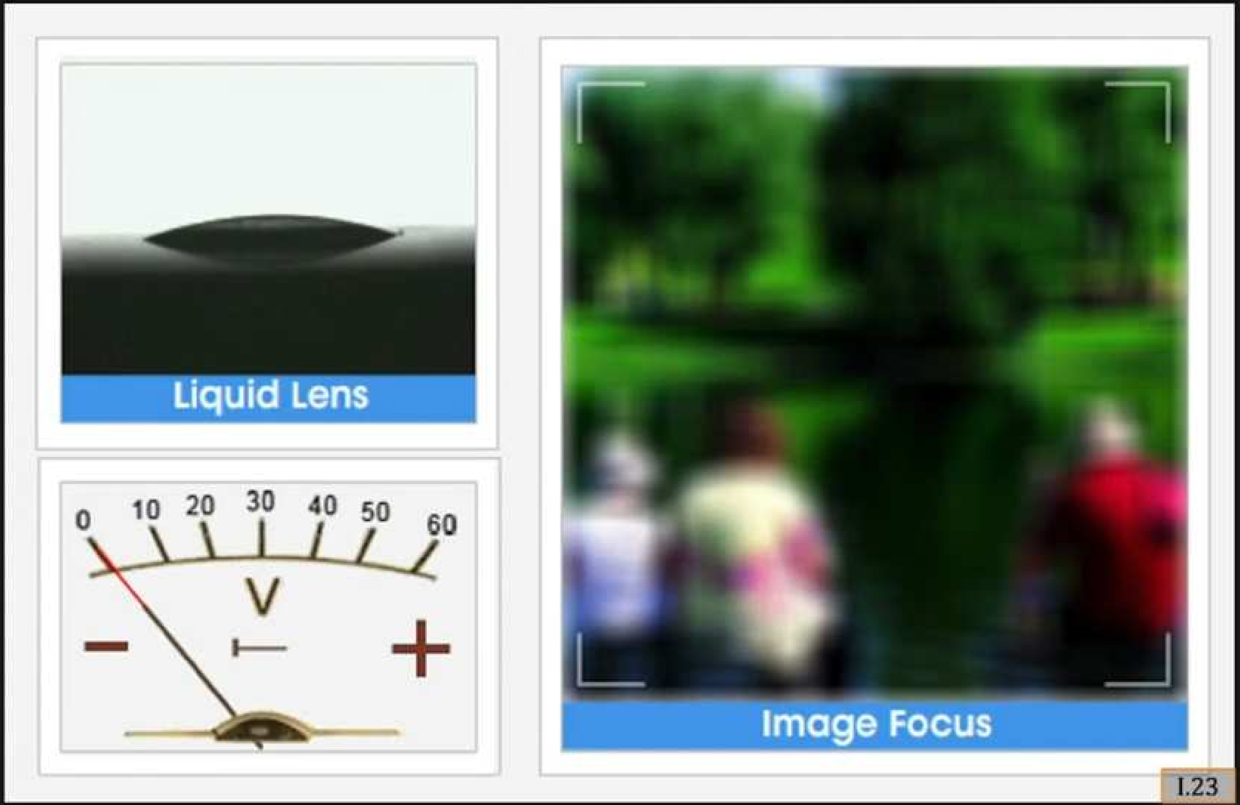


Inability to focus on nearby objects



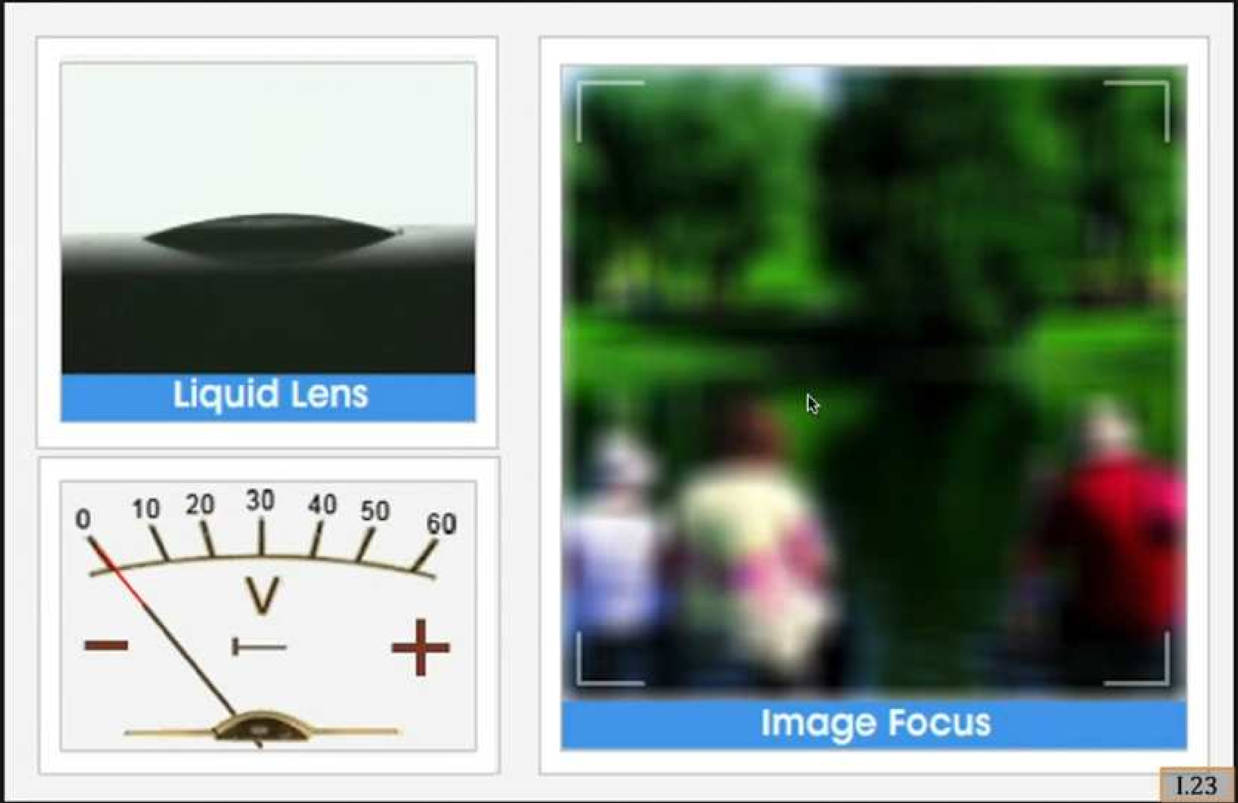
Can be fixed by using a converging (convex) lens

Liquid Lens



The shape and hence the focal length of the liquid lens can be precisely controlled by applying a voltage.

Liquid Lens



I.23

The shape and hence the focal length of the liquid lens can be precisely controlled by applying a voltage.

Image Sensing

Need to convert Optical Images to Digital Images (*numbers*) for computer representation and use.

Topics:

(1) A Brief History of Imaging

(2) Types of Image Sensors

(3) Resolution, Noise, Dynamic Range

(4) Sensing Color

(5) Camera Response and HDR Imaging

Image Sensing

Need to convert Optical Images to Digital Images (*numbers*) for computer representation and use.

Topics:

(1) A Brief History of Imaging

(2) Types of Image Sensors

(3) Resolution, Noise, Dynamic Range

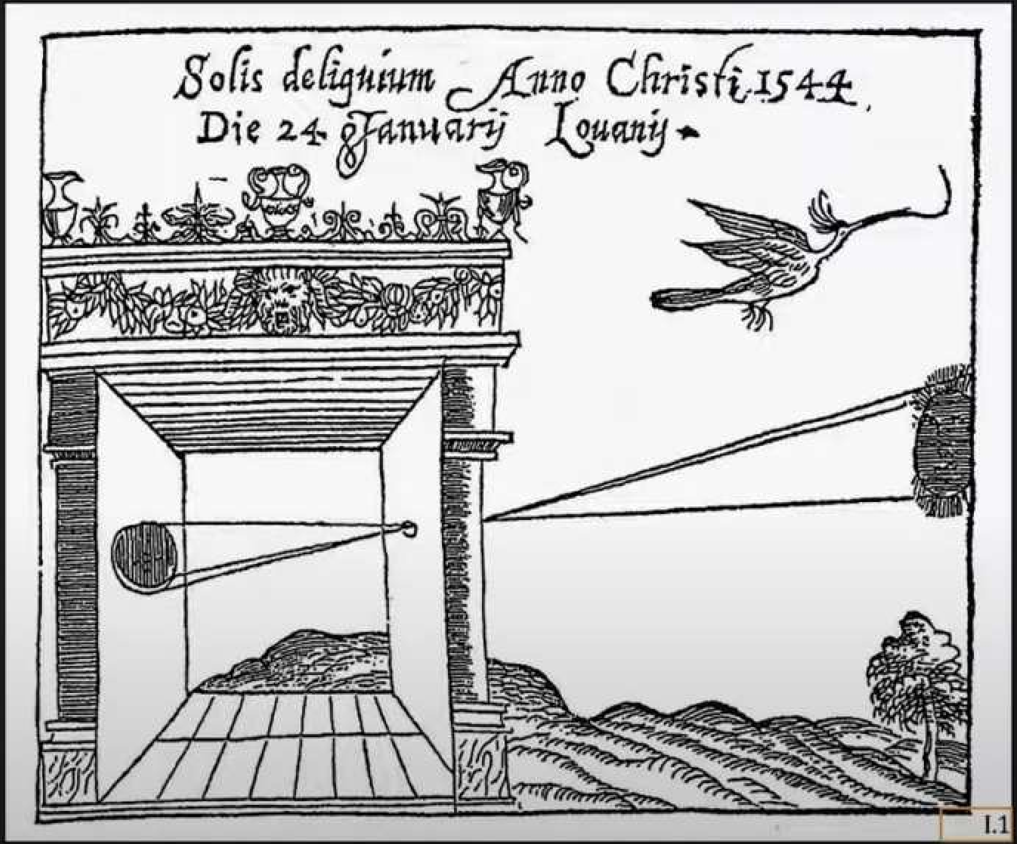
(4) Sensing Color

(5) Camera Response and HDR Imaging

(6) Nature's Image Sensors

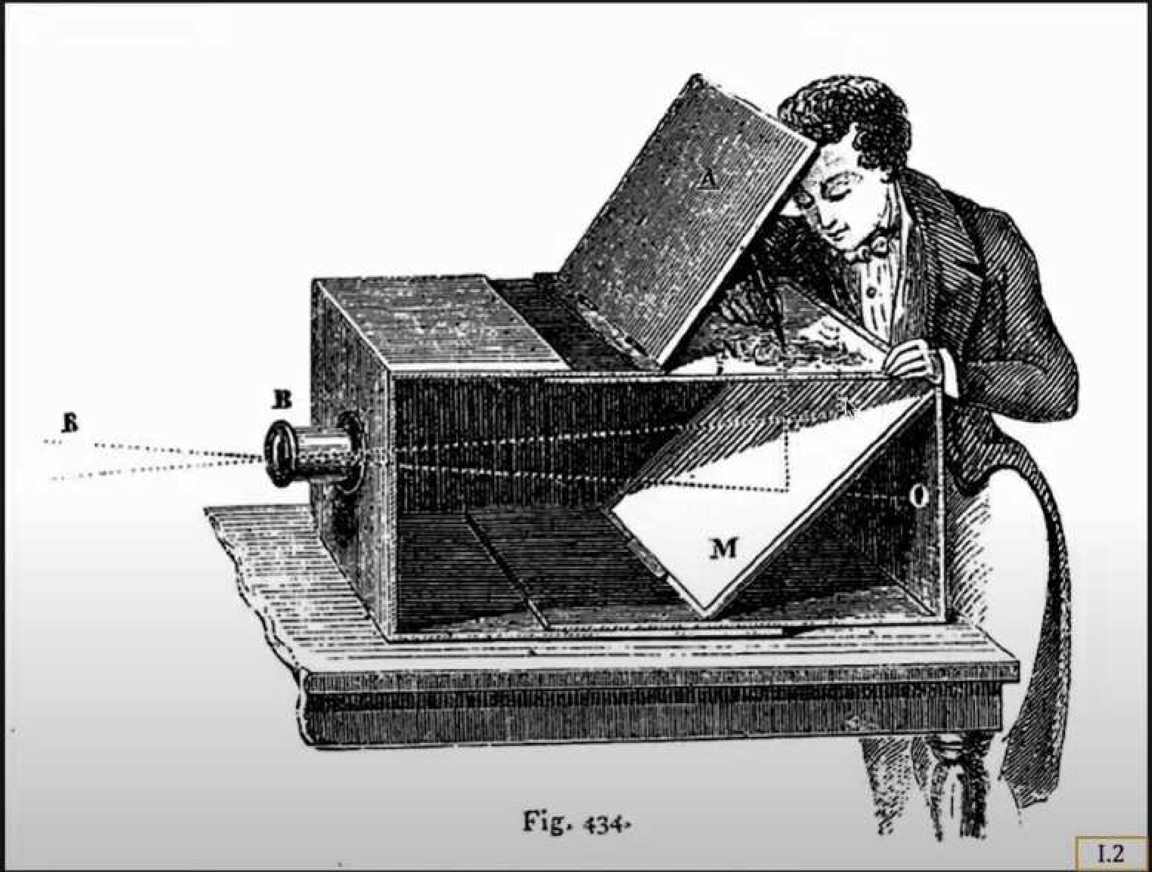
Pinhole Camera

1558



Camera Obscura

Lens Based Camera Obscura



1558

1568

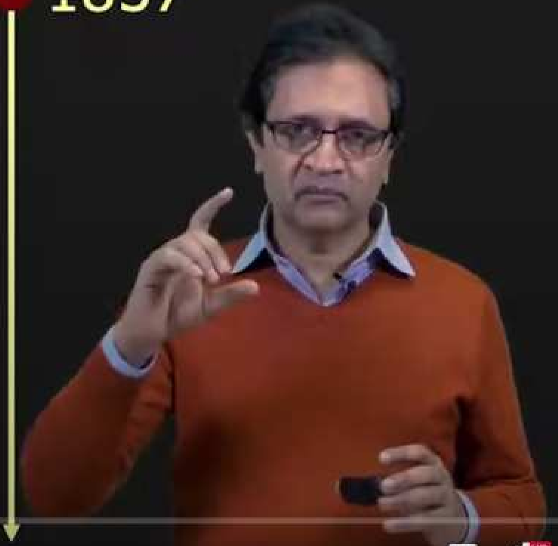
Invention of Film

Still Life, Louis Jaques Mande Daguerre, 1837

1558

1568

1837



Color Film

Louis Ducos du Hauron, 1887

1558
1568
1837
1877

Ernemann Camera

1558

1568

1837

Silicon Image Detector

1558
1568
1837
1887
1928
19

Digital Cameras

1558
1568
1837
1887
1928
19

A Brief History of Imaging | Image Sensing

Phones with Cameras

iPhone 1

1558

1568

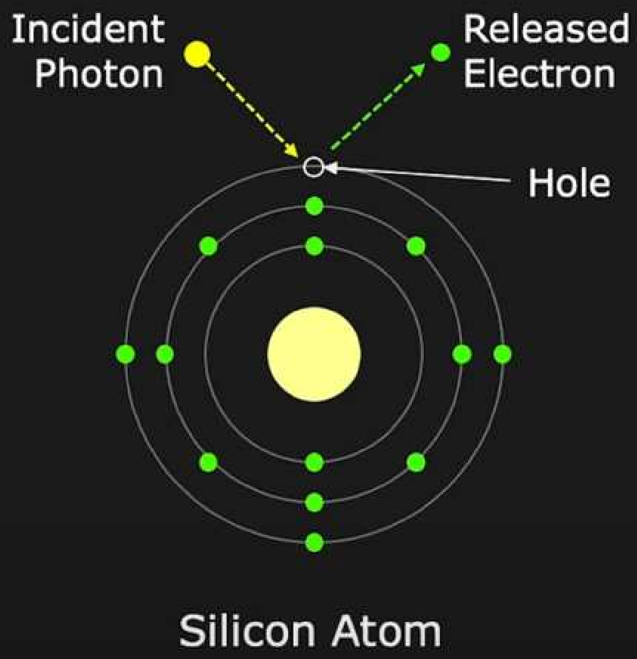
1837

1887

1928

7:41 / 15:14 • Phones with Cameras

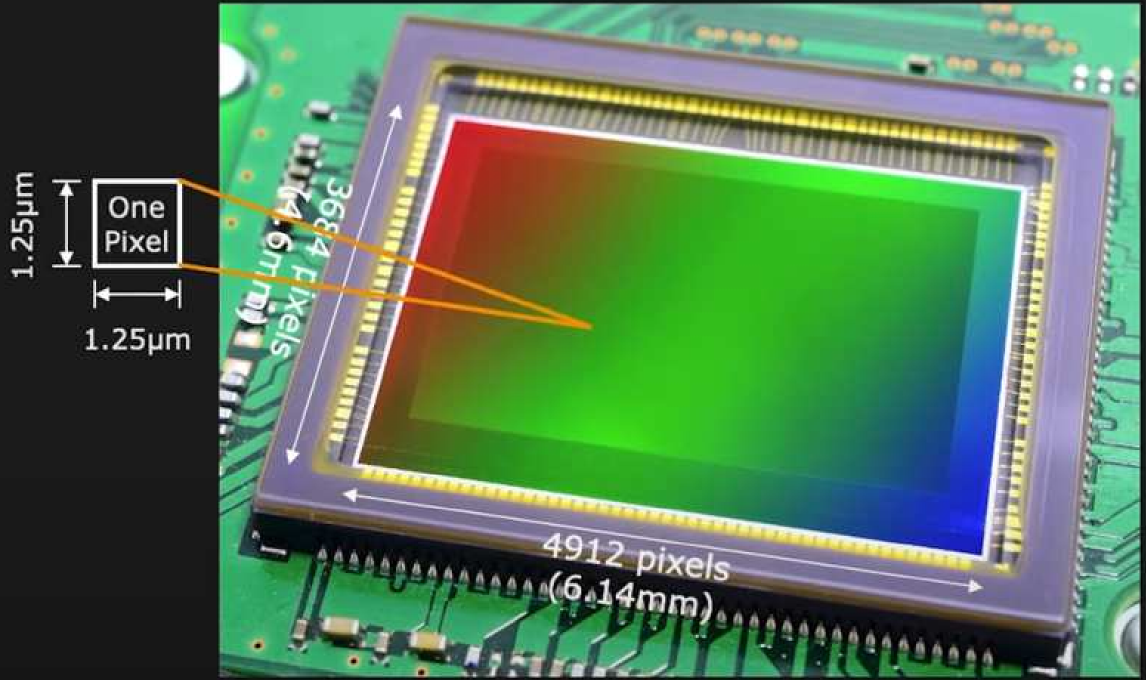
Converting Light into Electric Charge



Photon with sufficient energy incident on a Si atom creates an electron-hole pair.

Types of Image Sensors | Image Sensing

Image Sensor: A Closer Look

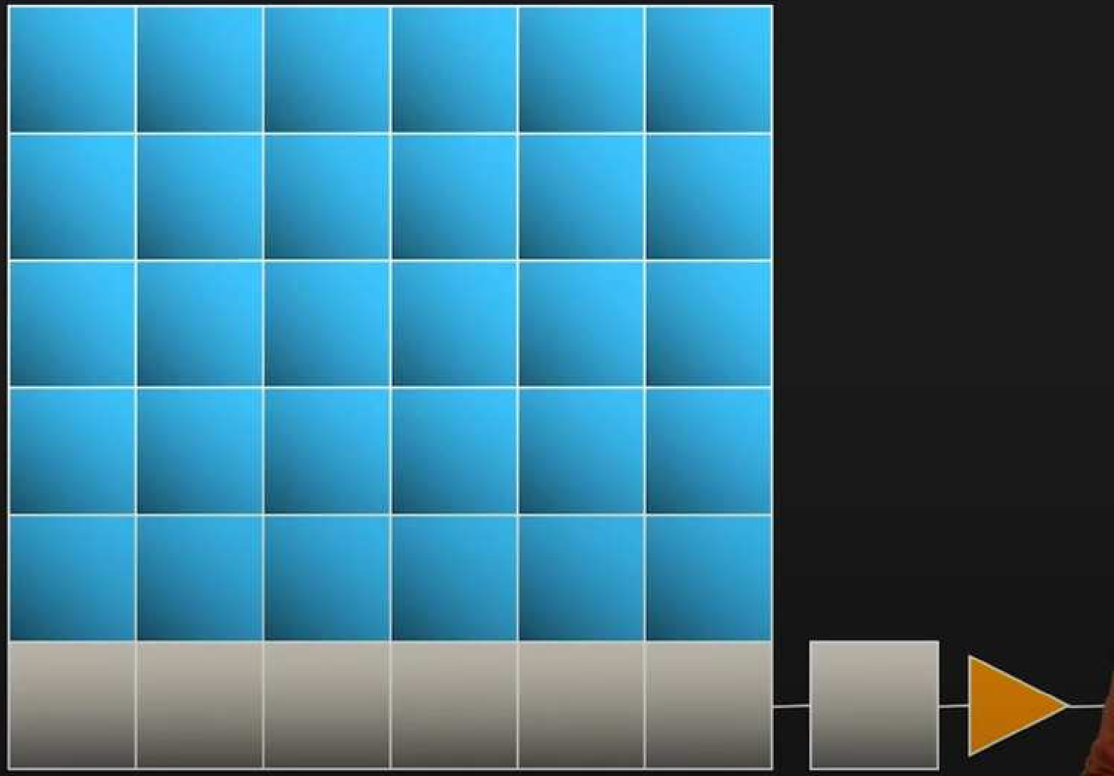


18 Megapixels

1:45 / 10:35

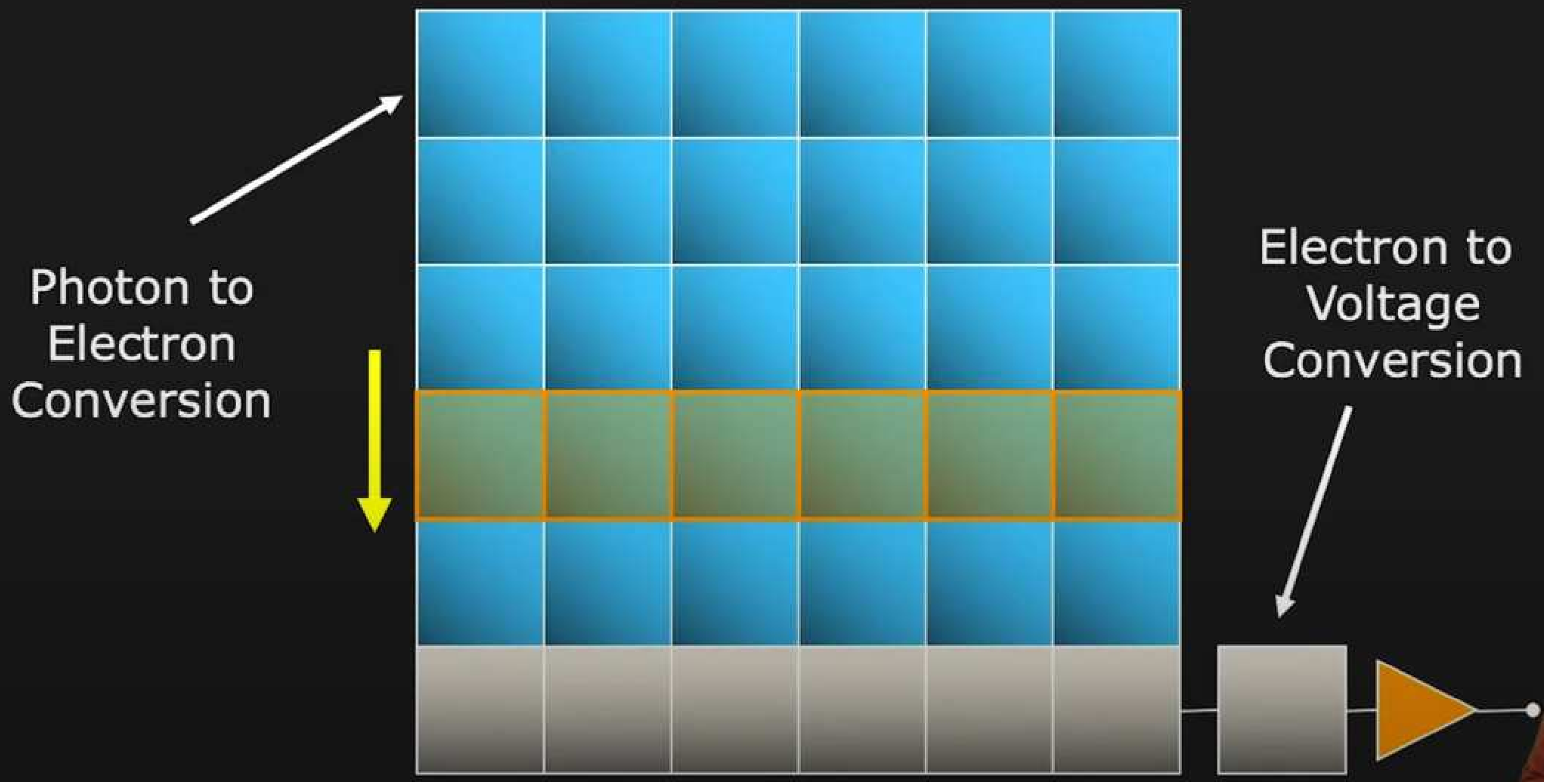
Types of Image Sensors: CCD

CCD: Charge Coupled Device



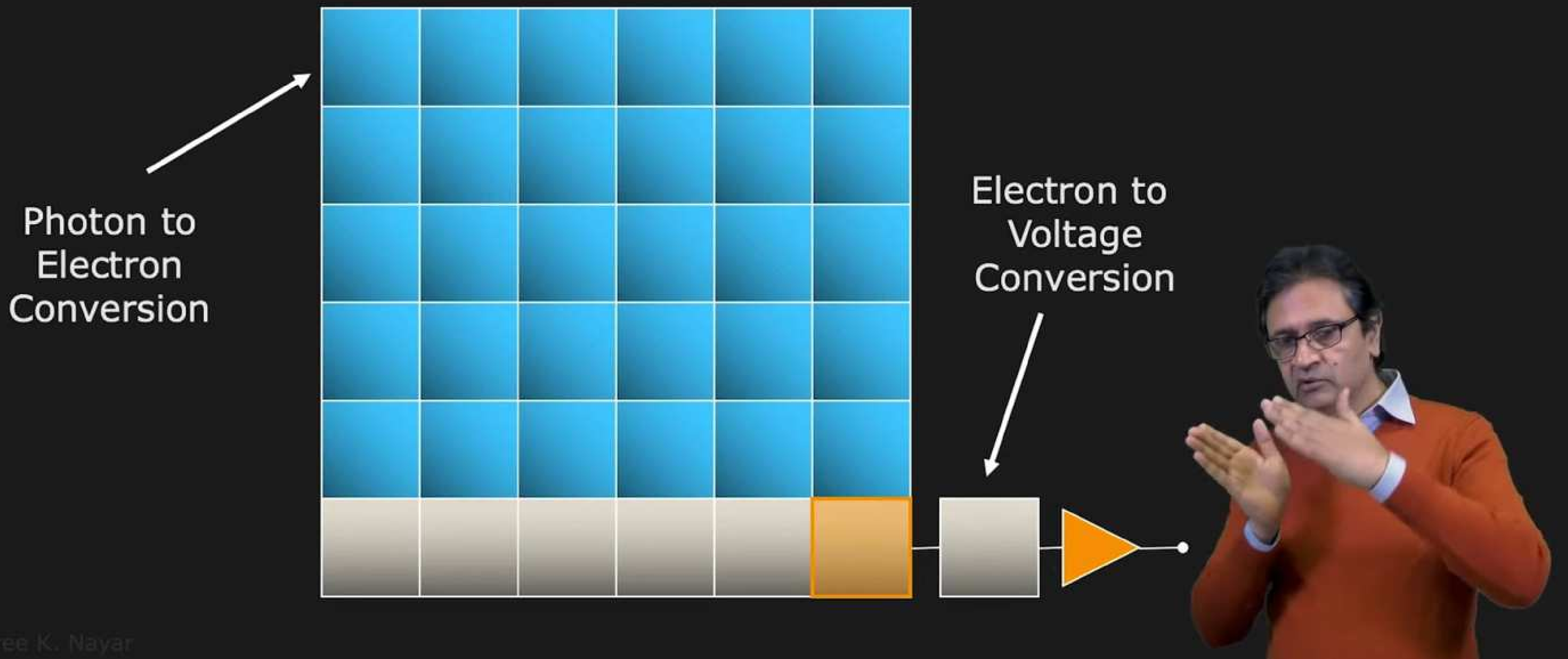
Types of Image Sensors: CCD

CCD: Charge Coupled Device



Types of Image Sensors: CCD

CCD: Charge Coupled Device



Types of Image Sensors: CMOS

CMOS: Complimentary Metal-Oxide Semiconductor

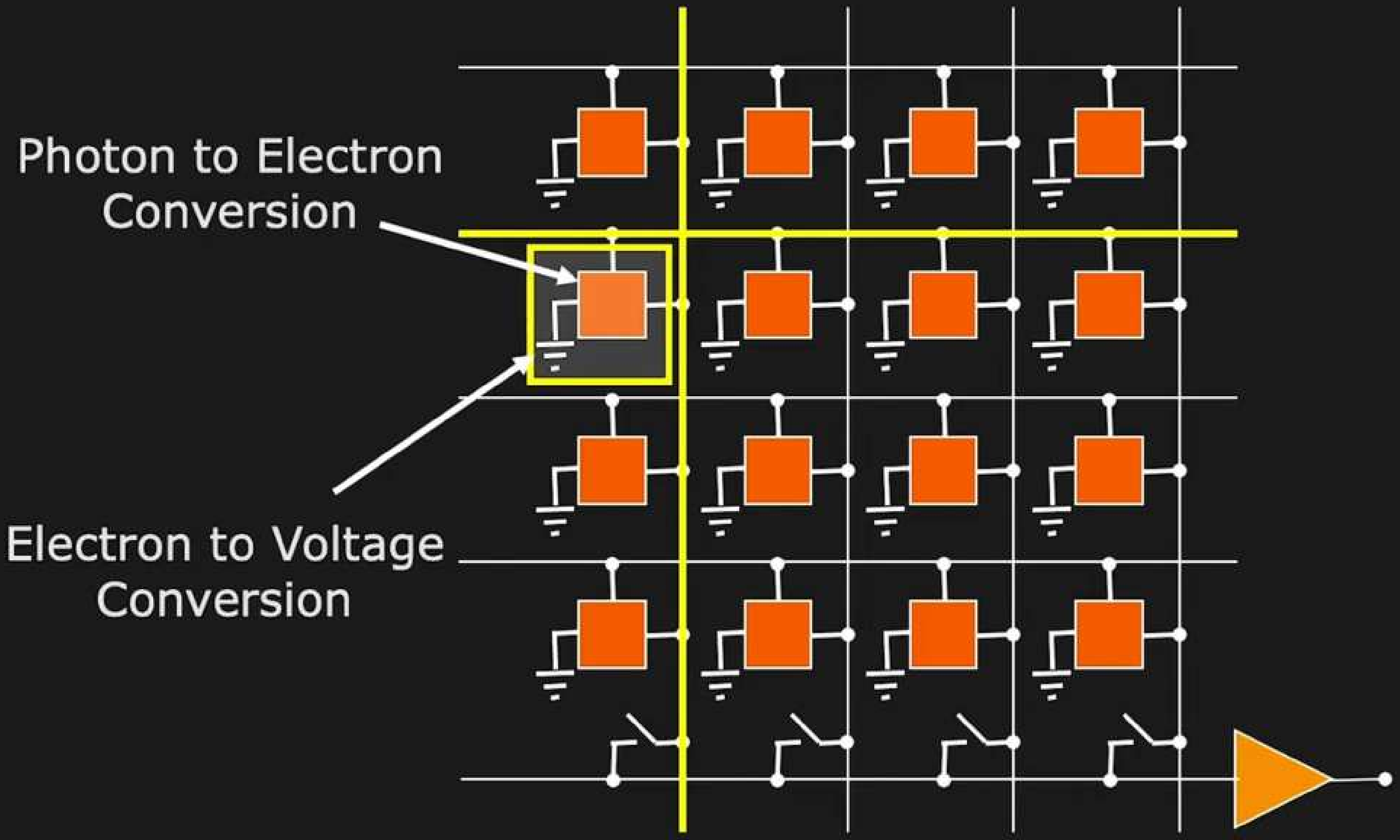
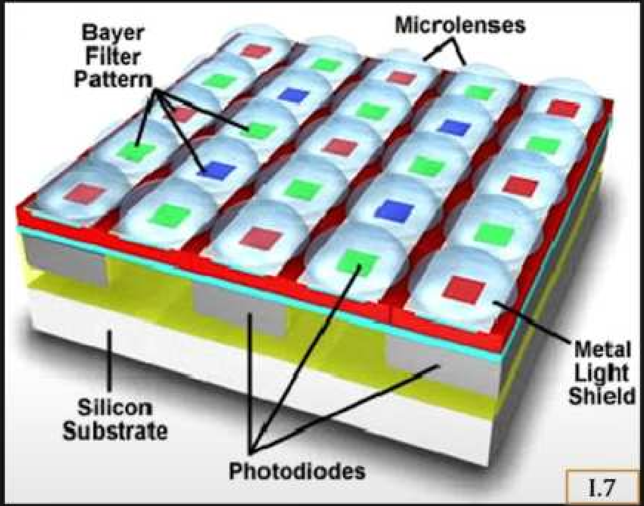
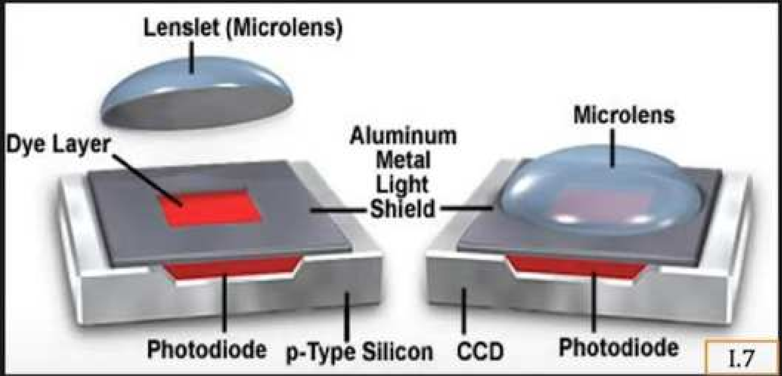


Image Sensor: A Closer Look

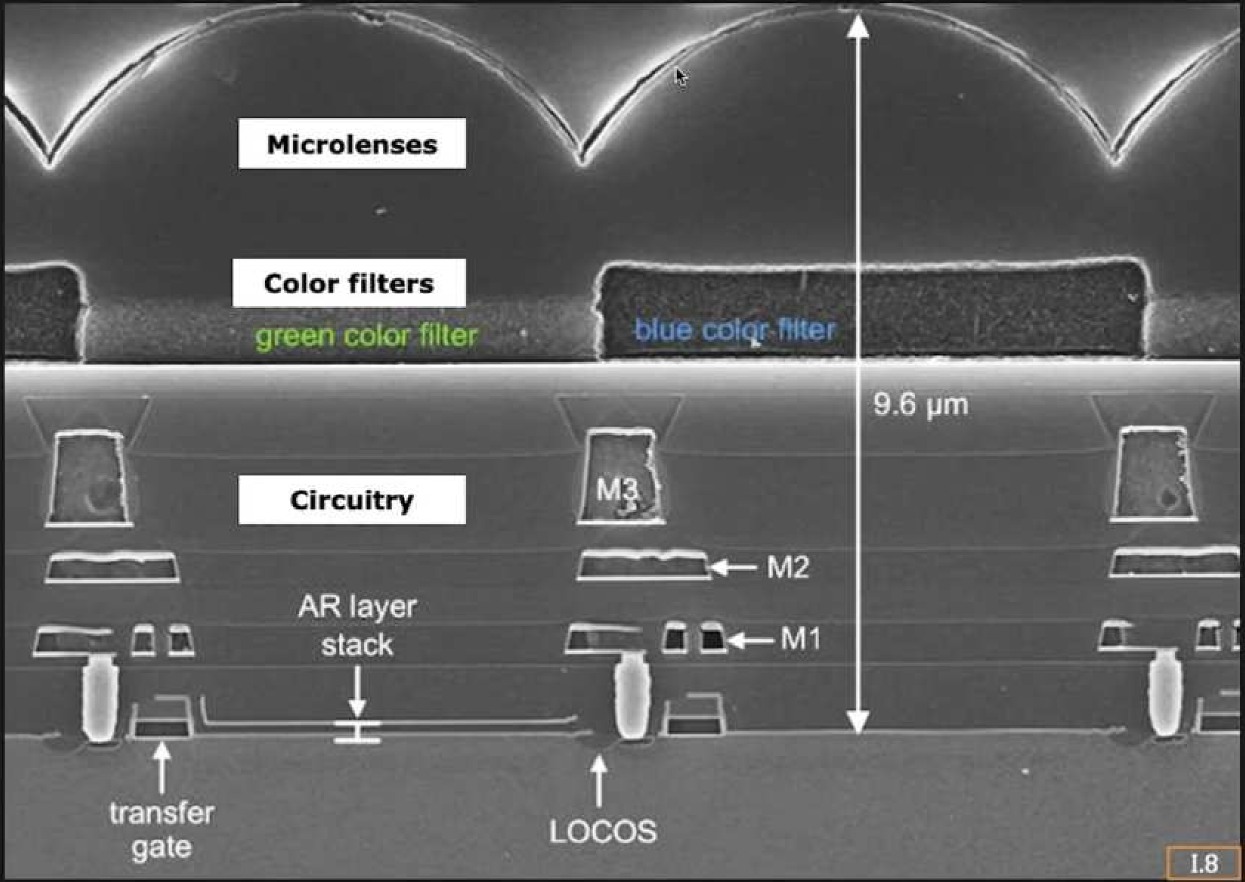


I.7



I.7

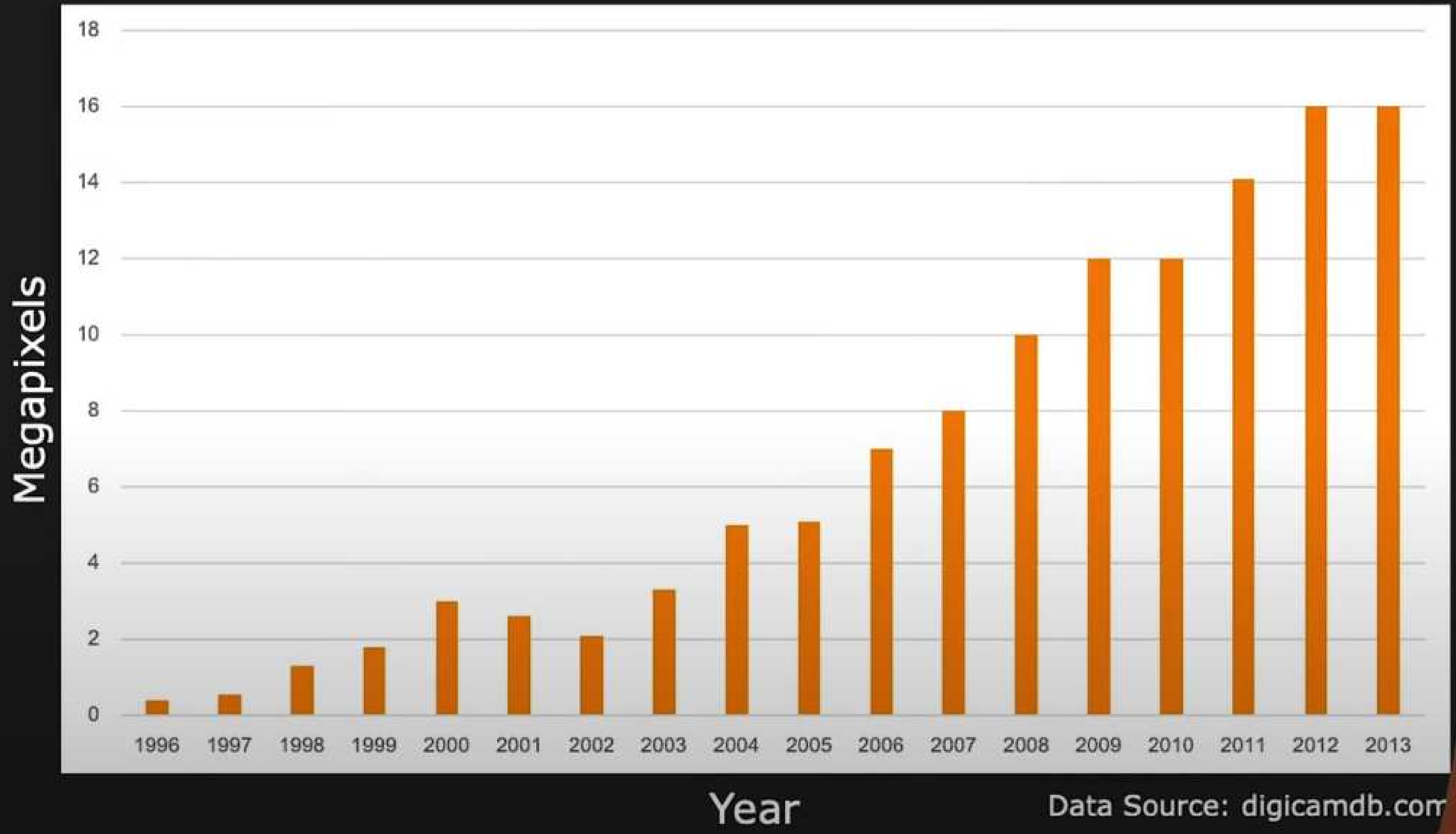
Image Sensor: A Closer Look



Cross Section of an Image Sensor
(Scanning Electron Microscope Image)

Image Sensor Resolution

Median Sensor Resolution in Consumer Cameras



Data Source: digicamdb.com

Noise in Image Sensors

Noise: Unwanted modification of signal during capture, conversion, transmission, processing.

- **Photon Shot Noise** (Scene Dependent)
  - Quantum nature of light
  - Random arrival of photons
- **Readout Noise** (Scene Independent)
  - Electronic Noise: Pre analog-to-digital conversion
  - Quantization Noise: Post analog-to-digital conversion
- **Other Sources** (Scene Independent)
  - Dark Current Noise: Thermally generated electrons
  - Fixed Pattern Noise: Defective pixels

Noise in Image Sensors

Noise: Unwanted modification of signal during capture, conversion, transmission, processing.

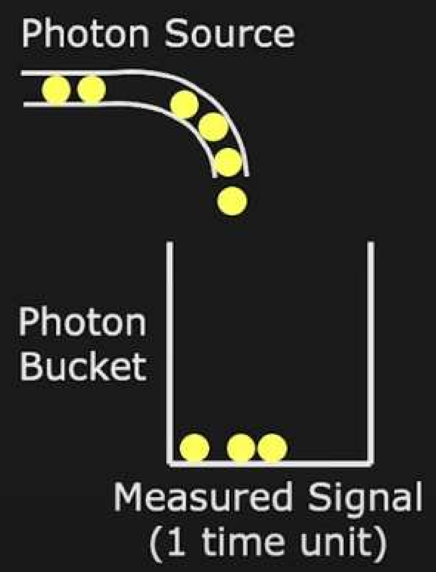
- **Photon Shot Noise** (Scene Dependent)
  - Quantum nature of light
  - Random arrival of photons
- **Readout Noise** (Scene Independent)
  - Electronic Noise: Pre analog-to-digital conversion
  - Quantization Noise: Post analog-to-digital conversion
- **Other Sources** (Scene Independent)
  - Dark Current Noise: Thermally generated electrons
  - Fixed Pattern Noise: Defective pixels

Noise in Image Sensors

Noise: Unwanted modification of signal during capture, conversion, transmission, processing.

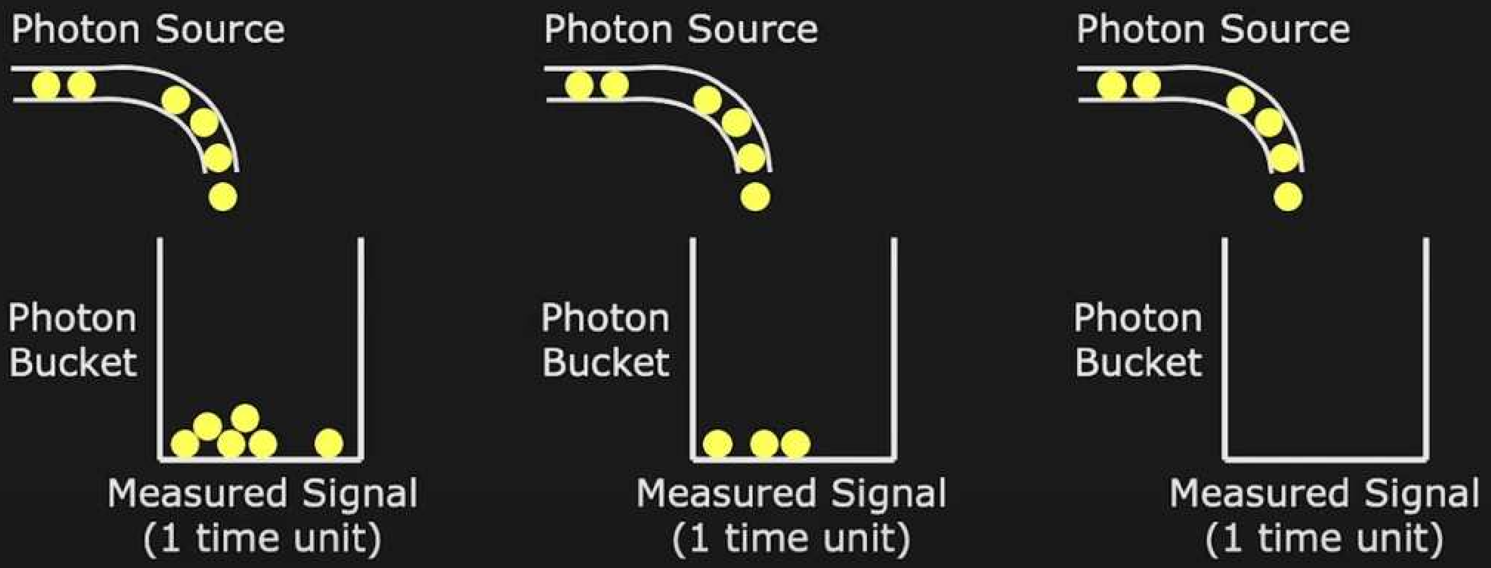
- **Photon Shot Noise** (Scene Dependent)
  - Quantum nature of light
  - Random arrival of photons
- **Readout Noise** (Scene Independent)
  - Electronic Noise: Pre analog-to-digital conversion
  - Quantization Noise: Post analog-to-digital conversion
- **Other Sources** (Scene Independent)
  - Dark Current Noise: Thermally generated electrons
  - Fixed Pattern Noise: Defective pixels

Photon Shot Noise



Average Photon Flux (Per Unit Time) = 3 Photons

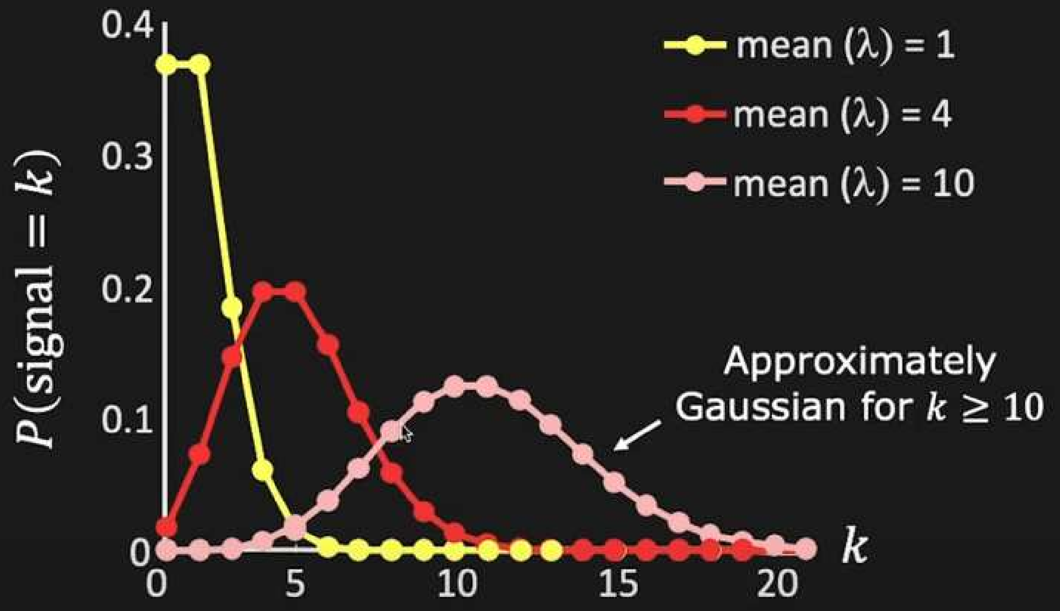
Photon Shot Noise



Average Photon Flux (Per Unit Time) = 3 Photons

Variation Due to Random Generation of Photons

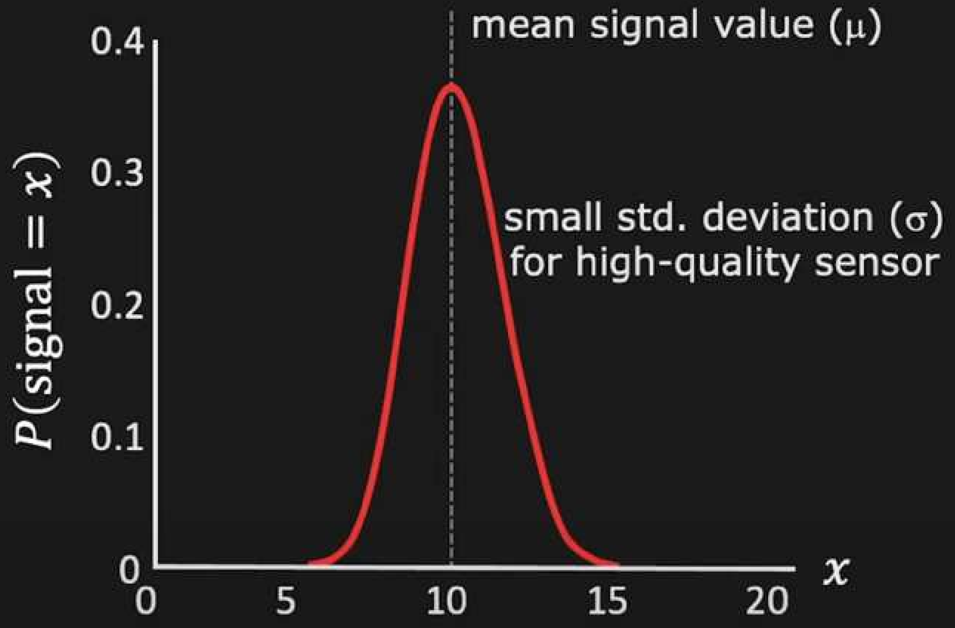
Photon Noise: Poisson Distribution



$$\boxed{P(\text{signal} = k) = \frac{\lambda^k e^{-\lambda}}{k!}}$$

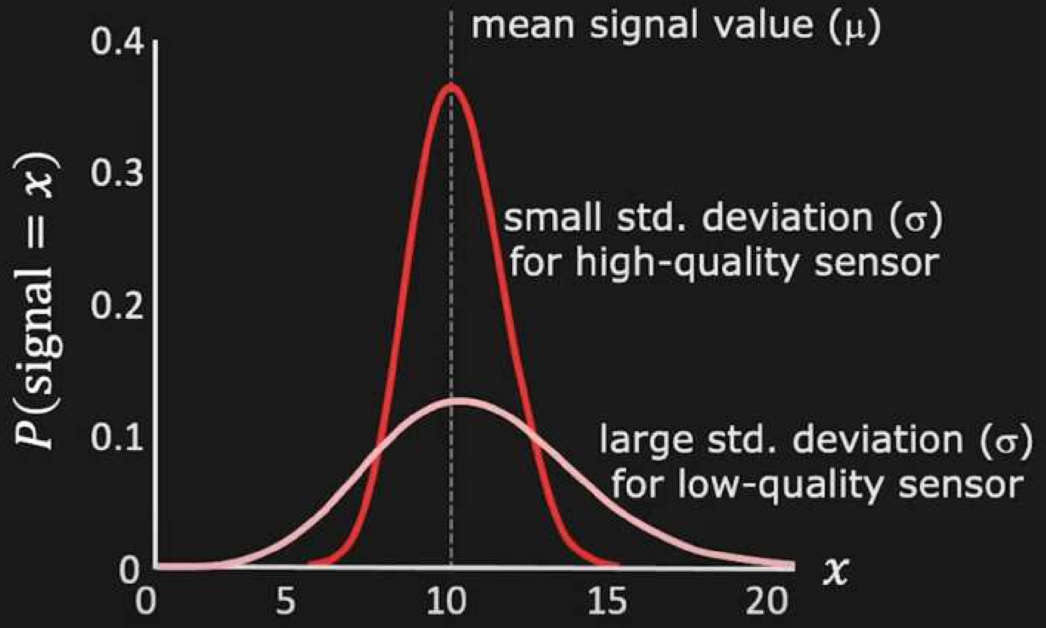
$Var\,[\text{signal}] = Mean\,[\text{signal}] \Rightarrow$ Scene Dependent Noise

Read Noise: Gaussian Distribution



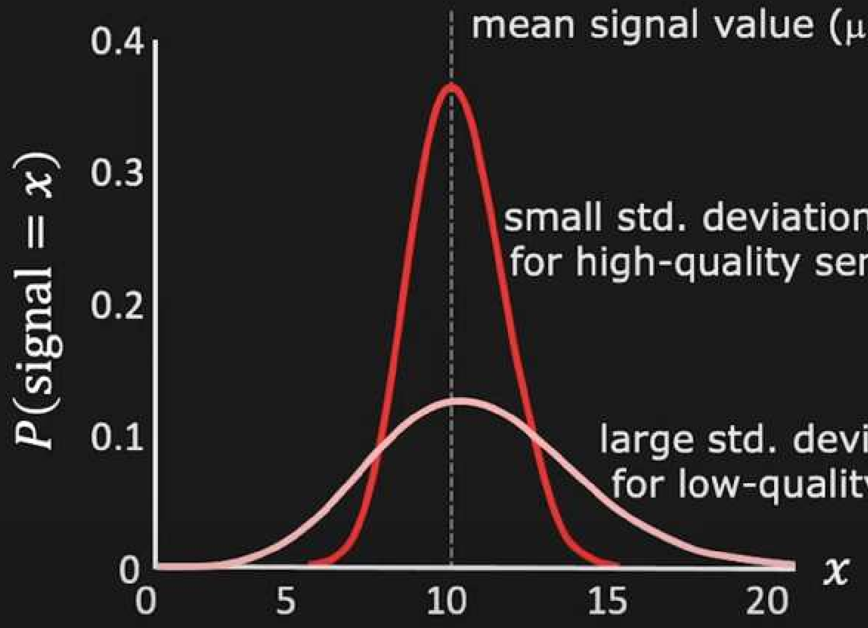
$$P(\text{signal} = x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Read Noise: Gaussian Distribution



$$P(\text{signal} = x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Read Noise: Gaussian Distribution

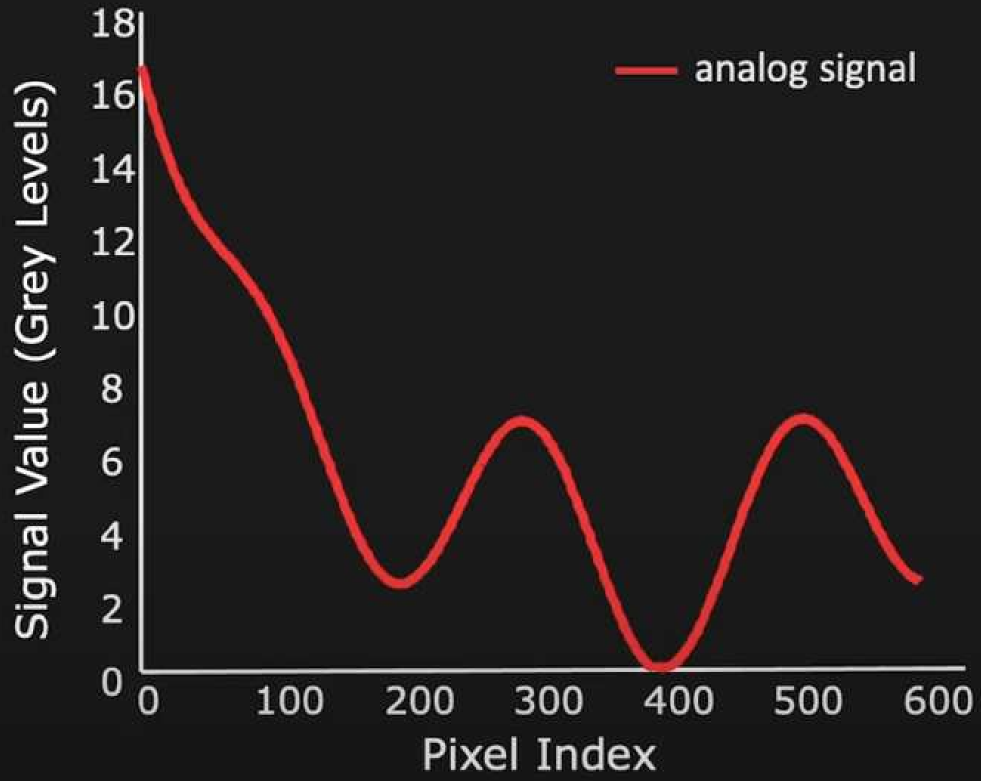


$$P(\text{signal} = x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Depends on Sensor Quality (Scene Independent)

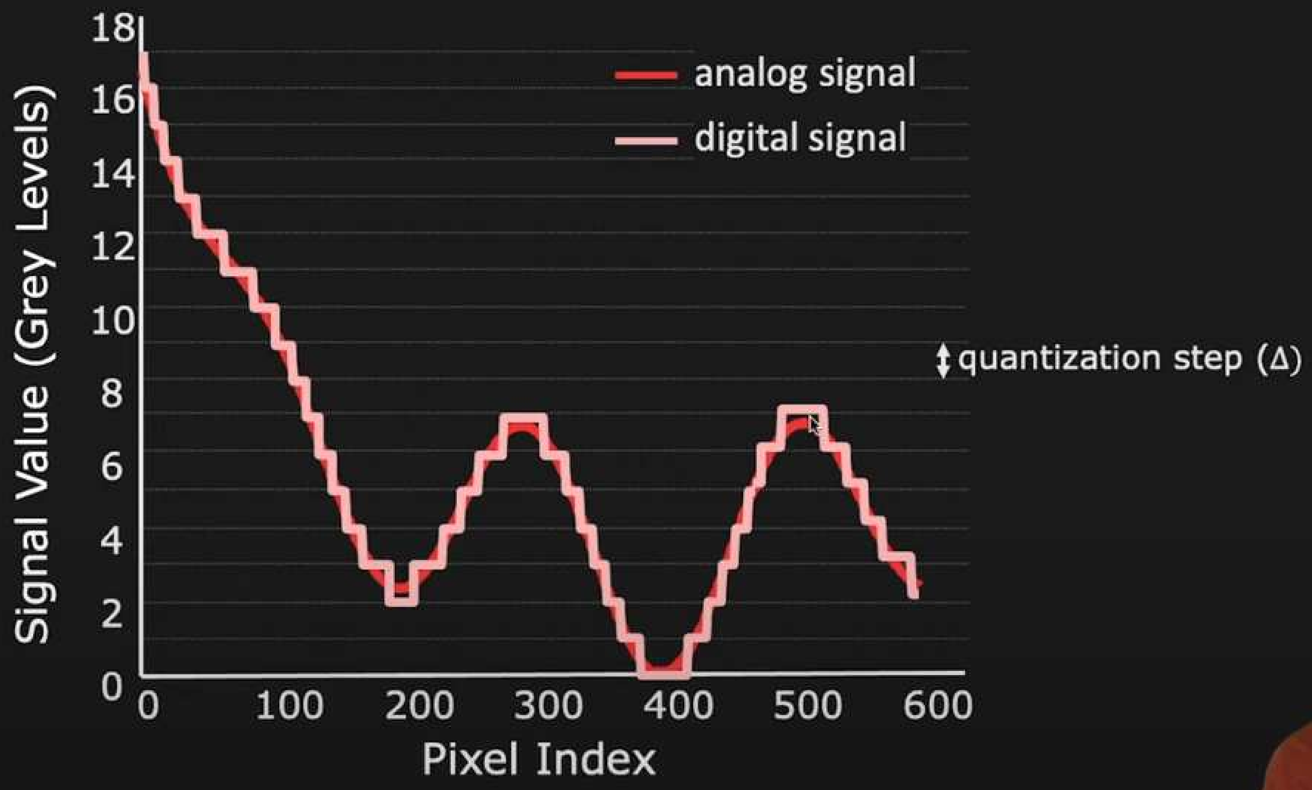
Resolution, Noise, Dynamic Range | Image Sensing

Quantization Noise

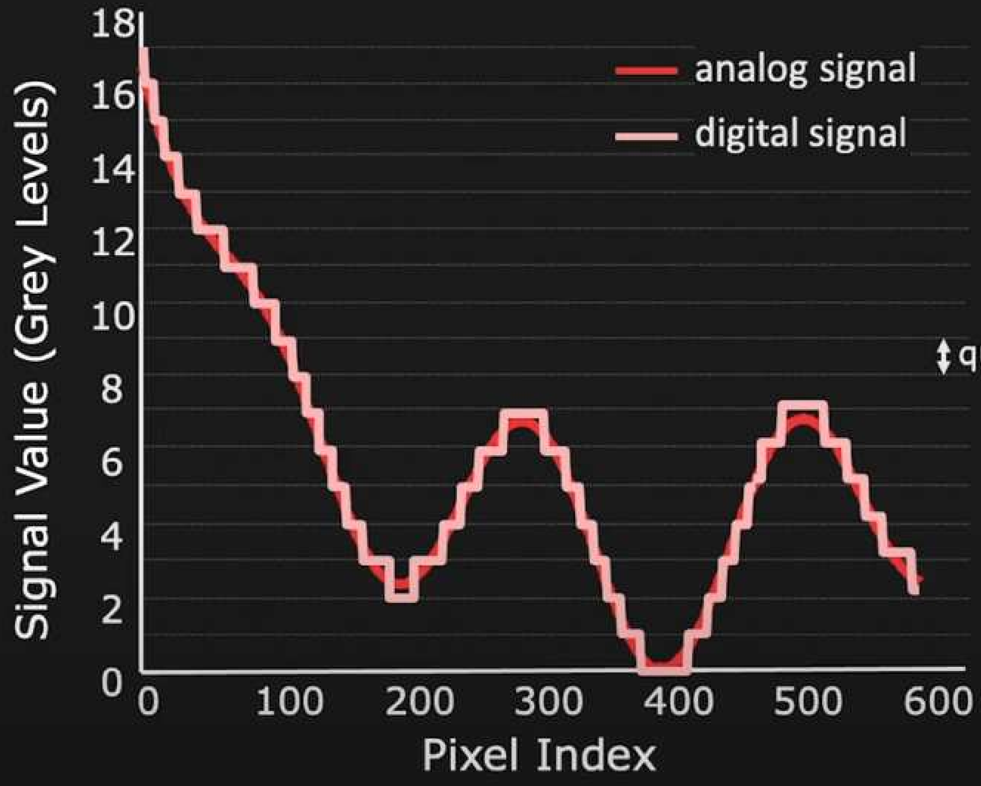


8:03 / 13:38 • Quantization Noise ›

Quantization Noise



Quantization Noise



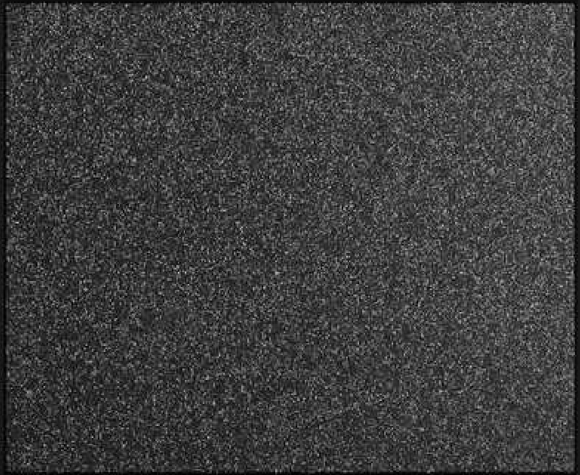
quantization step (Δ)

$$Var = \frac{\Delta}{12}$$

Negligible in Modern Sensors
Due to High Intensity Resolution (12-14 bits)

Other Noise Sources

Dark Current Noise (Thermal)

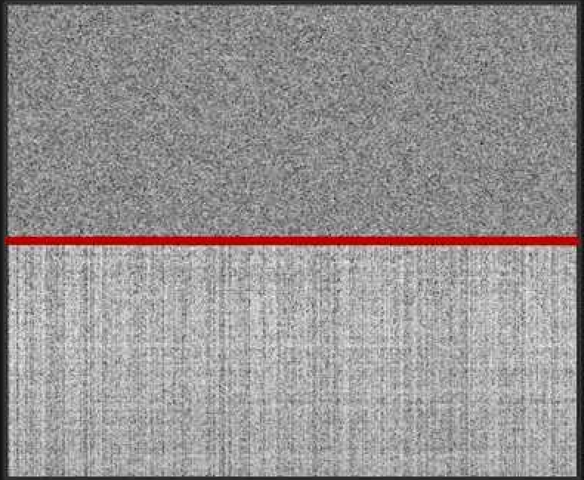


Follows Poisson Distribution

Significant Only for Long (>2 min) Exposures (Astronomy)

Fixed Pattern Noise (Defective Pixels)

Random Noise



Fixed Pattern Noise

Can be Reduced by Dark Frame Subtraction

Sensor Dynamic Range

$$\boxed{\text{Dynamic Range} = 20 \log\left(\frac{B_{max}}{B_{min}}\right)} \text{ decibels (dB)}$$

B_{max}: The maximum possible photon energy (full potential well)

B_{min}: The minimum detectable photon energy (in the presence of noise)

Sensor	B_{max}: B_{min}	dB
Human Eye	1,000,000:1	120
HDR Display	200,000:1	106
Digital Camera	4096:1	72.2
Film Camera	2948:1	66.2
Digital Video	45:1	33.1

Sensor Dynamic Range

$$\boxed{\text{Dynamic Range} = 20 \log \left(\frac{B_{max}}{B_{min}}\right)} \text{ decibels (dB)}$$

B_{max}: The maximum possible photon energy
(full potential well)

B_{min}: The minimum detectable photon energy
(in the presence of noise)

Quantum Efficiency

Incoming light can vary in Wavelength (λ)

Pixel Intensity (Electron Flux)

Incoming Light (Photon Flux)

$$I \longleftarrow \boxed{q(\lambda)} \longleftarrow p(\lambda)$$

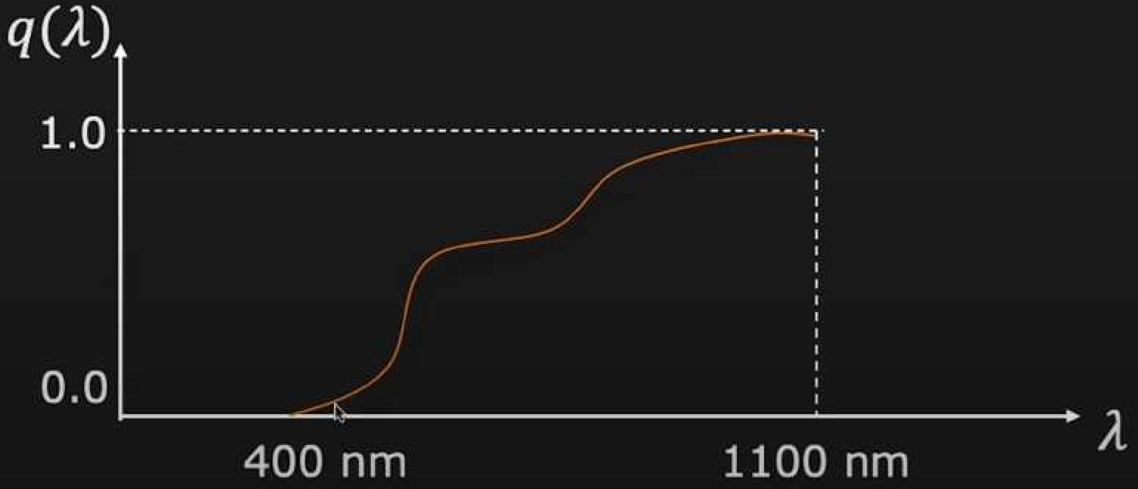
Silicon Pixel

Quantum Efficiency:

$$q(\lambda) = \frac{\text{Electron Flux Generated}}{\text{Photon Flux of wavelength } \lambda}$$

Quantum Efficiency of Silicon

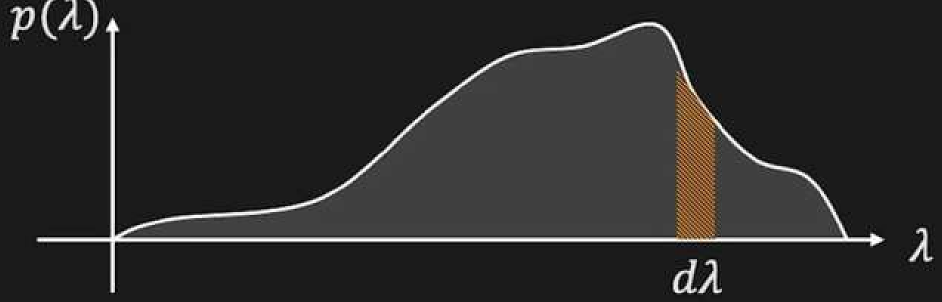
$$\text{Silicon is:} \begin{cases} \text{Transparent} & \text{for } \lambda > 1100\, nm \\ \text{Opaque} & \text{for } \lambda < 400\, nm \end{cases}$$



Assume Monochromatic Light $\lambda = \lambda_i$ with flux $p(\lambda_i)$:

$$\boxed{I = q(\lambda_i)p(\lambda_i)}$$

What if incoming light has Spectral Distribution $p(\lambda)$?



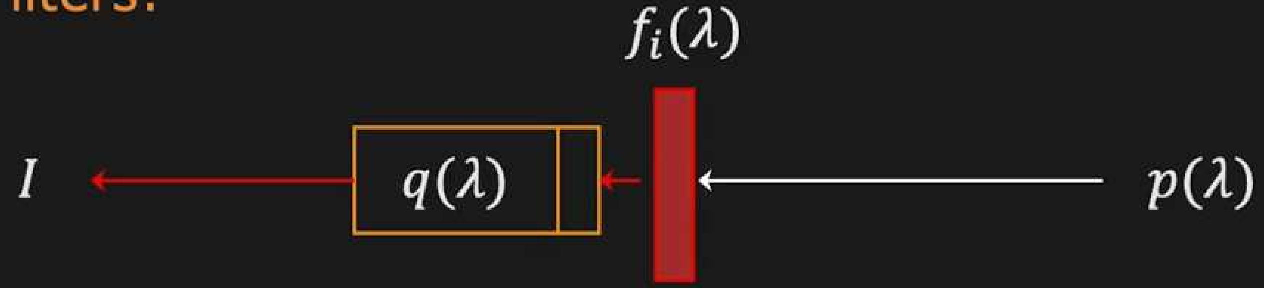
Flux (Energy) of Light with wavelength between λ and $\lambda + d\lambda$ is $p(\lambda)d\lambda$. Therefore:

$$\boxed{I = \int_0^\infty q(\lambda)p(\lambda)d\lambda}$$

$$I = \int_0^\infty q(\lambda)p(\lambda)d\lambda$$

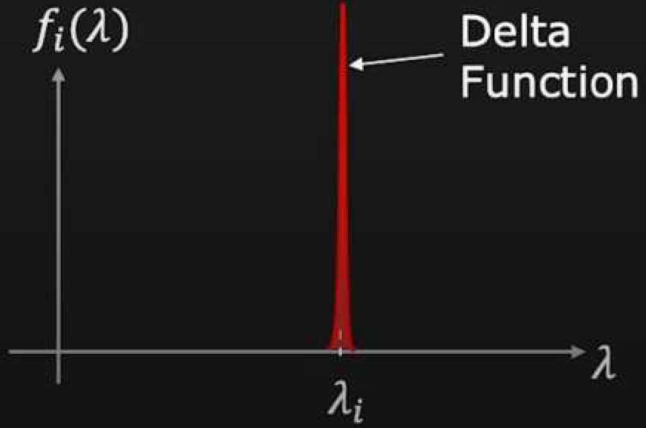
If we know I and $q(\lambda)$, can we find $p(\lambda)$?

Use Filters:



Let $f_i(\lambda) = \delta(\lambda - \lambda_i)$

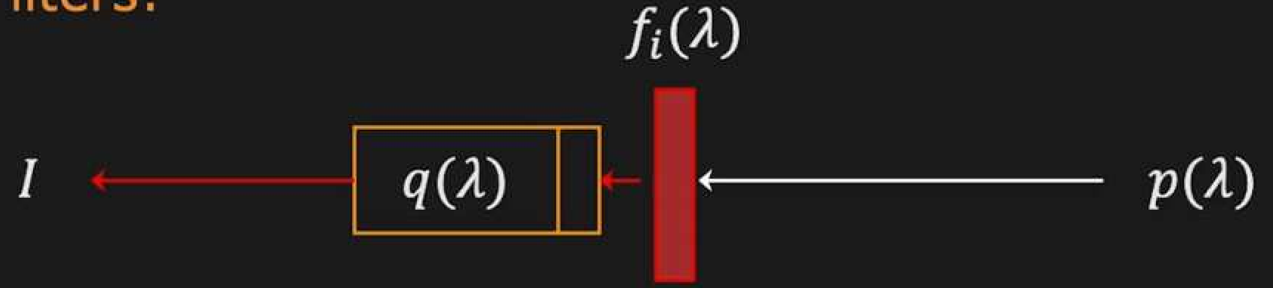
Note: $\int_0^\infty \delta(\lambda - \lambda_i)d\lambda = 1$



$$\boxed{I = \int_0^\infty q(\lambda)p(\lambda)d\lambda}$$

If we know I and $q(\lambda)$, can we find $p(\lambda)$?

Use Filters:



$$I = \int_0^\infty q(\lambda)p(\lambda)f_i(\lambda)d\lambda \quad = \int_0^\infty q(\lambda)p(\lambda)\delta(\lambda - \lambda_i)d\lambda$$

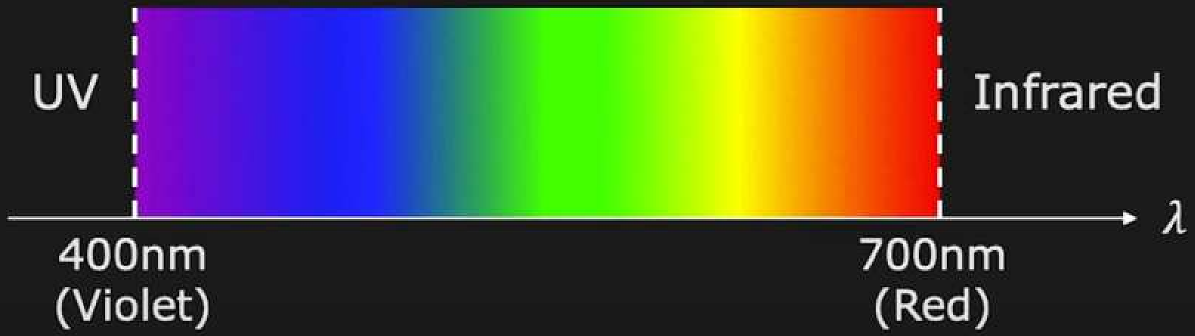
$$\boxed{I = q(\lambda_i)p(\lambda_i)}$$

How many filters do we need to recover $p(\lambda)$? Infinite?

What is “Color”?

Human Response to different wavelengths

Visible light:

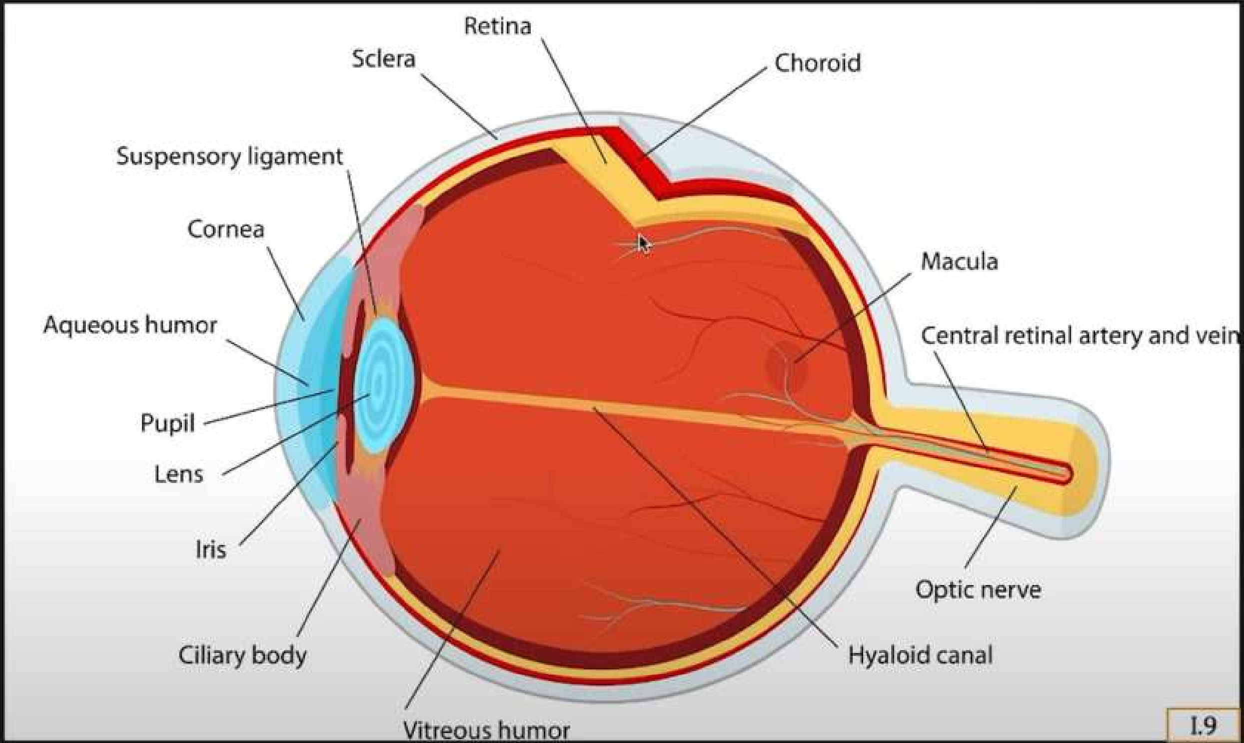


Do We recover spectral distribution $p(\lambda)$?

Sensors in the human eye: Rods & Cones
Neurochemical Sensors (3 types)

Sensing Color | Image Sensing

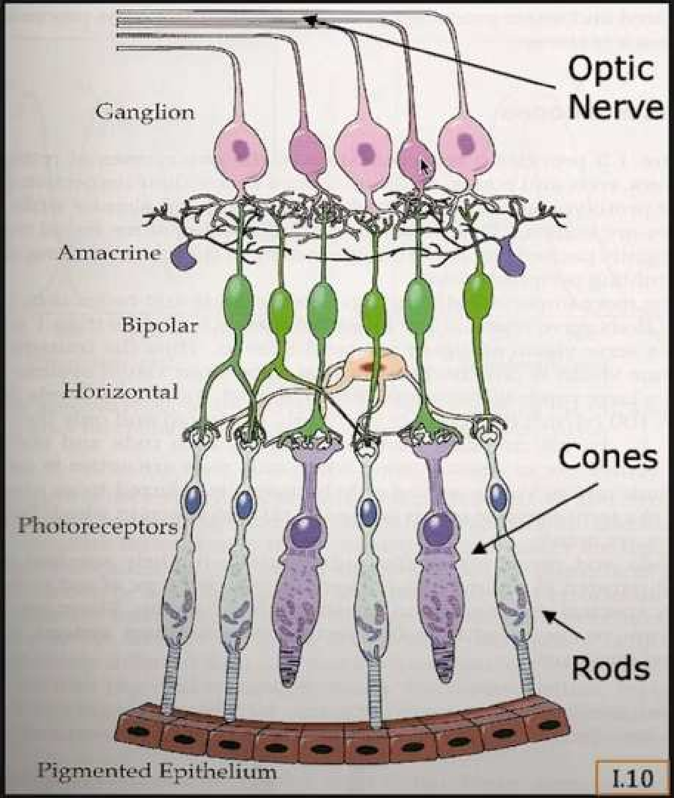
The Human Eye



I.9

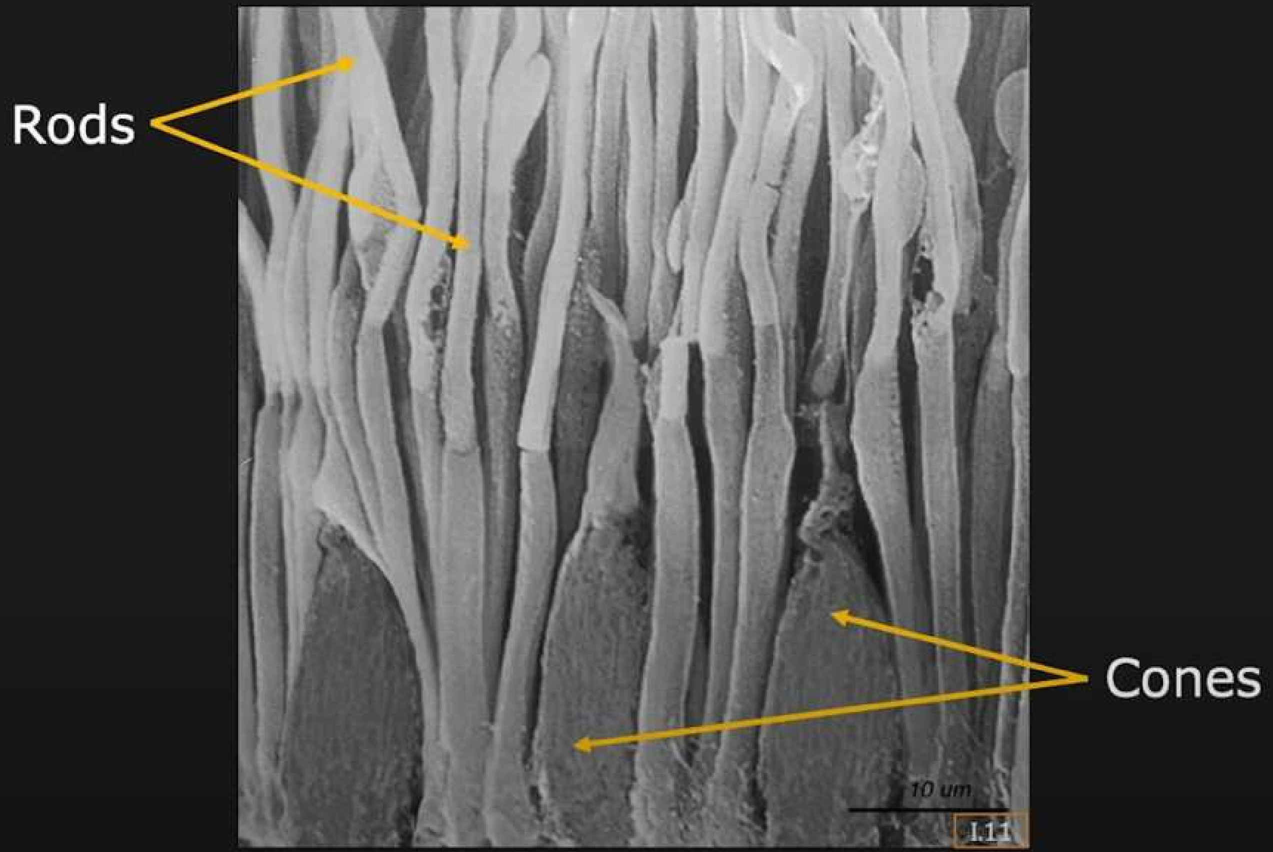
7:35 / 18:35 • The Human Eye ›

A Cross-Section of the Retina

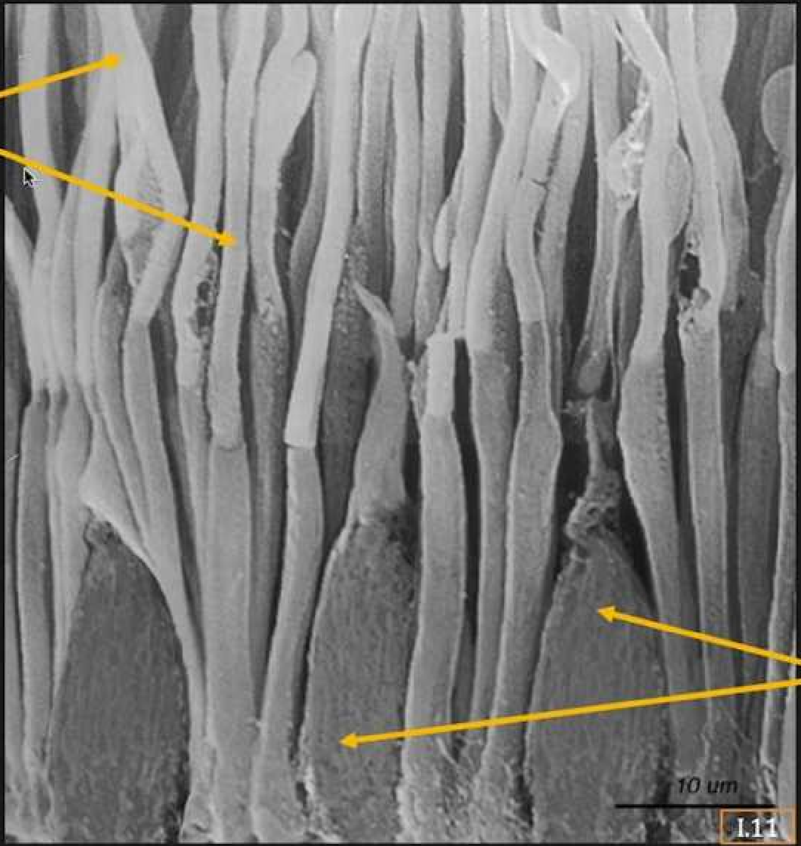


[Fairchild 200

The Eye's Pixels

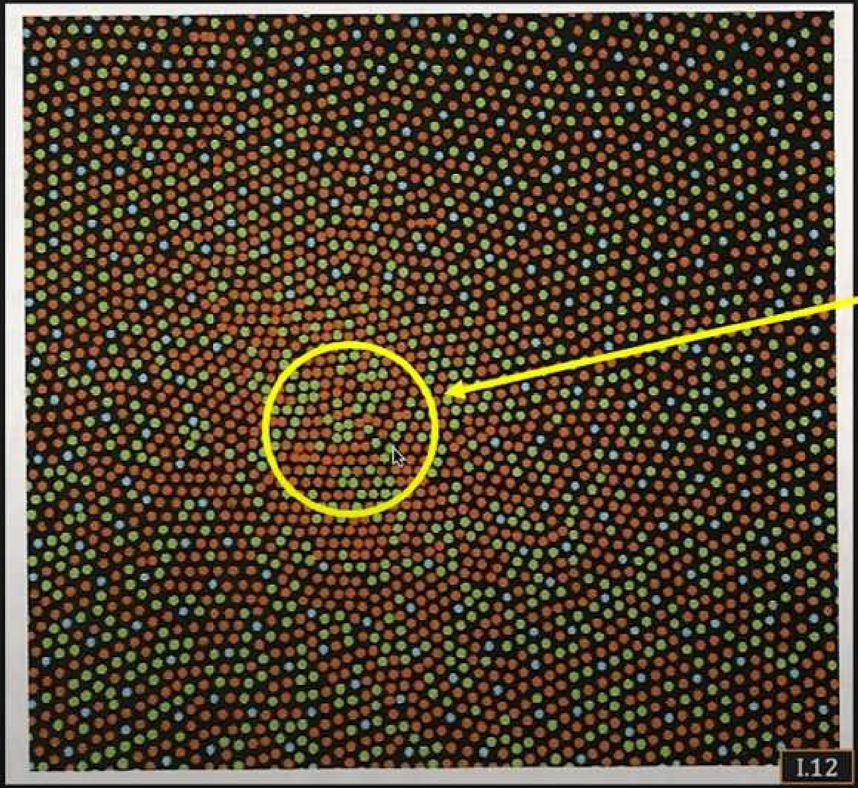


The Eye's Pixels



Distribution of Cones in Human Retina

Three types of cones for sensing red, green, blue

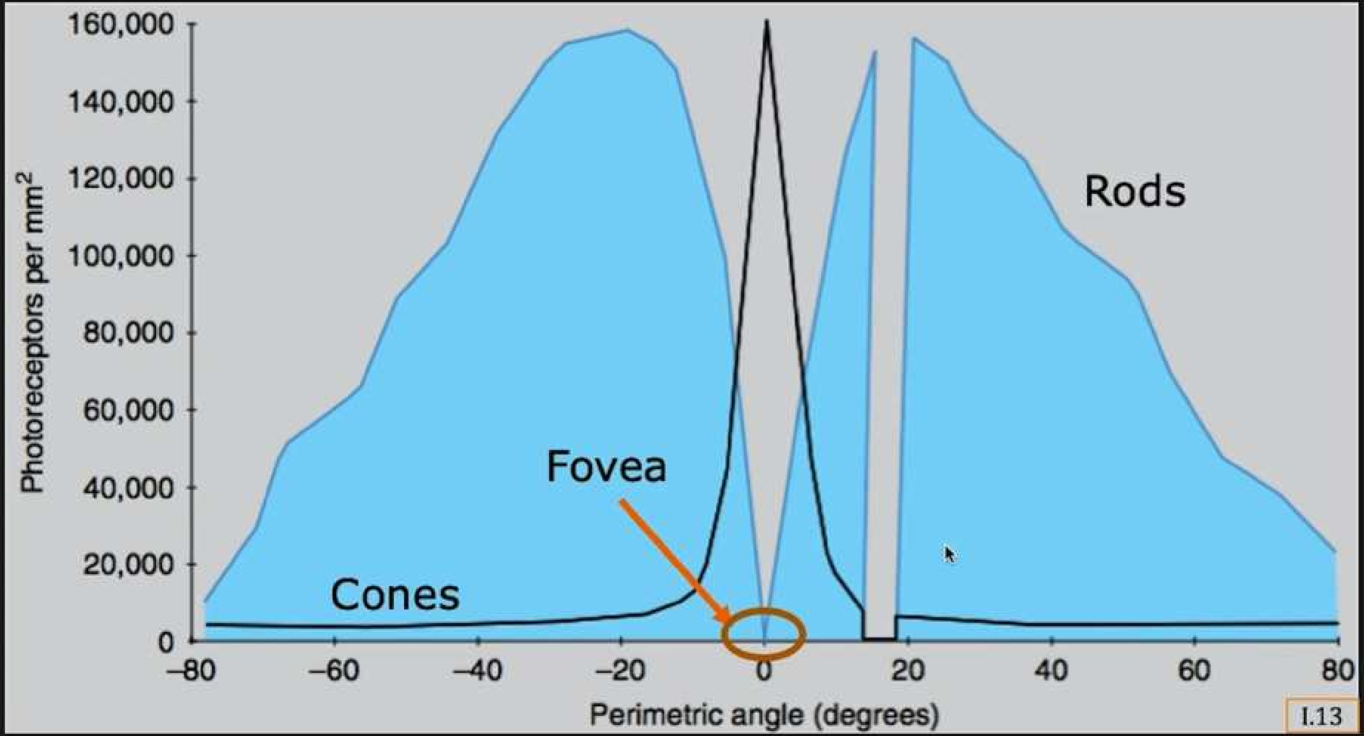


Fovea

[William 19

Resolution of Rods and Cones

On average, ~120M rods and ~7M cones per retina.



[Fairchild 20

Tristimulus Values

Three Intensities (R, G, B):

$$R = \int_{-\infty}^{\infty} h_r(\lambda) p(\lambda) d\lambda$$

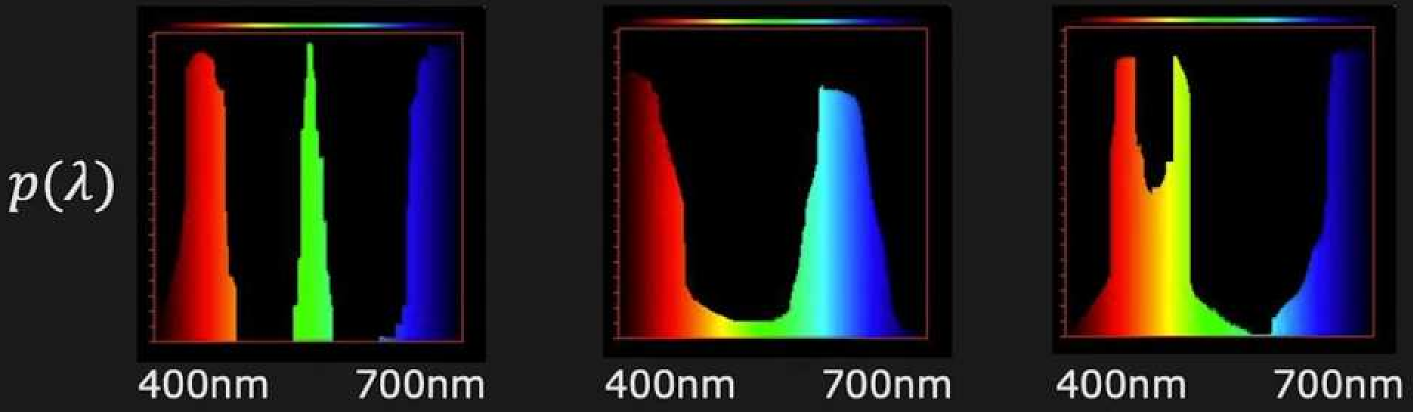
$$G = \int_{-\infty}^{\infty} h_g(\lambda) p(\lambda) d\lambda$$

$$B = \int_{-\infty}^{\infty} h_b(\lambda) p(\lambda) d\lambda$$

Metamers

Metamers: Different $p(\lambda)$ that produce same (R, G, B)

For example, different spectra:



Same perceived color:

$(R, G, B) = (115, 60, 108) =$

The Mixing of Colors

Young's Experiment on Color Mixture

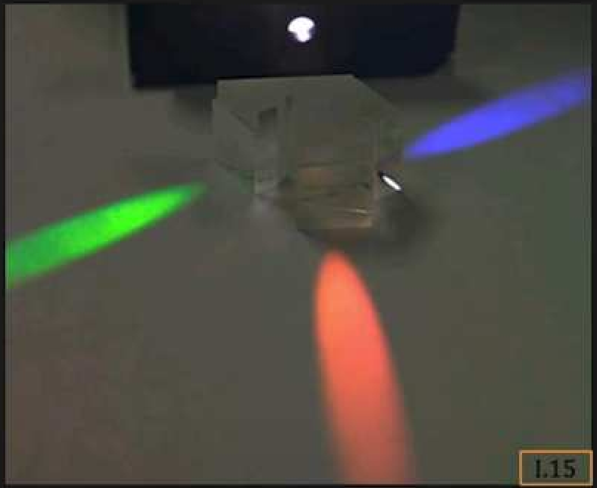
Human Sensation of nearly all colors can be produced using 3 wavelengths!

$$(\lambda_r, \lambda_g, \lambda_b) = (650, 530, 410)nm$$

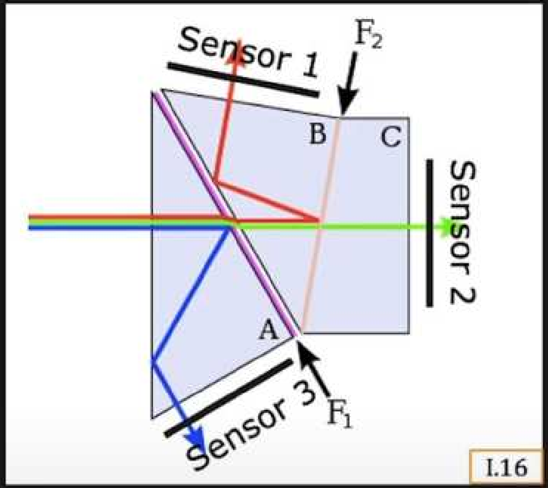
Hence, cameras and displays often use 3 filters:

(red, green, blue)

Sensing Color using Dichroic Prism



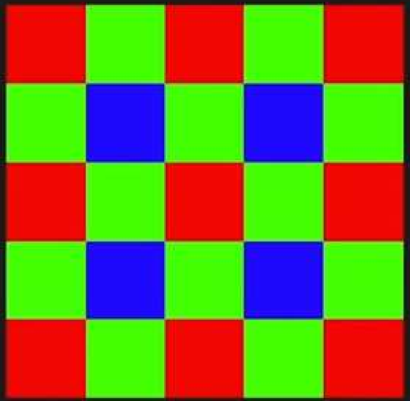
Dichroic Prism



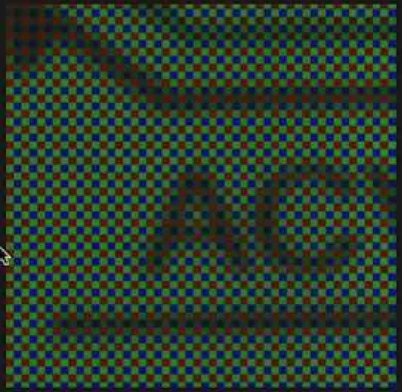
3-CCD Camera using Dichroic Prism

Each Sensor Detects One Color

Sensing Color Using Color Mosaic



Bayer Pattern
(Color Filter Mosaic)



Raw Image

Interpolated Image

Color Filled in by Interpolation (Demosaicing)

Camera Response Function $f(\cdot)$

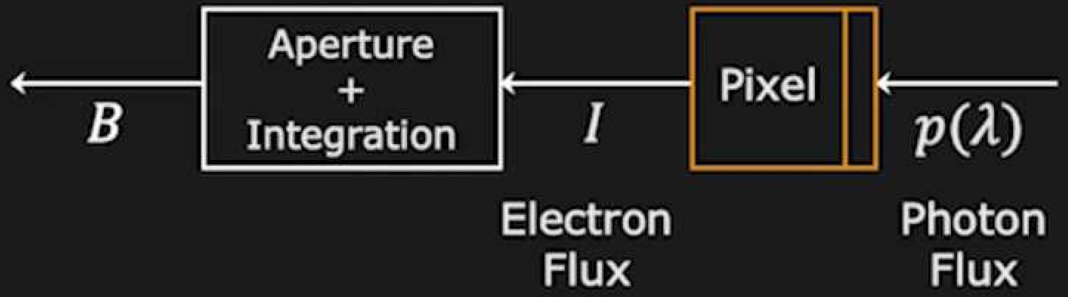
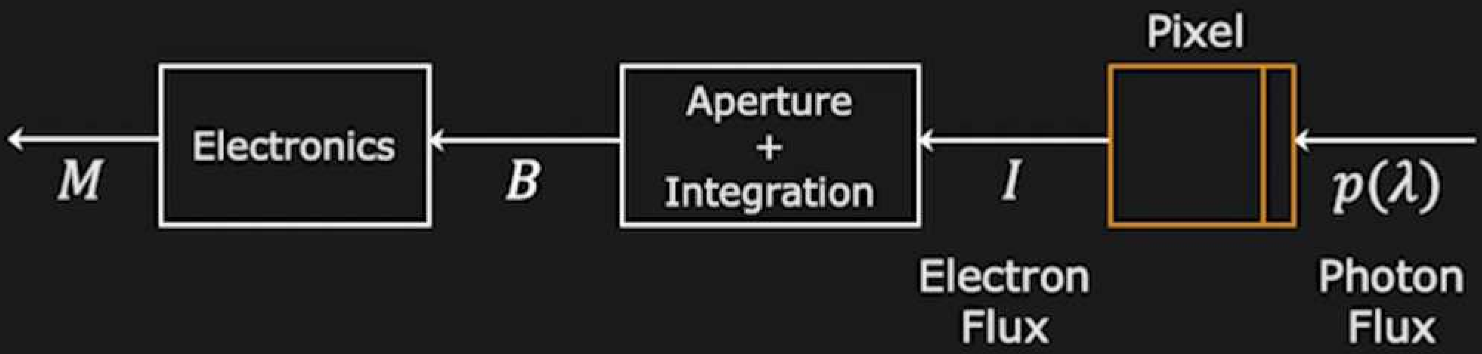


Image Brightness: $\boxed{B = I \cdot \left(\frac{\pi d^2}{4}\right) \cdot t}$

Aperture Area (diameter d)

Integration Time

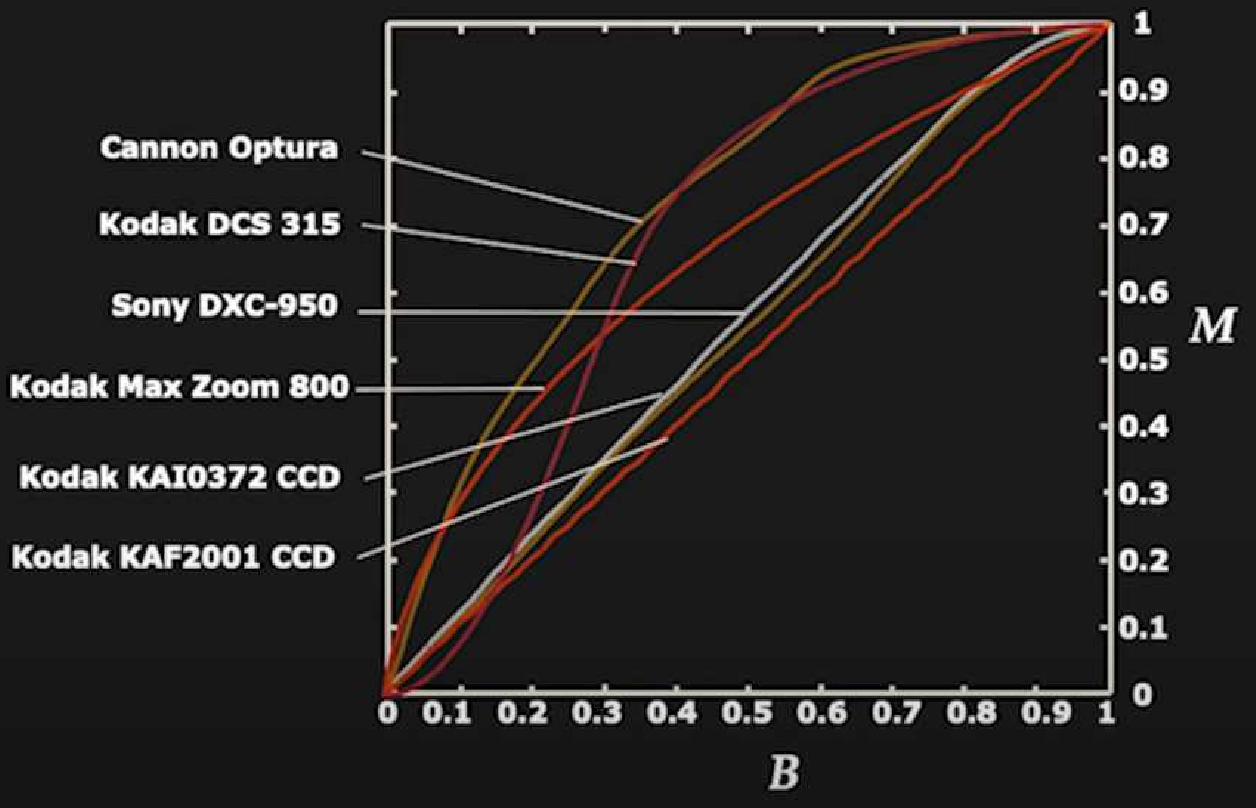
Camera Response Function $f(\cdot)$



Measured Brightness: $\boxed{M = f(B) = f(I \cdot e)}$

where Exposure, $e = \left(\dfrac{\pi d^2}{4}\right) \cdot t$

Camera Response Function $f(\cdot)$



"Gamma Curves"

[Grossberg 2

Radiometric Calibration: Finding $f(\cdot)$

Calibration using a chart:

1. Patches with known reflectance (when uniformly lit)
2. Fit linear segments or curve

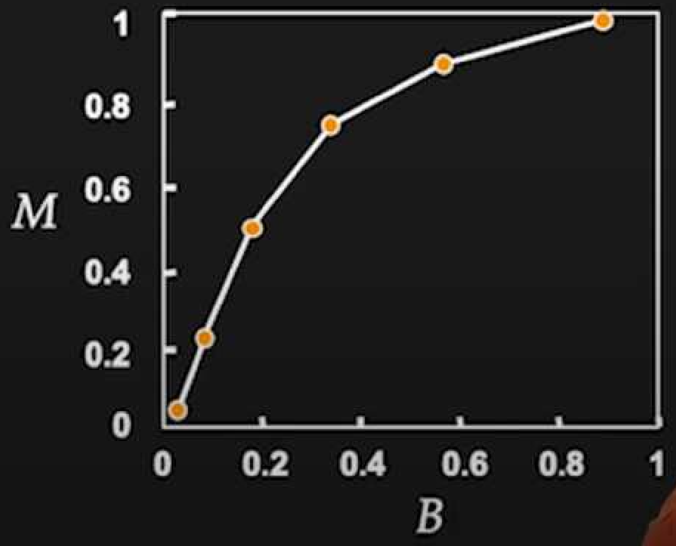
Macbeth Chart

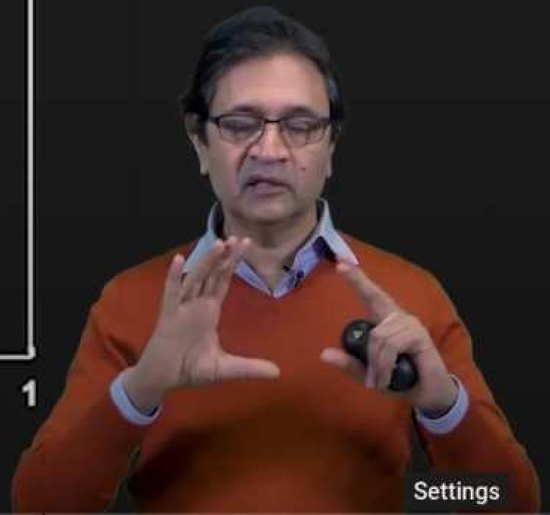
Radiometric Calibration: Finding $f(\cdot)$

Calibration using a chart:

1. Patches with known reflectance (when uniformly lit)
2. Fit linear segments or curve

Macbeth Chart





High Dynamic Range: Multiple Exposures

Assume Camera Response $f(\cdot)$ is Linear

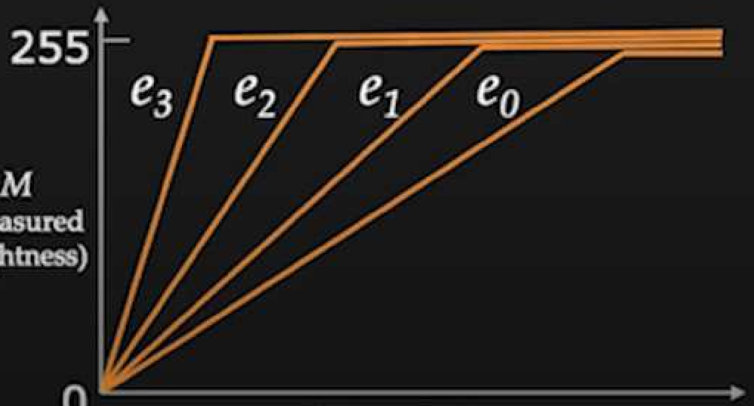
with e_0 $\quad e_1 \quad e_2 \quad e_3$

$$M_0 = \min(e_0 \cdot P, 255)$$

$$M_1 = \min(e_1 \cdot P, 255)$$

$$M_2 = \min(e_2 \cdot P, 255)$$

$$M_3 = \min(e_3 \cdot P, 255)$$

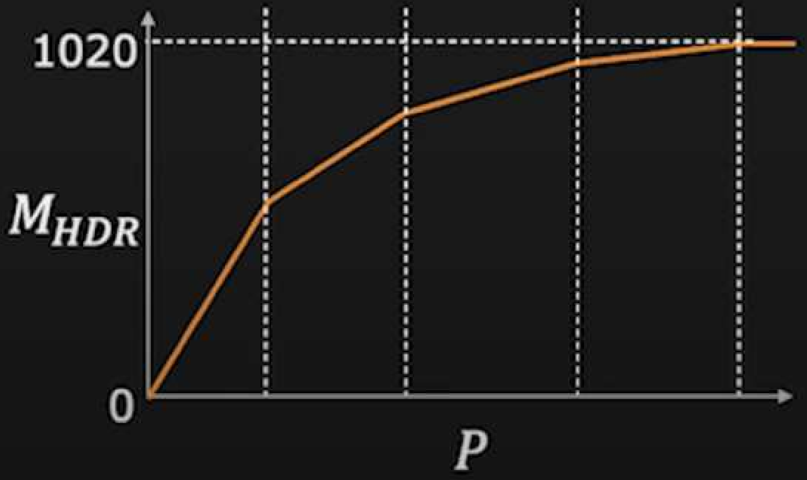


[Nayar 20

High Dynamic Range: Multiple Exposures

Aggregate Image: $M_{HDR} = M_0 + M_1 + M_2 + M_3$

Camera Response $f(\cdot)$ for Aggregate Image:

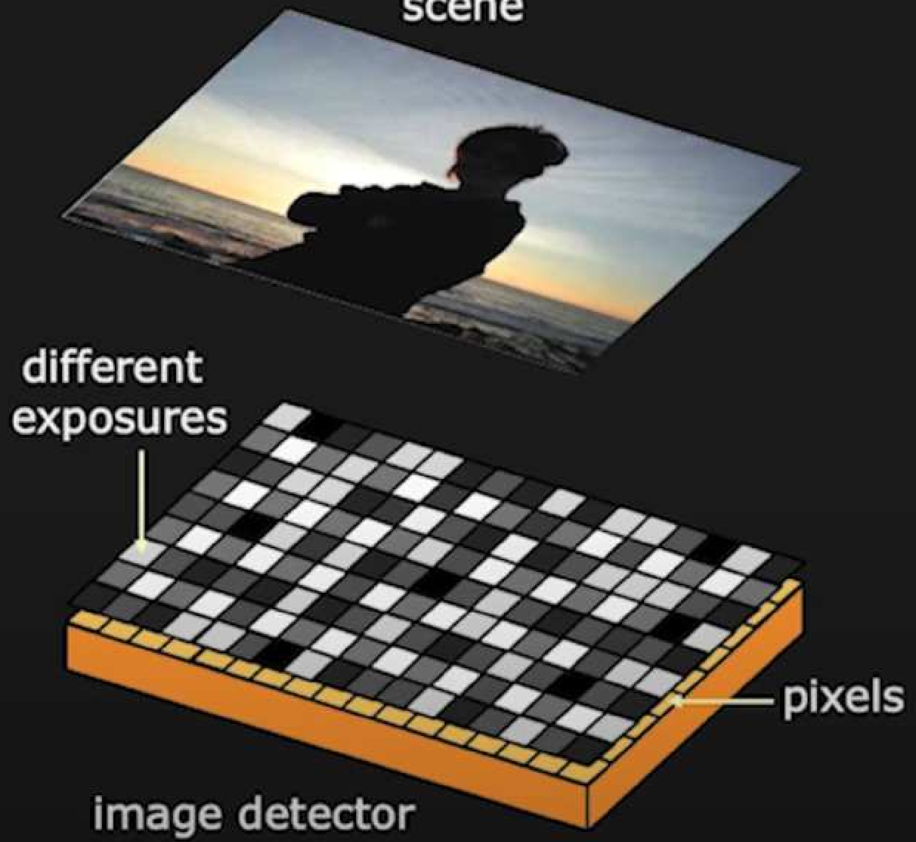


[Nayar 2000

The Motion Problem

iPhone 4 HDR Mode

Single Shot HDR Imaging

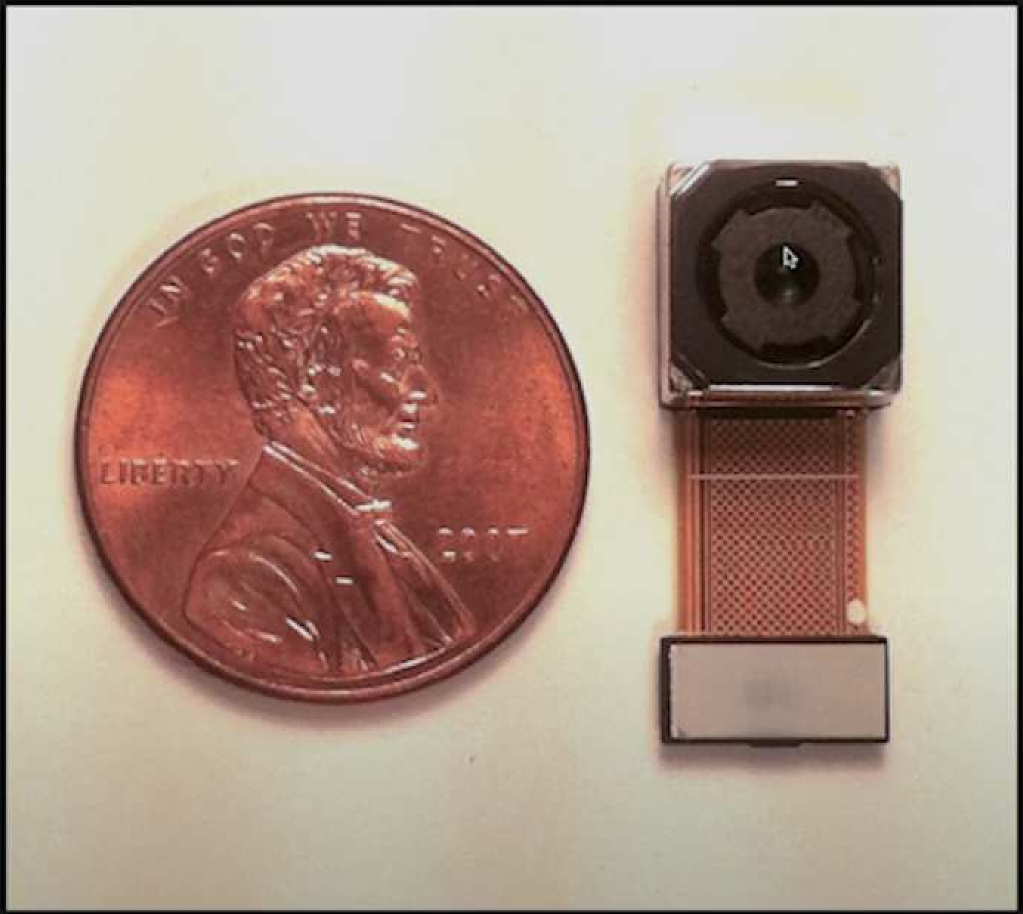


Assorted Pixels

[Nayar 2

Good Old Camera

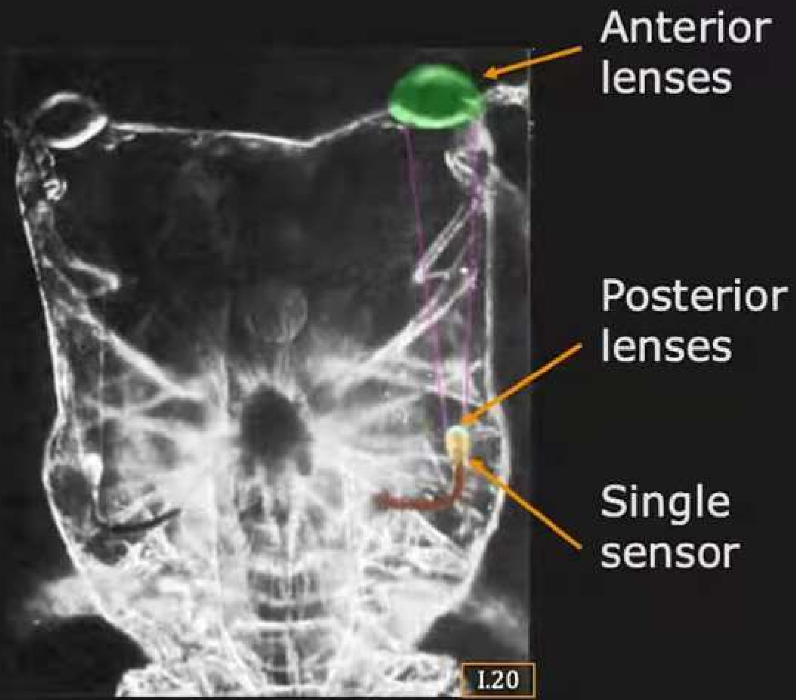
Camera Response and HDR Imaging | Image Sensing



Settings

16:18 / 16:34 • Single Shot HDR Imaging ›

The Curious Eye of Copilia

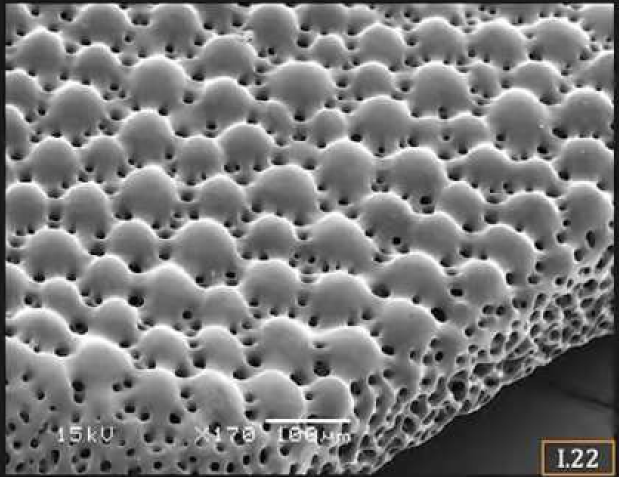


[Gregory 1

It's All Eyes

I.21

Brittle Star
(Ophiocoma wendtii)



I.22

SEM image of the lenses

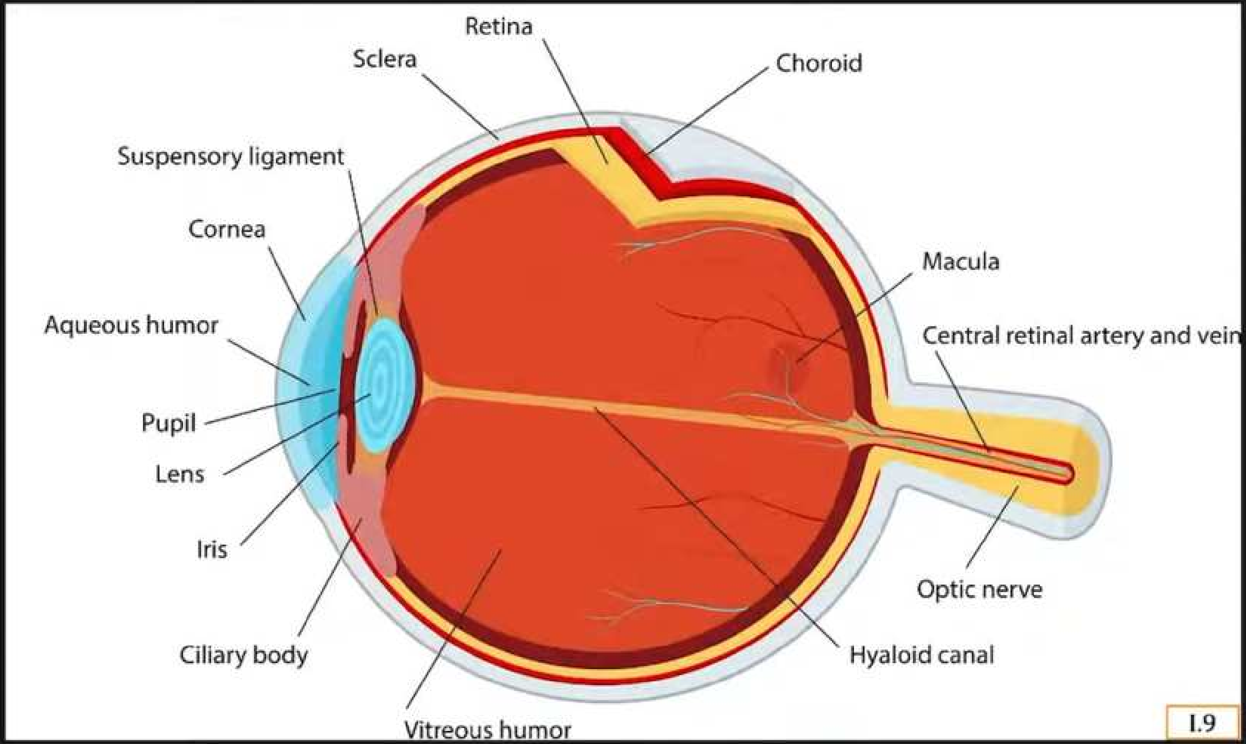
The entire body is covered with lenses

[Aizenberg

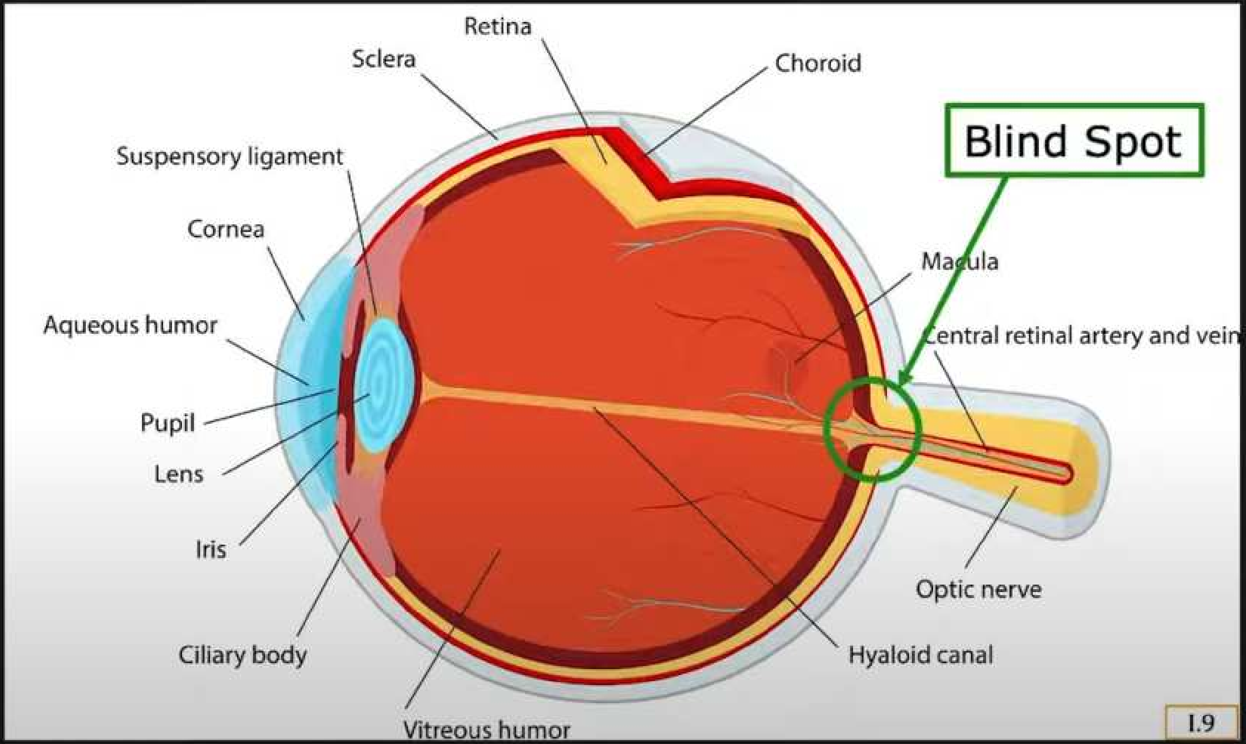
Camouflage in Octopus

[Hanlon 2(

Blind Spot

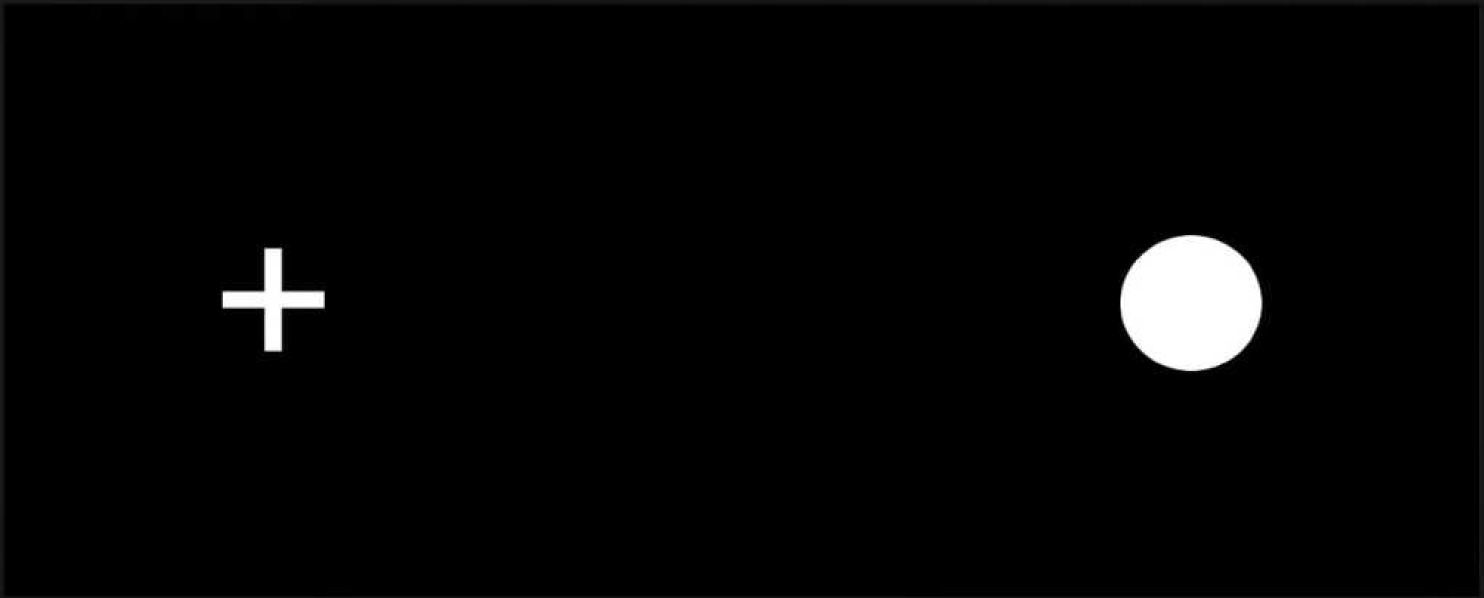

Retina
Sclera
Choroid
Suspensory ligament
Cornea
Macula
Aqueous humor
Central retinal artery and vein
Pupil
Lens
Iris
Optic nerve
Ciliary body
Hyaloid canal
Vitreous humor
I.9

Blind Spot


Retina
Sclera
Choroid
Blind Spot
Suspensory ligament
Cornea
Macula
Aqueous humor
Central retinal artery and vein
Pupil
Lens
Iris
Optic nerve
Ciliary body
Hyaloid canal
Vitreous humor
I.9

Nature's Image Sensors | Image Sensing

Blind Spot



Close your left eye and look at the cross with your right eye.
Move your head slowly towards or away from the image,
until at some point the white disc disappears.

EYE AND BRAIN

5:52 / 7:05 • Blind Spot Experiment ›

Binary Images

Binary Image: Can have only two values (0 or 1).

Simple to process and analyze.

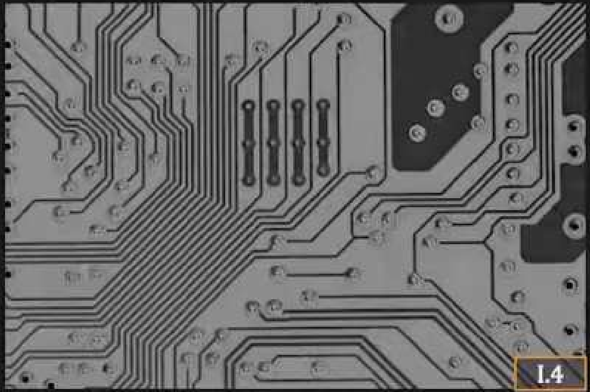
Making Binary Images

Binary Image $b(x,y)$: Usually obtained from Gray-level image $g(x,y)$ by **Thresholding.**

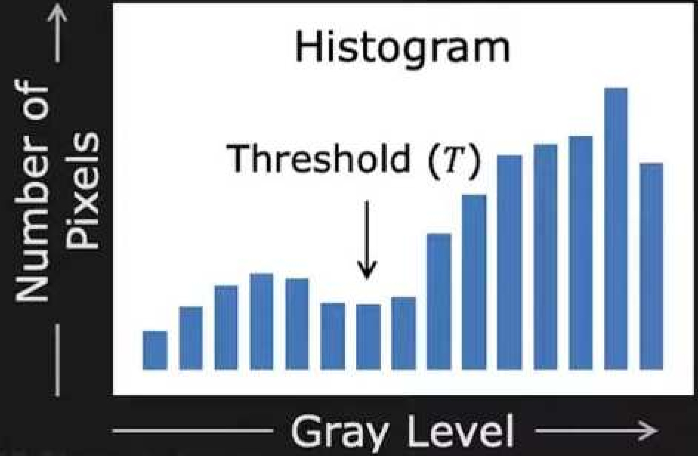
Characteristic Function:

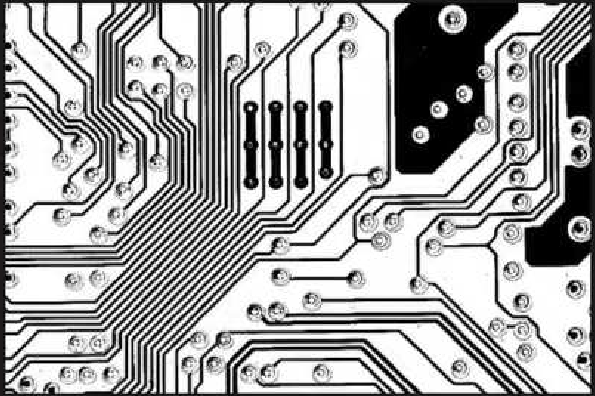
$$b(x,y) = \begin{cases} 0, & g(x,y) < T \\ 1, & g(x,y) \geq T \end{cases}$$

Selecting a Threshold (T)



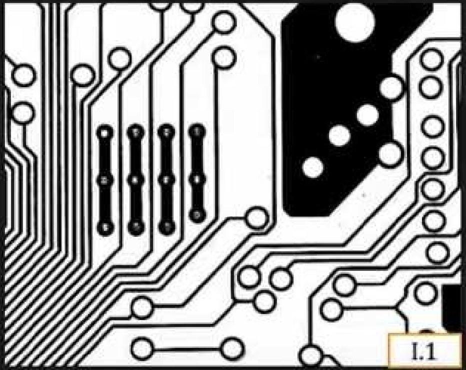
Gray Image $g(x, y)$





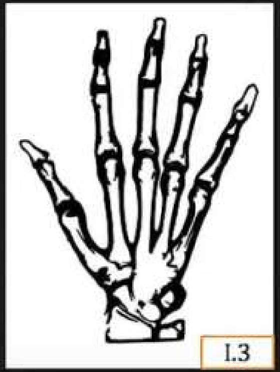
Binary Image $b(x, y)$

Examples of Binary Images



I.1

I.2



I.3

Capturing a Binary Image

Capturing a Binary Image

Backlighting

Binary Images

Binary Image: Can have only two values (0 or 1).
Simple to process and analyze.

Topics:

(1) Geometric Properties

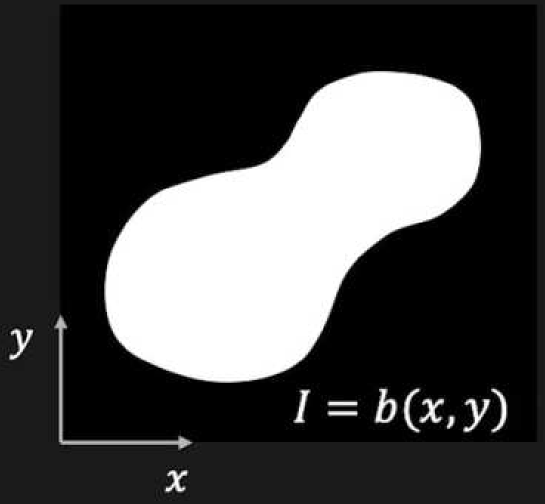
(2) Segmenting Binary Images

(3) Iterative Modification

Geometric Properties of Binary Images

Assume:

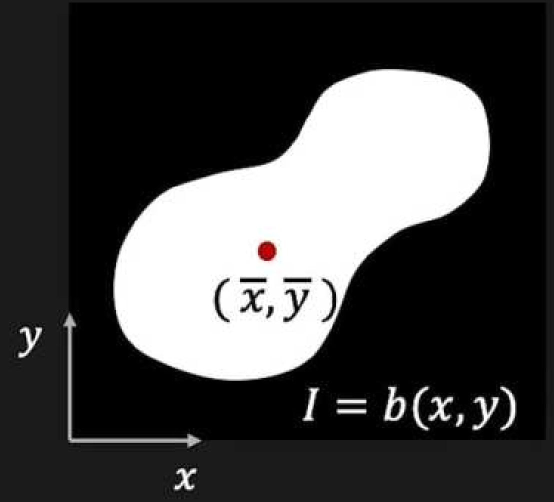
- $b(x, y)$ is continuous
- Only one object



Area and Position

Area: (Zeroth Moment)

$$A = \iint_I b(x,y)\,dx\,dy$$

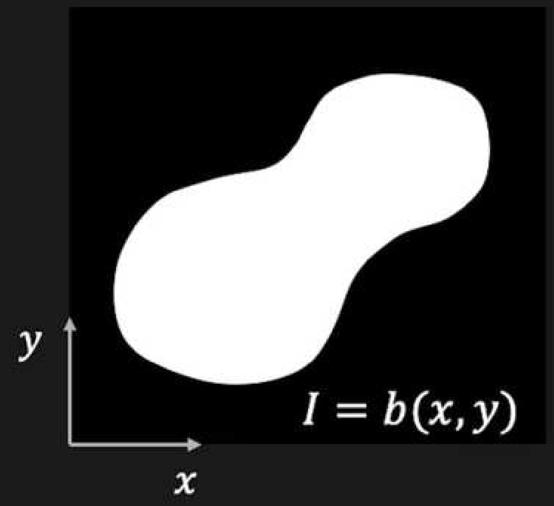


Position: Center of Area (First Moment)

$$\bar{x} = \frac{1}{A}\iint_I x\,b(x,y)\,dx\,dy \quad , \quad \bar{y} = \frac{1}{A}\iint_I y\,b(x,y)\,dx\,dy$$

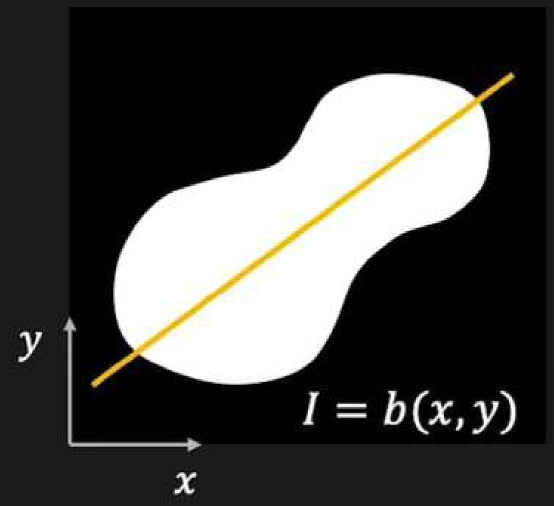
Orientation

Difficult to define!



Orientation

Difficult to define!

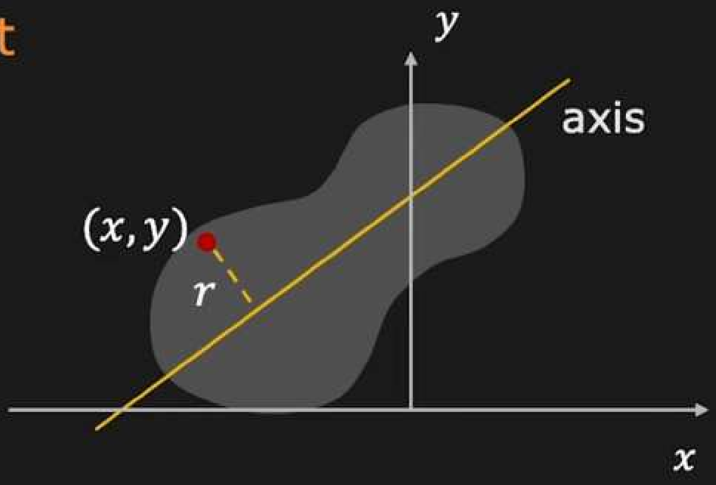


Use: Axis of Least Second Moment

Orientation

Axis of Least Second Moment minimizes:

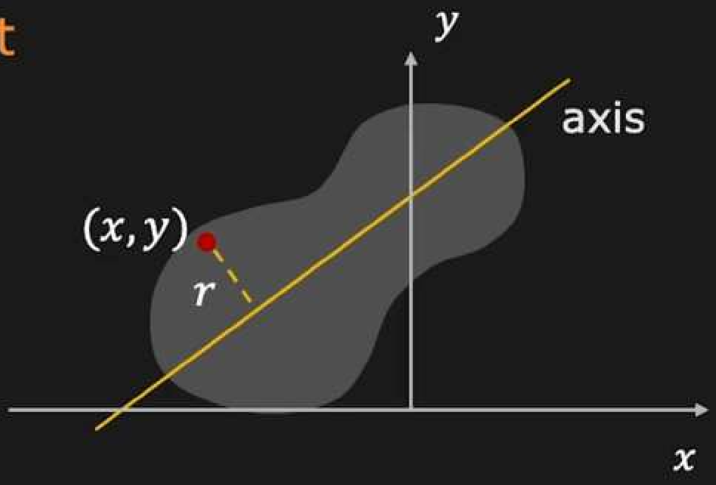
$$E = \iint_I r^2 \, b(x,y) \, dx \, dy$$



Orientation

Axis of Least Second Moment minimizes:

$$E = \iint_I r^2\, b(x,y)\, dx\, dy$$



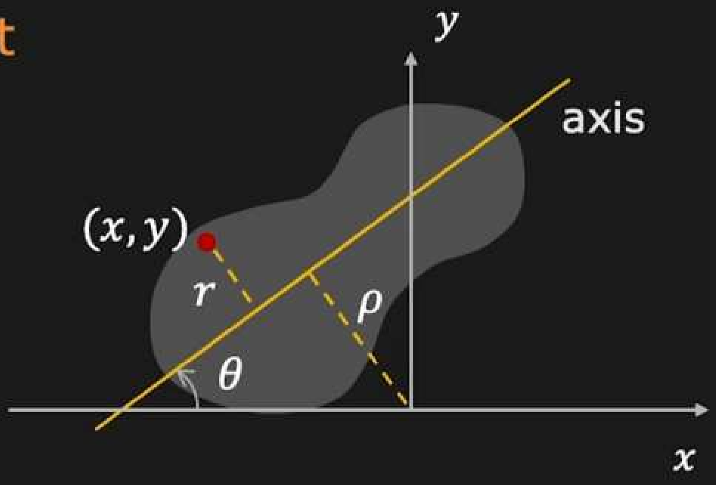
Which equation to use for axis?

$$y = mx + b \;? \qquad -\infty \le m \le \infty$$

Orientation

Axis of Least Second Moment minimizes:

$$E = \iint_I r^2\, b(x,y)\, dx\, dy$$



Which equation to use for axis?

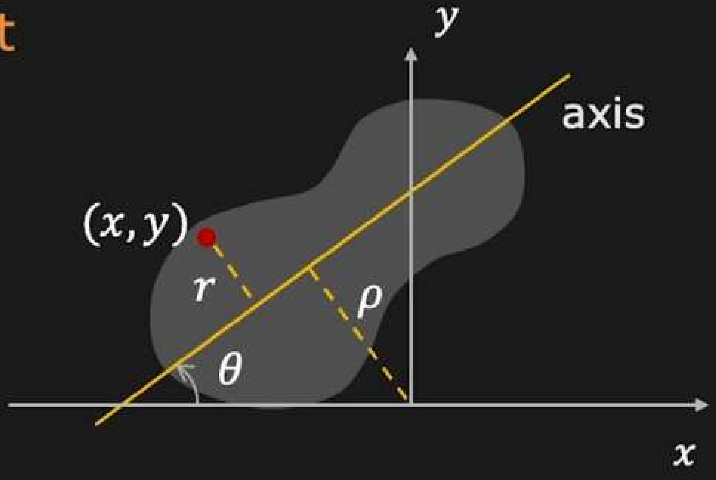
$$y = mx + b\ ? \qquad -\infty \le m \le \infty$$

Use: $\boxed{x \sin\theta - y\cos\theta + \rho = 0}$ ρ, θ are finite

Orientation

Axis of Least Second Moment minimizes:

$$E = \iint_I r^2\, b(x, y)\, dx\, dy$$



Which equation to use for axis?

$$y = mx + b \text{ ?} \qquad -\infty \leq m \leq \infty$$

Use: $\boxed{x \sin\theta - y\cos\theta + \rho = 0}$ ρ, θ are finite

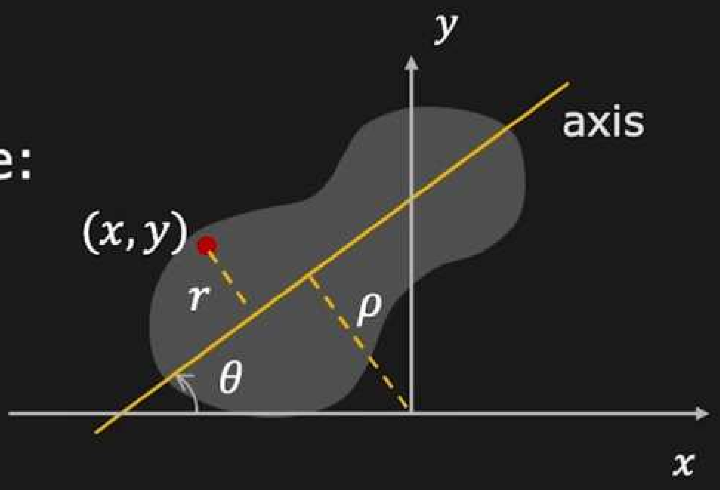
Find ρ and θ that minimize E for given $b(x, y)$

Distance Between Point and Line

Given a line $ax + by + c = 0$

Distance of point (x, y) from line:

$$r = \left|\frac{ax + by + c}{\sqrt{a^2 + b^2}}\right|$$



Similarly, given axis $x \sin\theta - y\cos\theta + \rho = 0$

Distance of point (x, y) from axis:

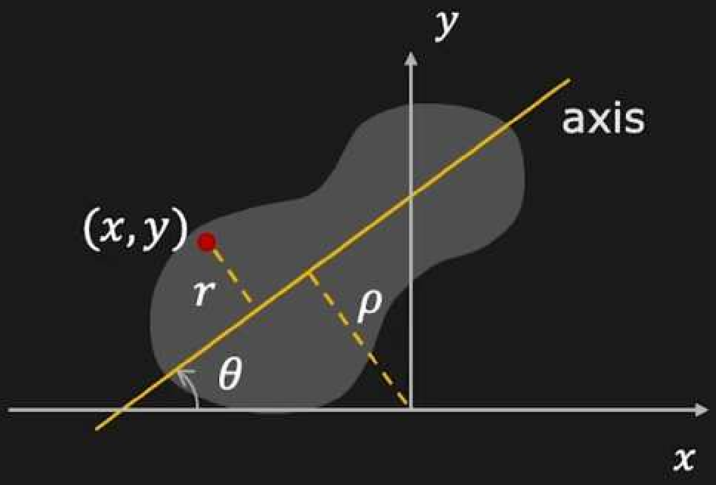
$$r = \left|\frac{x \sin\theta - y\cos\theta + \rho}{\sqrt{\sin^2\theta + \cos^2\theta}}\right|$$

$$\boxed{r = |x \sin\theta - y\cos\theta + \rho|}$$

Minimizing Second Moment

Axis of Least Second Moment minimizes:

$$E = \iint_I r^2 \, b(x,y) \, dx \, dy$$



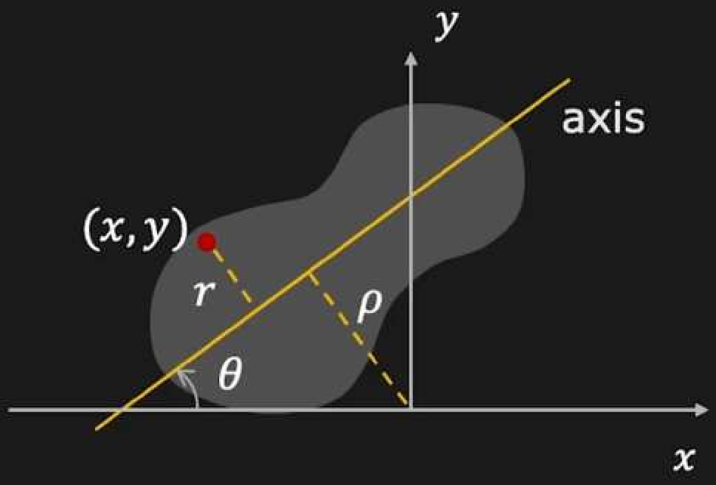
So, minimize:

$$E = \iint_I (x \sin\theta - y \cos\theta + \rho)^2 b(x,y) \, dx \, dy$$

Minimizing Second Moment

Axis of Least Second Moment minimizes:

$$E = \iint_I r^2\, b(x,y)\, dx\, dy$$



So, minimize:

$$E = \iint_I (x\sin\theta - y\cos\theta + \rho)^2 b(x,y)\, dx\, dy$$

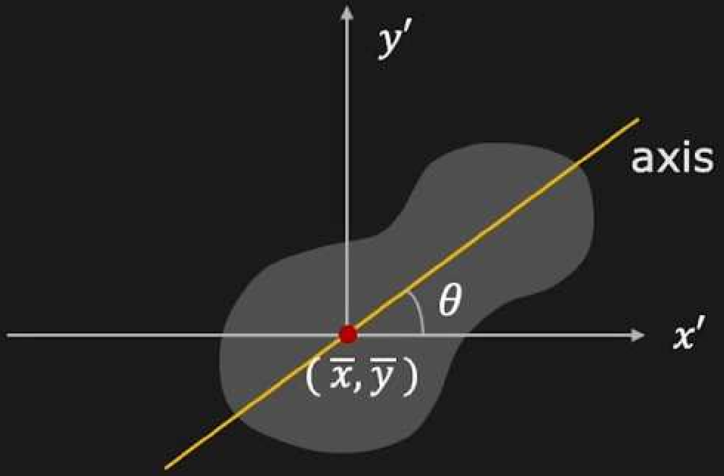
Using $\dfrac{\partial E}{\partial \rho} = 0$ we get: $A(\overline{x}\sin\theta - \overline{y}\cos\theta + \rho) = 0$

Shift the Coordinate System

Change coordinates:

$$x' = x - \overline{x},\ y' = y - \overline{y}$$

$$x \sin\theta - y\cos\theta + \rho$$
$$= x' \sin\theta - y' \cos\theta$$



Therefore, we can rewrite E as:

$$\boxed{E = a\sin^2\theta - b\sin\theta\cos\theta + c\cos^2\theta}$$

where: $\begin{cases} a = \iint\limits_{I'} (x')^2\, b(x,y)\, dx'\, dy' \\ c = \iint\limits_{I'} (y')^2\, b(x,y)\, dx'\, dy' \end{cases}$

$$b = 2\iint\limits_{I'} (x'y')\, b(x,y)\, dx'\, dy'$$

(a, b, c are easy to compute)

Finally, Minimize E

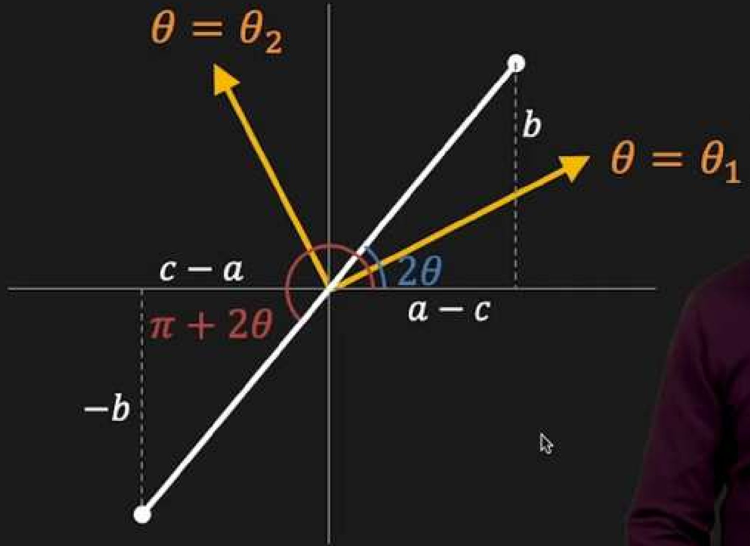
Using $\frac{dE}{d\theta} = (a-c)\sin 2\theta - b\cos 2\theta = 0$ we get: $\boxed{\tan 2\theta = \frac{b}{a-c}}$

We know that: $\tan 2\theta = \tan(2\theta + \pi) = \frac{-b}{c-a}$

θ has two solutions.

1. $\theta = \theta_1$

2. $\theta = \theta_2 = \theta_1 + \frac{\pi}{2}$



One gives Minimum of E
and the other Maximum of E

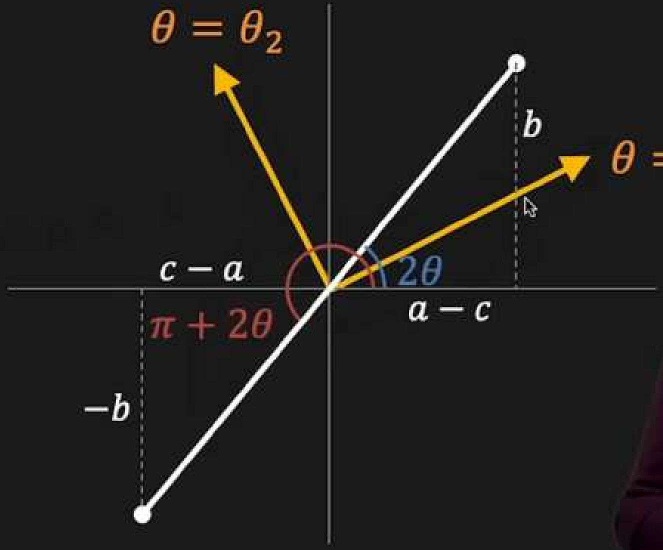
Finally, Minimize E

Using $\frac{dE}{d\theta} = (a-c)\sin 2\theta - b\cos 2\theta = 0$ we get: $\boxed{\tan 2\theta = \frac{b}{a-c}}$

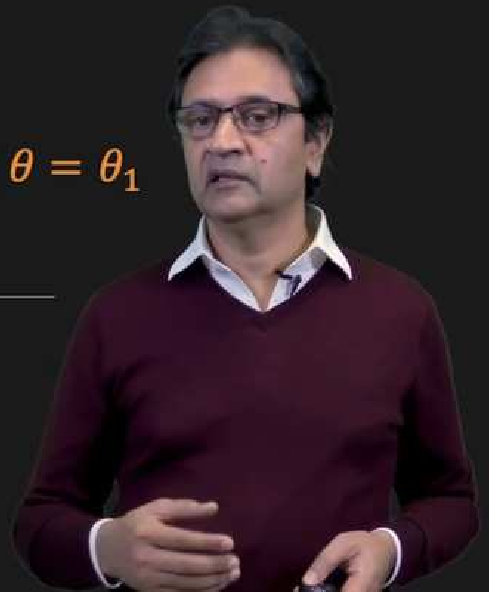
We know that: $\tan 2\theta = \tan(2\theta + \pi) = \frac{-b}{c-a}$

θ has two solutions.

1. $\theta = \theta_1$
2. $\theta = \theta_2 = \theta_1 + \frac{\pi}{2}$



One gives Minimum of E
and the other Maximum of E



Which One To Use?

Using second derivative test:

If $\dfrac{d^2E}{d\theta^2} = (a-c)\cos 2\theta + b\sin 2\theta \begin{cases} > 0 \text{ then Minimum} \\ < 0 \text{ then Maximum} \end{cases}$

Substituting $\cos 2\theta_1$, $\sin 2\theta_1$, $\cos 2\theta_2$ and $\sin 2\theta_2$:

$$\frac{d^2E}{d\theta^2}(\theta_1) > 0 \quad \text{and} \quad \frac{d^2E}{d\theta^2}(\theta_2) < 0$$

Therefore,

Orientation: $\boxed{\theta = \theta_1 = \dfrac{atan2(b, a-c)}{2}}$

Which One To Use?

Using second derivative test:

If $\dfrac{d^2E}{d\theta^2} = (a-c)\cos 2\theta + b\sin 2\theta \begin{cases} > 0 \text{ then Minimum} \\ < 0 \text{ then Maximum} \end{cases}$

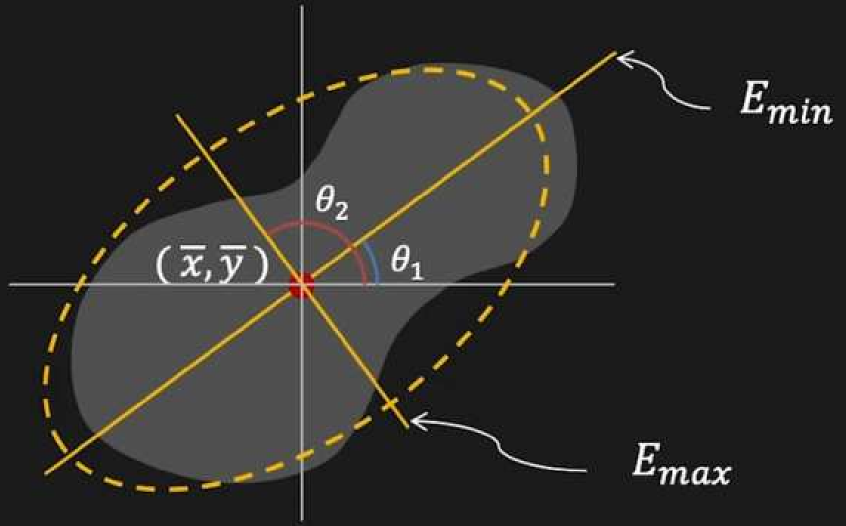
Substituting $\cos 2\theta_1$, $\sin 2\theta_1$, $\cos 2\theta_2$ and $\sin 2\theta_2$:

$$\frac{d^2E}{d\theta^2}(\theta_1) > 0 \quad \text{and} \quad \frac{d^2E}{d\theta^2}(\theta_2) < 0$$

Therefore,

Orientation: $\boxed{\theta = \theta_1 = \dfrac{atan2(b, a-c)}{2}}$

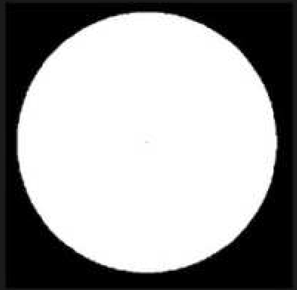
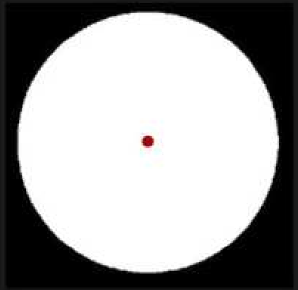
Roundedness



$$\boxed{\text{Roundedness} = \frac{E_{min}}{E_{max}}}$$

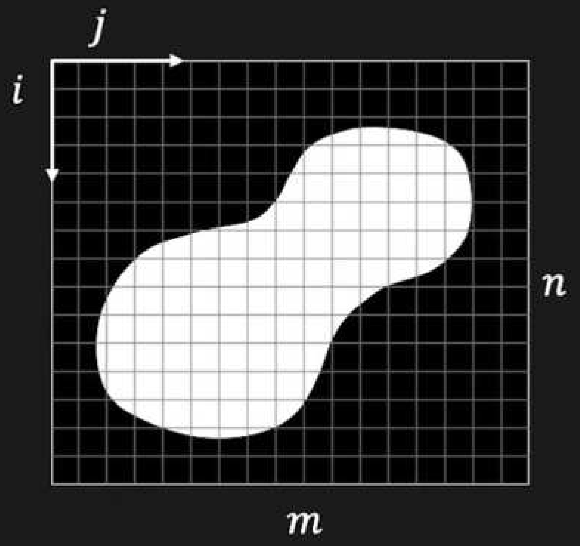
where: $E_{min} = E(\theta_1)$ and $E_{max} = E(\theta_2)$

Examples

Gray Image	Binary Image	Orientation	Roundedness
			0.19
			0.49
			1.0

Discrete Binary Images

b_{ij}: Value at cell (pixel) in row i and column j.
Assume pixel area = 1.



Area: $$A = \sum_{i=1}^{n} \sum_{j=1}^{m} b_{ij}$$

Position: Center of Area (First Moment)

$$\bar{x} = \frac{1}{A} \sum_{i=1}^{n} \sum_{j=1}^{m} i b_{ij} \qquad \bar{y} = \frac{1}{A} \sum_{i=1}^{n} \sum_{j=1}^{m} j b_{ij}$$

Discrete Binary Images

Second Moments:

$$a' = \sum_{i=1}^{n} \sum_{j=1}^{m} i^2 b_{ij} \qquad b' = 2 \sum_{i=1}^{n} \sum_{j=1}^{m} i j b_{ij} \qquad c' = \sum_{i=1}^{n} \sum_{j=1}^{m} j^2 b_{ij}$$

Note: a', b', c' are second moments w.r.t origin.
a, b, c (w.r.t. center) can be found from a', b', c', $\bar{x}$, $\bar{y}$, A

Hint: Expand $a = \sum_{i=1}^{n} \sum_{j=1}^{m} (i - \bar{x})^2 b_{ij}$ and represent in terms of a', $\bar{x}$, A.

Discrete Binary Images

Second Moments:

$$a' = \sum_{i=1}^{n}\sum_{j=1}^{m} i^2 b_{ij} \qquad b' = 2\sum_{i=1}^{n}\sum_{j=1}^{m} ij\, b_{ij} \qquad c' = \sum_{i=1}^{n}\sum_{j=1}^{m} j^2 b_{ij}$$

Note: a', b', c' are second moments w.r.t origin.
a, b, c (w.r.t. center) can be found from a', b', c', $\bar{x}$, $\bar{y}$, A

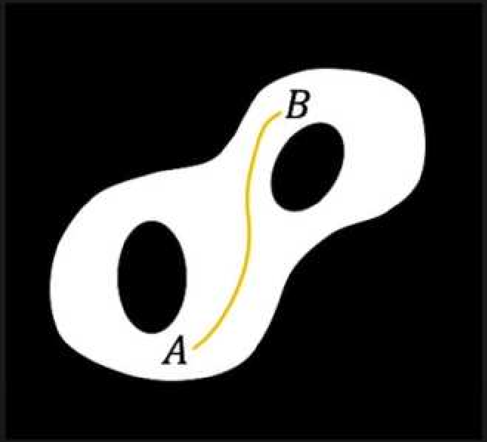
Hint: Expand $a = \sum_{i=1}^{n}\sum_{j=1}^{m}(i-\bar{x})^2 b_{ij}$ and represent in terms of a', $\bar{x}$, A.

Multiple Objects

Need to Segment image into separate Components

Connected Component

Maximal Set of Connected Points



A and B are connected if path exists between A and B along which $b(x,y)$ is constant.

Connected Component Labeling

Region Growing Algorithm

(a) Find Unlabeled "Seed" point with $b = 1$.
If not found, Terminate.

(b) Assign New Label to seed point

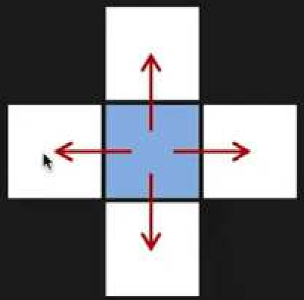
(c) Assign Same Label to its Neighbors with $b = 1$

(d) Assign Same Label to Neighbors of Neighbors with $b = 1$. Repeat until no more Unlabeled Neighbors with b=1.

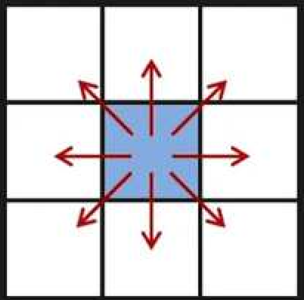
(e) Go to (a)

What do we mean by Neighbors?

Connectedness



4-Connectedness
4-C



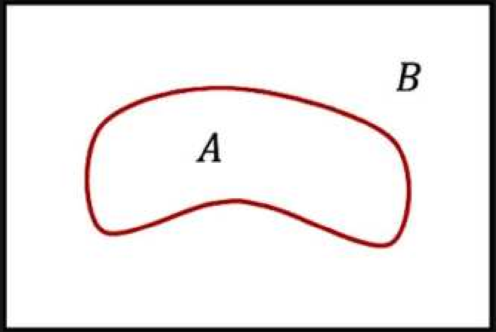
8-Connectedness
8-C

Neither is Perfect!

Connectedness

Jordan's Curve Theorem

Closed curve
→ 2 Connected Regions



Consider

0	1	0
1	0	1
0	1	0

→

B1	O1	B1
O4	B2	O2
B1	O3	B1

4-C
Hole without a closed loop!

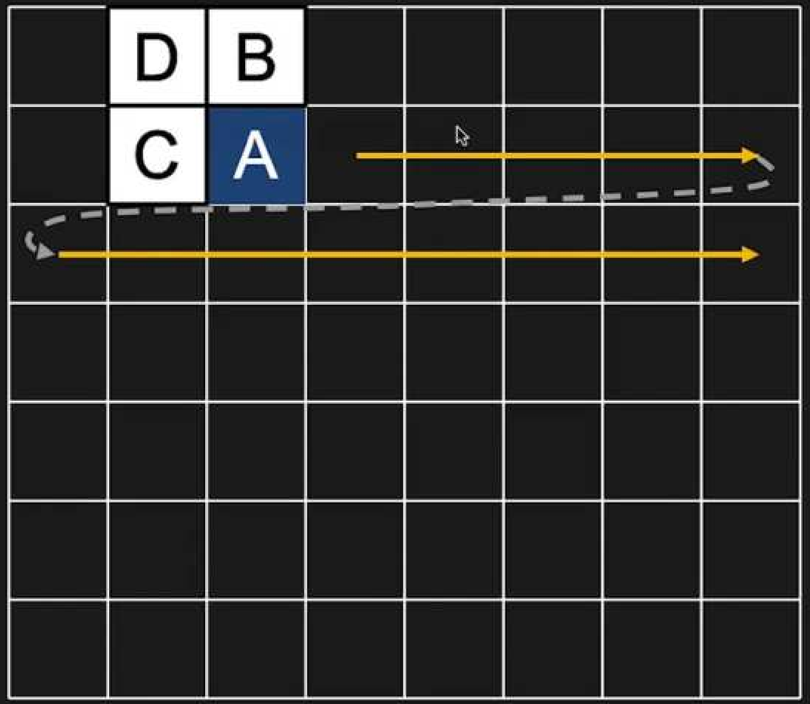
B	O	B
O	B	O
B	O	B

8-C
Connected backgrounds with a closed loop!

Hexagonal Tessellation

Above asymmetry makes a Square Grid behave like a Hexagonal Grid

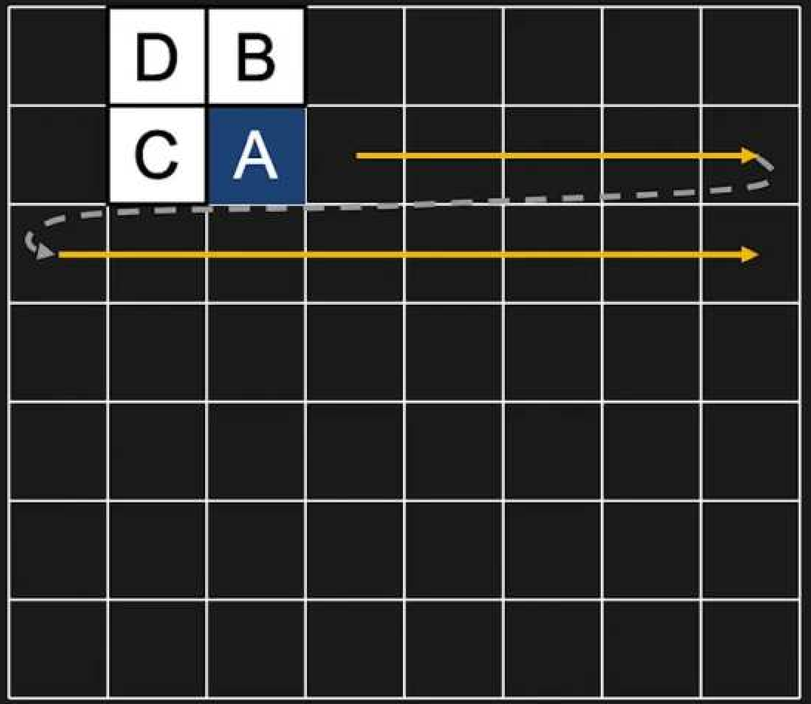
Sequential Labeling Algorithm



Raster Scanning

We want to label A.
B, C, D are already labeled.

Sequential Labeling Algorithm



Raster Scanning

We want to label A.
B, C, D are already labeled.

Sequential Labeling Algorithm

X	X
X	0

→ label(A) = "background"

0	0
0	1

→ label(A) = new label

D	X
X	1

→ label(A) = label(D)

0	0
C	1

→ label(A) = label(C)

0	B
0	1

→ label(A) = label(B)

0	B
C	1

→ If

label(B) = label(

then,

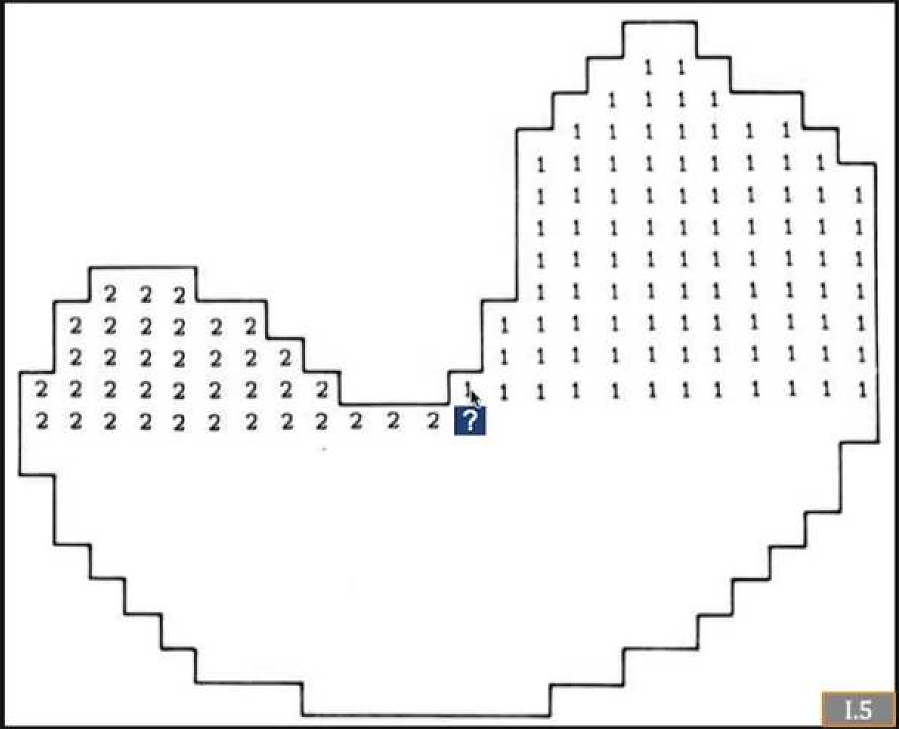
label(A) = label

X: Value does not matter (Can be 0 or 1)

Sequential Labeling Algorithm

0	B
C	1

→ What if label(B) not equal to label(C)?



Sequential Labeling Algorithm

0	B
C	1

→ What if label(B) not equal to label(C)?

Solution: Create Equivalence Table

- Note down that label(B) ≡ label(C)
- Assign label(A) = label(B)

$2 \equiv 1$
$7 \equiv 3, 6, 4$
$8 \equiv 5$
⋮

Euler Number (E)

No. of Bodies (B) – No. of Holes (H)

Letter B: $E_b = -1$

Letter i: $E_i = 2$

Letter n: $E_n = 1$

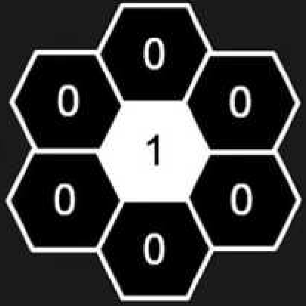
Image : $E = 4$

$$E_{image} = \sum E_{non-overlapping\ regions}$$

Euler Differential (E^*)

Change in the Euler number of the image.

$E = 0$



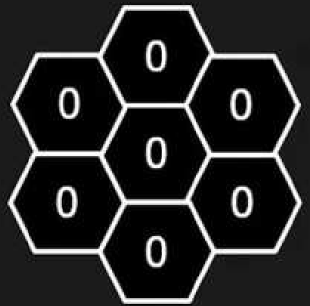
$E = 1$

Euler Differential: $E^* = 1$

Neighborhood Sets Based on E^*

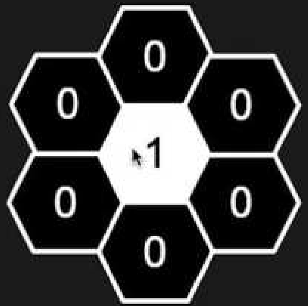
Each pixel has $2^6 = 64$ possible neighborhoods

Neighborhood patterns are classified based on the Euler Differential they generate, assuming the center pixel goes from 0 to 1.

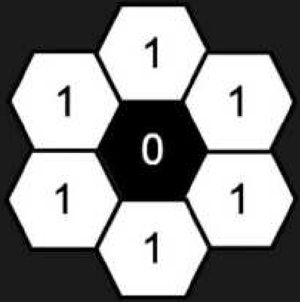


Neighborhood $\in N_{+1}$

$0 \rightarrow 1, E^* = 1$



Neighborhood Sets Based on E^*



Neighborhood $\in N_{+1}$

$0 \rightarrow 1, E^* = 1$

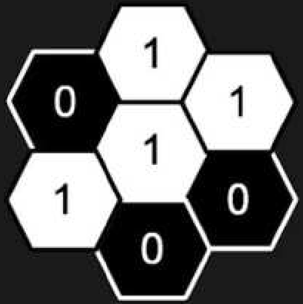
Neighborhood $\in N_0$

$0 \rightarrow 1, E^* = 0$

Neighborhood Sets Based on E^*

Neighborhood $\in N_{-1}$

$0 \rightarrow 1, E^* = -1$

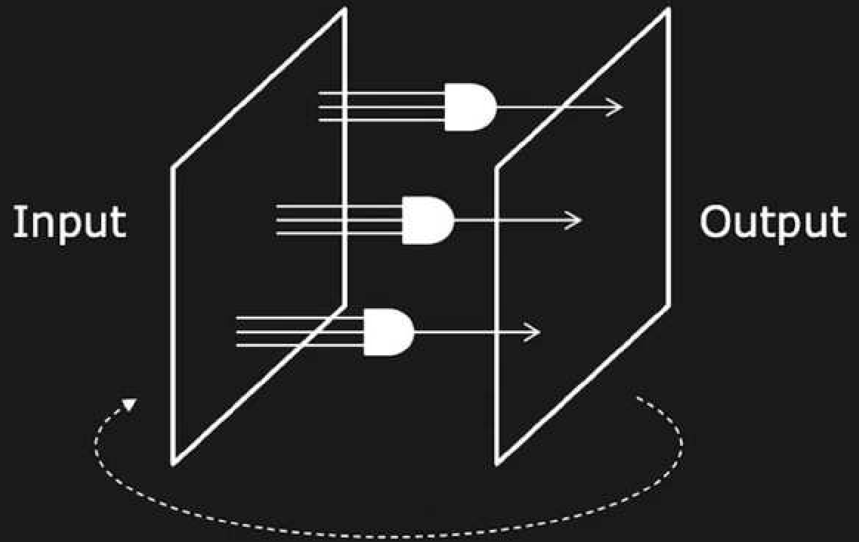


Neighborhood $\in N_{-2}$

$0 \rightarrow 1, E^* = -2$

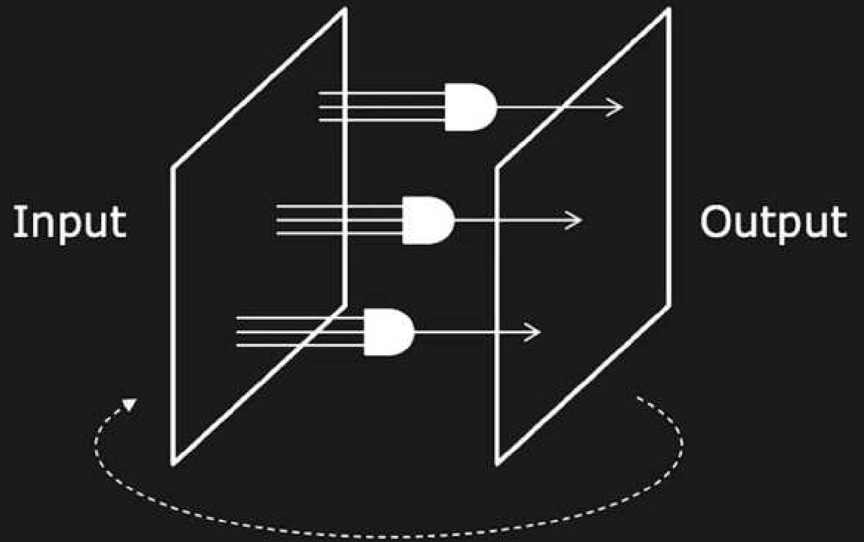
Iterative Neighborhood Operations

Incrementally apply neighborhood operations on images



Iterative Neighborhood Operations

Incrementally apply neighborhood operations on images



Conservative Operations do not change the Euler number of the image.

Notation for Iterative Modification

Specify Neighborhood Set, S. Ex: S can be N_{+1}, N_0, N_{-1} or N_{-2} or a combination of these.

Consider pixel (i, j). Let:

- $a_{ij} = 1$ if Neighborhood of $(i, j) \in S$ else 0
- b_{ij} = current value of pixel (i, j)
- c_{ij} = new value of pixel (i, j)

a_{ij}	b_{ij}
0	0
0	1
1	0
1	1

c_{ij}
?
?
?
?

Iterative Modification Algorithms

Specify Neighborhood Set S and apply one of the 16 algorithms to each pixel.

16 algorithms

a_{ij}	b_{ij}	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Iterative Modification Algorithms

Specify Neighborhood Set S and apply one of the 16 algorithms to each pixel.

16 algorithms

a_{ij}	b_{ij}	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Growing Objects: $S \in N_0$ and $Algorithm\ 7$

Thinning Objects: $S \in N_0$ and $Algorithm\ 4$

Finding Skeletons

Thinning without changing the Euler number

$S \in N_0$ and $Algorithm\ 4$

Finding Skeletons

Thinning without changing the Euler number

$S \in N_0$ and $Algorithm\ 4$

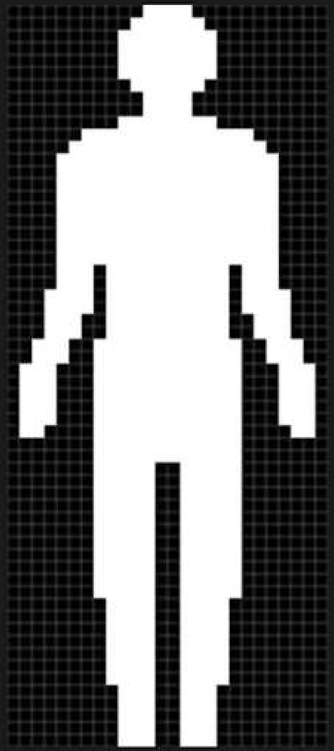


Image Processing I

Transform image to new one that is clearer or easier to analyze.

Image Processing I

Transform image to new one that is clearer or easier to analyze.

Topics:

(1) Pixel Processing

(2) LSIS and Convolution

(3) Linear Image Filters

Image Processing I

Transform image to new one that is clearer or easier to analyze.

Topics:

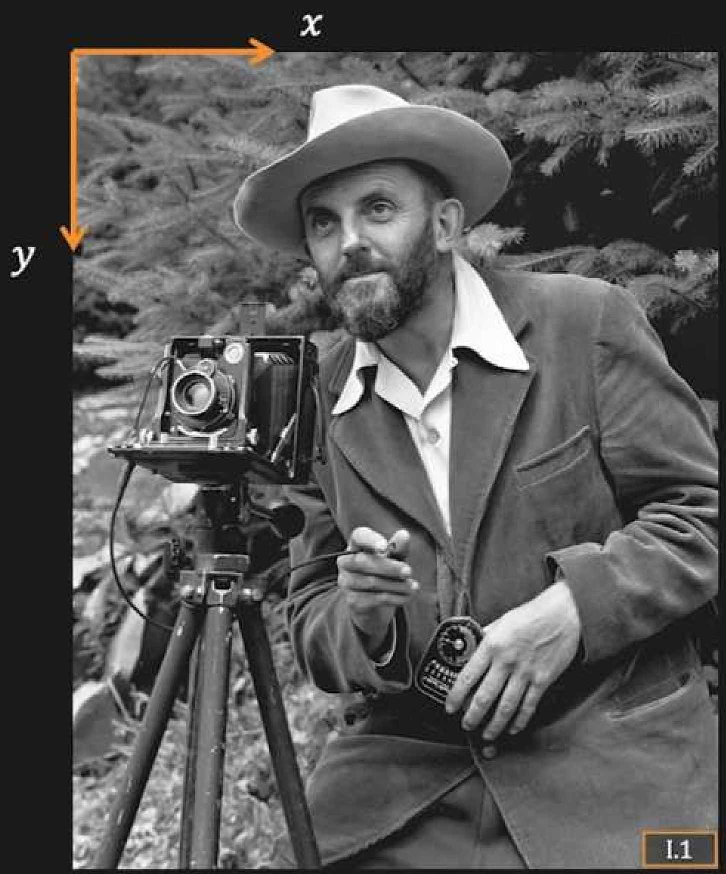
(1) Pixel Processing

(2) LSIS and Convolution

(3) Linear Image Filters

(4) Non-Linear Image Filters

Image as a Function



$f(x, y)$ is the image intensity at position (x, y)

Pixel (Point) Processing

Transformation T of intensity f at each pixel to intensity g:

$$g(x, y) = T\left(f(x, y)\right)$$

Point Processing

Darken $(f - 128)$

Original (f)

Lighten $(f + 128)$

Invert $(255 - f)$

Pixel Processing

Original (f)

Low Contrast $(f/2)$

High Contrast $(f * 2)$

Gray $(0.3f_R + 0.6f_G + 0.1$

Linear Shift Invariant System (LSIS)

$$f(x) \longrightarrow \boxed{\text{LSIS}} \longrightarrow g(x)$$

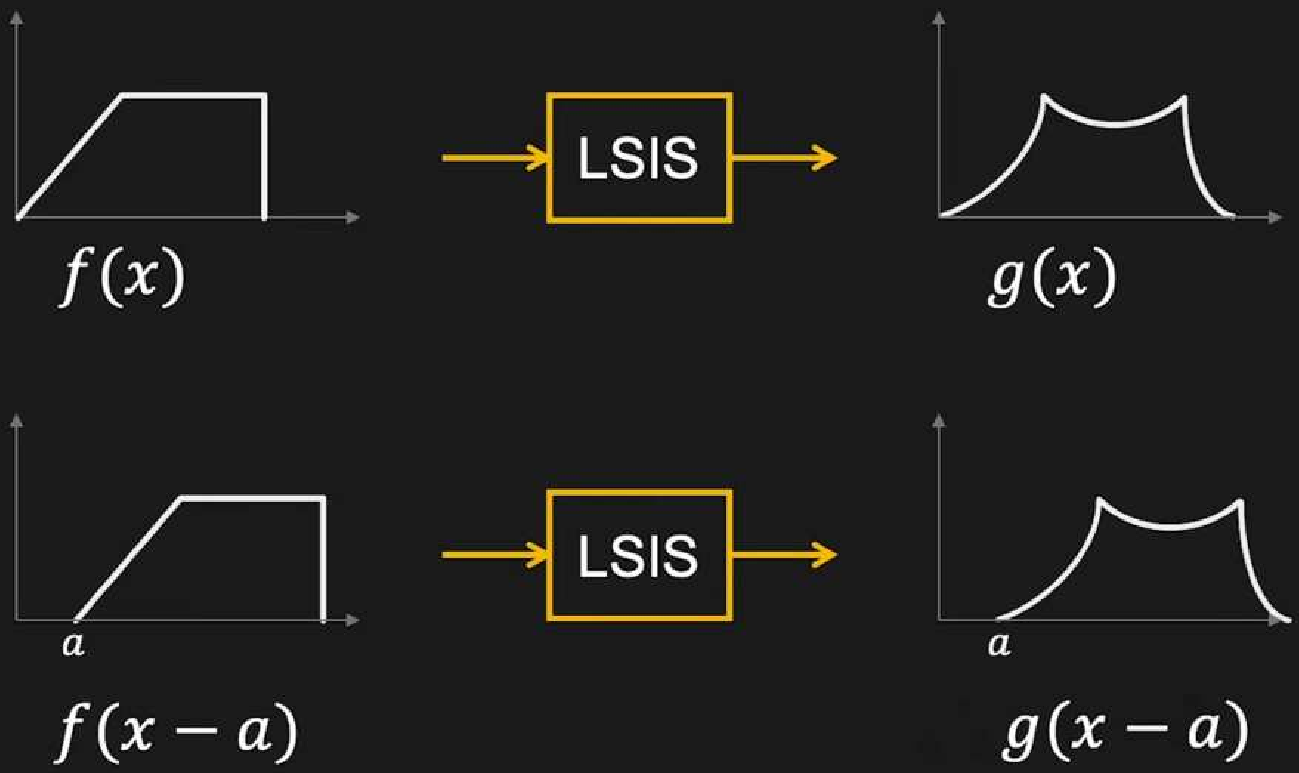
Study of Linear Shift Invariant Systems (LSIS) leads to useful image processing algorithms.

LSIS: Linearity

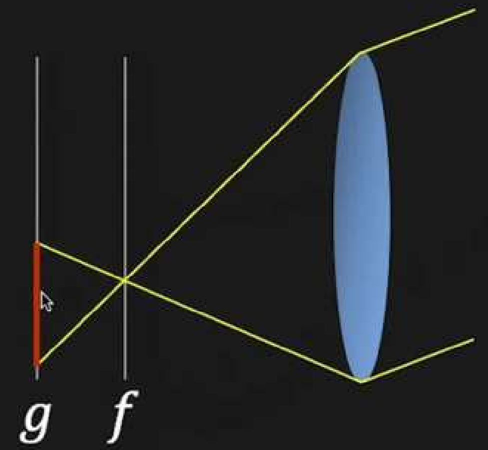
$$f_1 \longrightarrow \boxed{\text{LSIS}} \longrightarrow g_1 \qquad\qquad f_2 \longrightarrow \boxed{\text{LSIS}} \longrightarrow g_2$$

$$\alpha f_1 + \beta f_2 \longrightarrow \boxed{\text{LSIS}} \longrightarrow \alpha g_1 + \beta g_2$$

LSIS: Shift Invariance

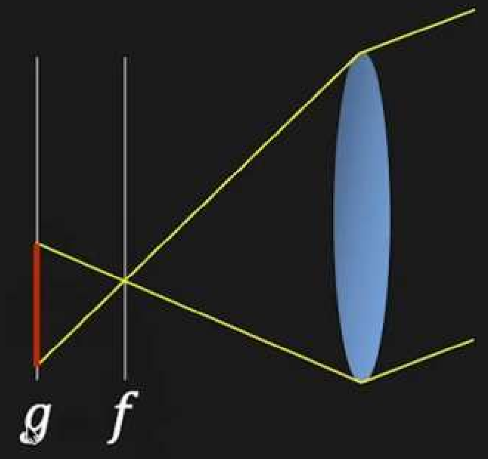


Ideal Lens is an LSIS



Defocused Image (g): Processed version of Focused Image (f)

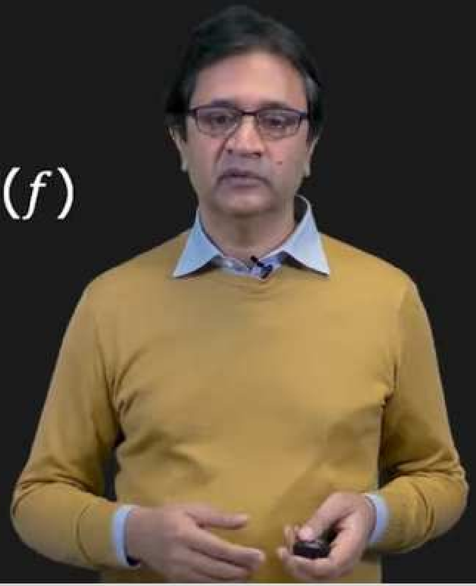
Ideal Lens is an LSIS



Defocused Image (g): Processed version of Focused Image (f)

Linearity: Brightness variation

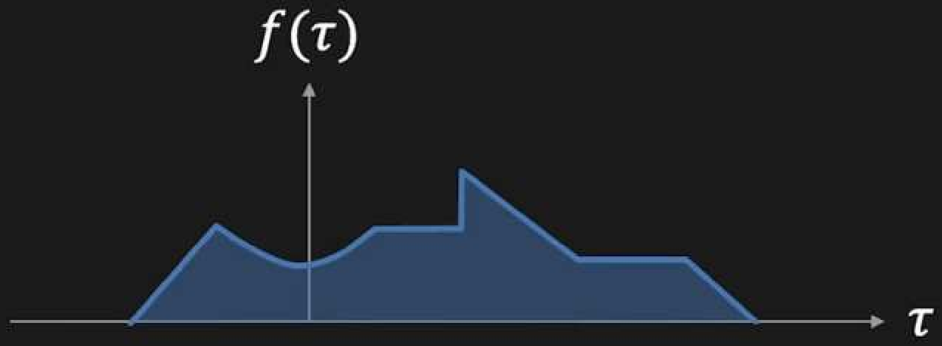
Shift invariance: Scene movement

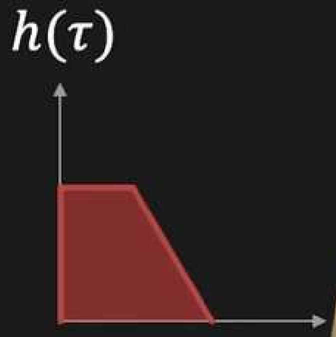


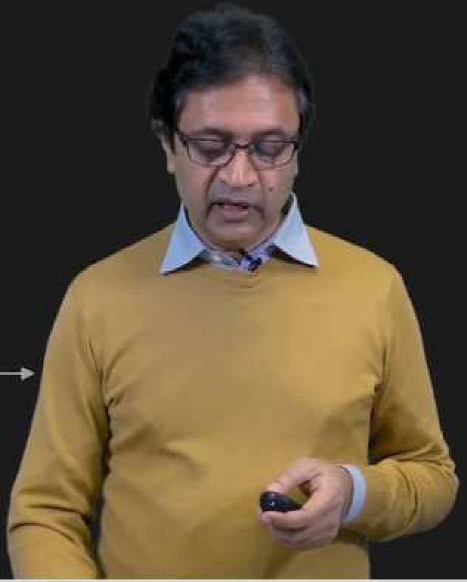
Convolution

Convolution of two functions $f(x)$ and $h(x)$

$$\boxed{g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau}$$



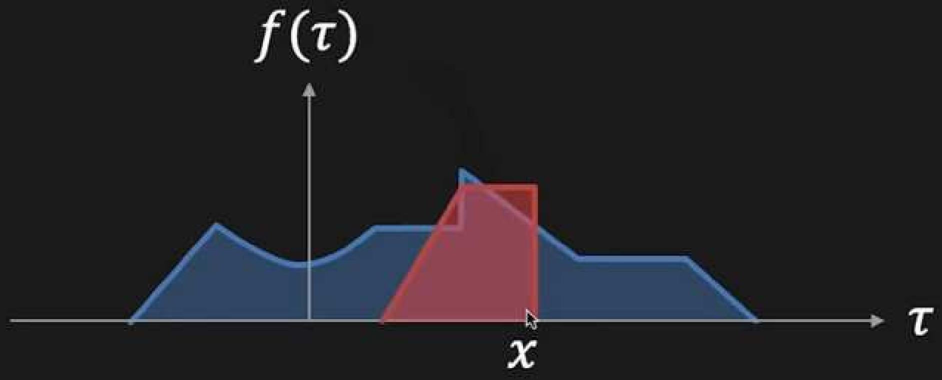


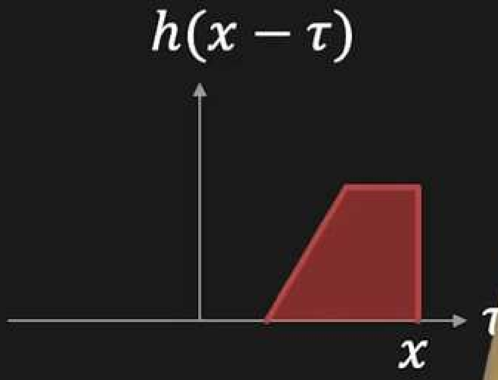


Convolution

Convolution of two functions $f(x)$ and $h(x)$

$$\boxed{g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau}$$

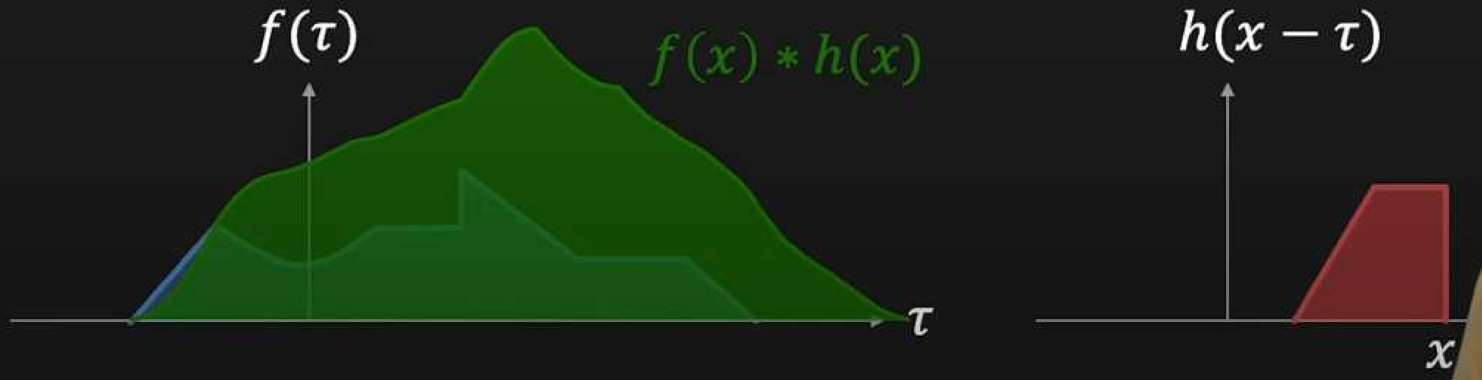




Convolution

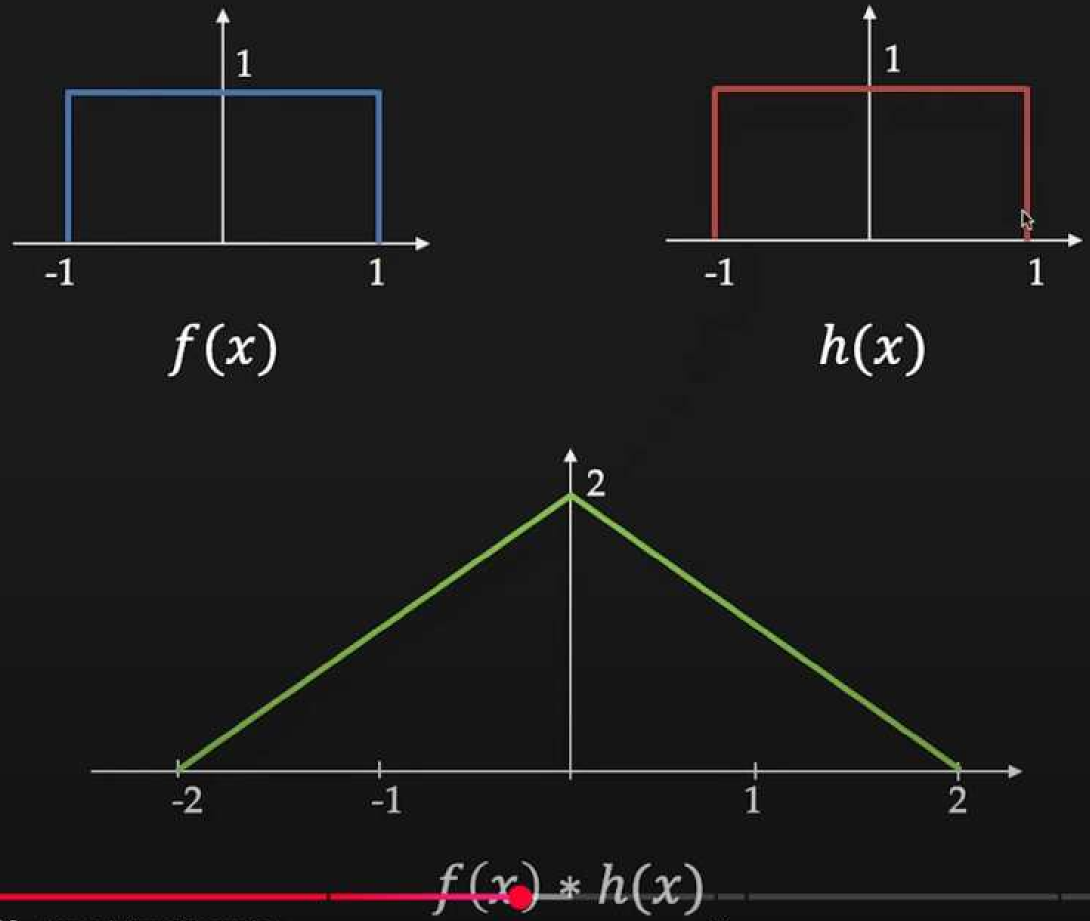
Convolution of two functions $f(x)$ and $h(x)$

$$\boxed{g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau}$$

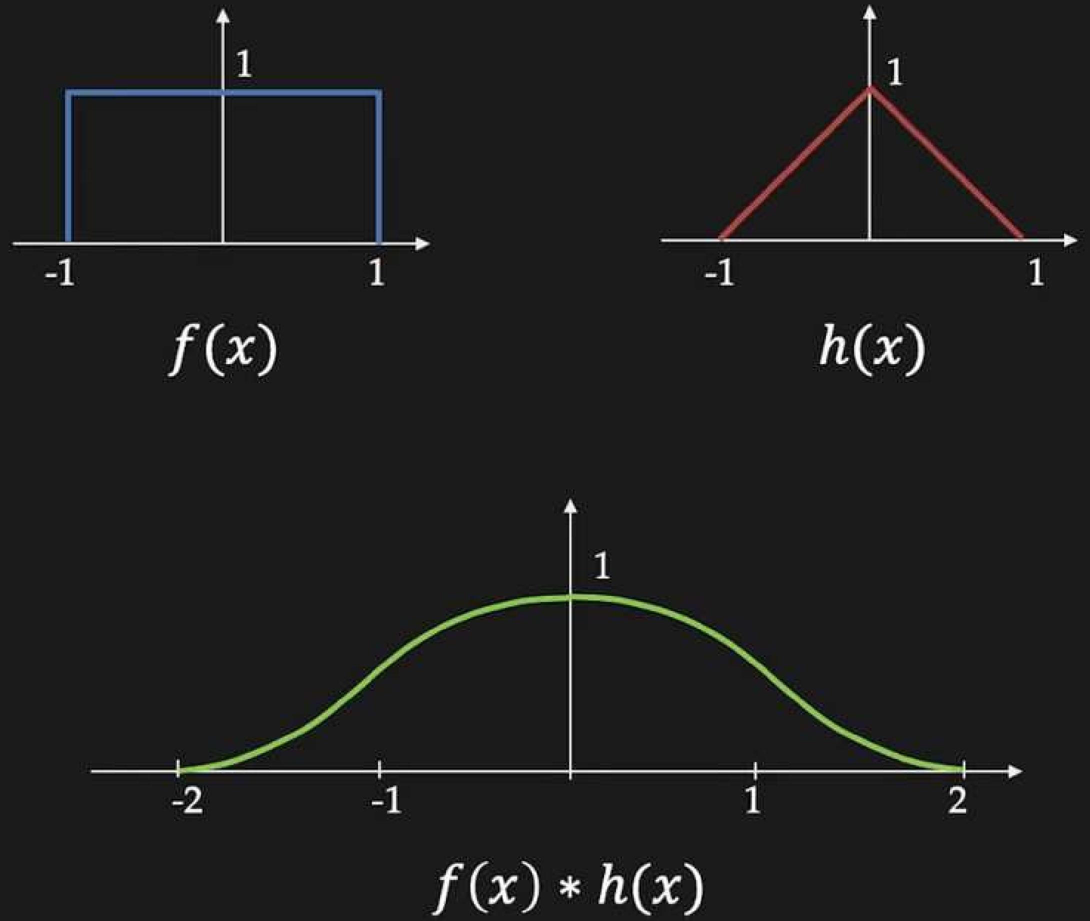


LSIS implies Convolution and Convolution implies LS

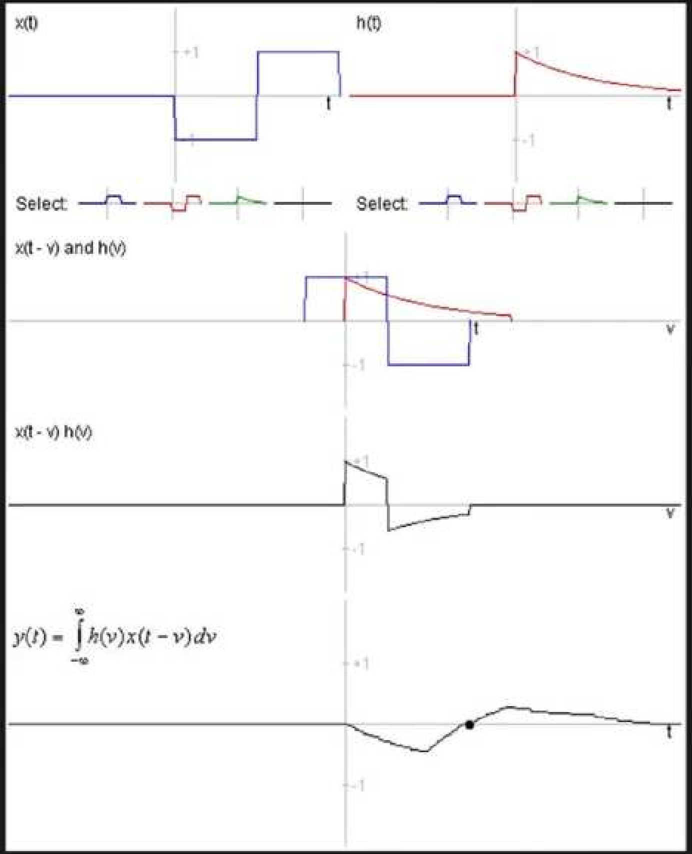
Convolution: Example



Convolution: Example



Convolution: Online Demo



http://www.jhu.edu/signals/convolve/

Convolution is LSIS

Linearity:

Let: $g_1(x) = \int_{-\infty}^{\infty} f_1(\tau)h(x-\tau)\,d\tau$ and $g_2(x) = \int_{-\infty}^{\infty} f_2(\tau)h(x-\tau)\,d\tau$

Then:

$$\int_{-\infty}^{\infty} (\alpha f_1(\tau) + \beta f_2(\tau))h(x-\tau)\,d\tau$$

$$= \alpha \int_{-\infty}^{\infty} f_1(\tau)h(x-\tau)\,d\tau + \beta \int_{-\infty}^{\infty} f_2(\tau)h(x-\tau)\,d\tau$$

$$= \alpha g_1(x) + \beta g_2(x)$$

Convolution is LSIS

Shift Invariance:

Let: $g(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau$

Then:

$$\int_{-\infty}^{\infty} f(\tau - a)h(x-\tau)\,d\tau$$

$$= \int_{-\infty}^{\infty} f(\mu)h(x-a-\mu)\,d\mu \qquad \text{(Substituting } \mu = \tau - a\text{)}$$

$$= g(x-a)$$

Can we find h?

$$f \longrightarrow \boxed{\;h\;} \longrightarrow g \qquad g(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau$$

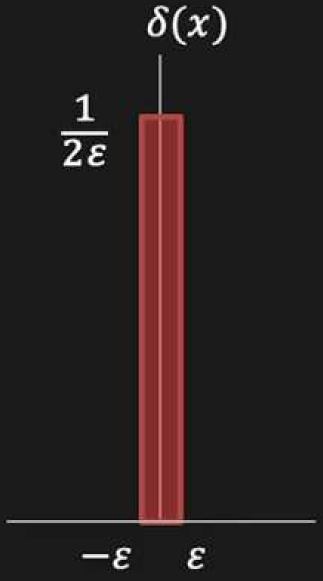
What input f will produce output $g = h$?

$$h(x) = \int_{-\infty}^{\infty} ?\,(\tau)h(x-\tau)\,d\tau$$

Unit Impulse Function

$$\delta(x) = \begin{cases} 1/2\varepsilon, & |x| \le \varepsilon \\ 0, & |x| > \varepsilon \end{cases}$$

$$\varepsilon \to 0$$



$$\int_{-\infty}^{\infty} \delta(\tau)\, d\tau = \frac{1}{2\varepsilon} . 2\varepsilon = 1$$

$$\boxed{\int_{-\infty}^{\infty} \delta(\tau) b(x-\tau)\, d\tau = b(x)}$$

Sifting Property

Impulse Response

$$f \longrightarrow \boxed{h} \longrightarrow g$$

$$\delta \longrightarrow \boxed{h} \longrightarrow h$$

Unit Impulse

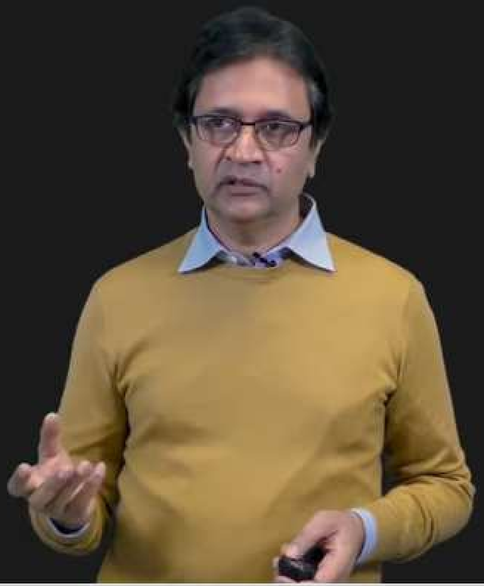
Impulse Response

$$g(x) = f(x) * h(x)$$

$$h(x) = \delta(x) * h(x)$$

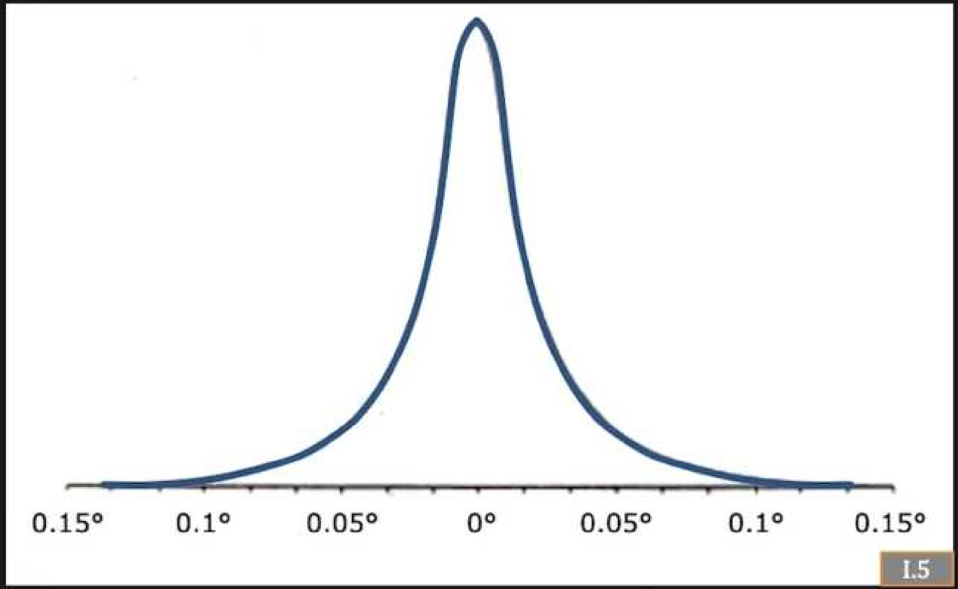
$$g(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau$$

$$h(x) = \int_{-\infty}^{\infty} \delta(\tau)h(x-\tau)\,d\tau$$



Impulse Response of Human Eye

Distant Star $\delta(x,y)$ → **Human Eye** → Point Spread Function (PSF) $h(x,y)$



Human Eye PSF

Properties of Convolution

Commutative $a * b = b * a$

Associative $(a * b) * c = a * (b * c)$

Cascaded System

$$f \longrightarrow \boxed{h_1} \longrightarrow \boxed{h_2} \longrightarrow g$$

$$\equiv \quad f \longrightarrow \boxed{h_1 * h_2} \longrightarrow g$$

2D Convolution

LSIS:

$$f(x,y) \longrightarrow \boxed{h(x,y)} \longrightarrow g(x,y)$$

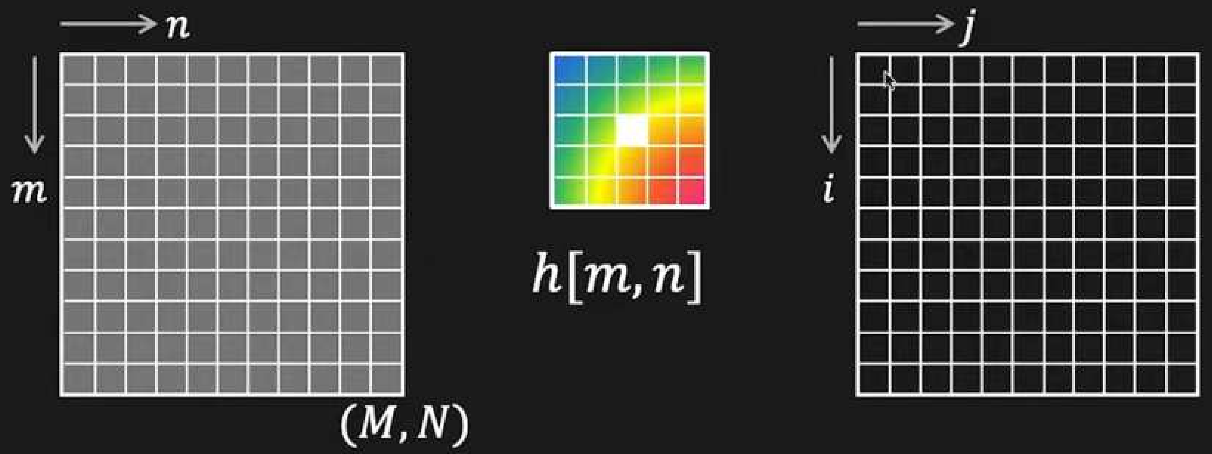
Convolution:

$$g(x,y) = \iint_{-\infty}^{\infty} f(\tau,\mu)h(x-\tau,y-\mu)\,d\tau d\mu$$

Convolution with Discrete Images

$$f[i,j] \longrightarrow \boxed{h[i,j]} \longrightarrow g[i,j]$$

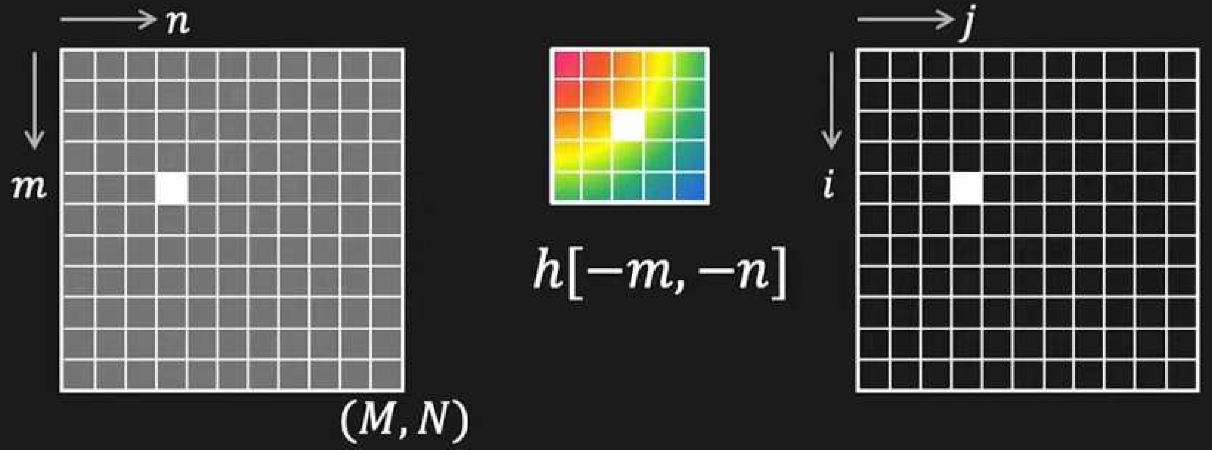
$$g[i,j] = \sum_{m=1}^{M} \sum_{n=1}^{N} f[m,n]h[i-m,j-n]$$



Convolution with Discrete Images

$$f[i,j] \longrightarrow \boxed{h[i,j]} \longrightarrow g[i,j]$$

$$g[i,j] = \sum_{m=1}^{M} \sum_{n=1}^{N} f[m,n]\underbrace{h[i-m,j-n]}_{\text{“Mask,” “Kernel,” “Filter”}}$$

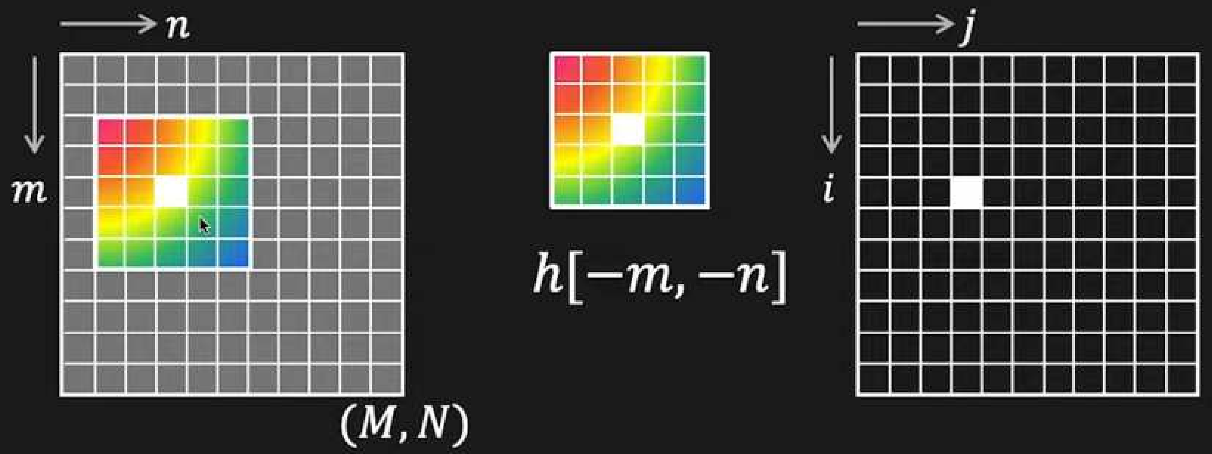


Convolution with Discrete Images

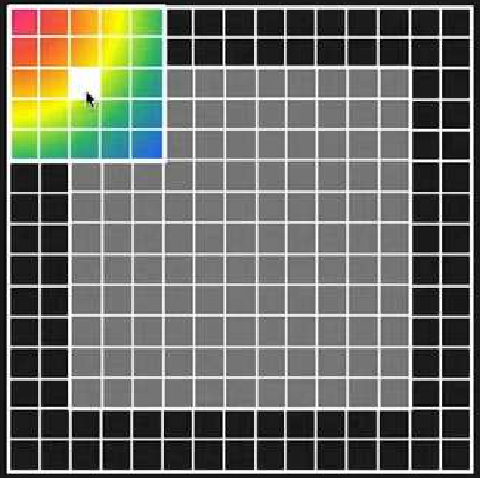
$$f[i,j] \longrightarrow \boxed{h[i,j]} \longrightarrow g[i,j]$$

$$g[i,j] = \sum_{m=1}^{M} \sum_{n=1}^{N} f[m,n]\underline{h[i-m,j-n]}$$

"Mask," "Kernel," "Filter"



Border Problem



Solution:

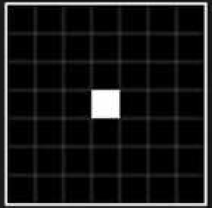
- Ignore border
- Pad with constant value
- Pad with reflection

Example: Impulse Filter

Input

$f(x, y)$

$*$



$\delta(x, y)$

$=$

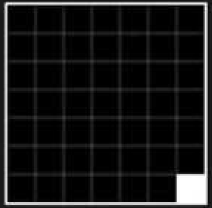
Output

$f(x, y)$

Example: Image Shift

Input

$*$



$=$

Output

$f(x, y)$

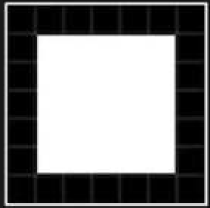
$\delta(x-u, y-v)$

$f(x-u, y-v)$

Example: Averaging

Input

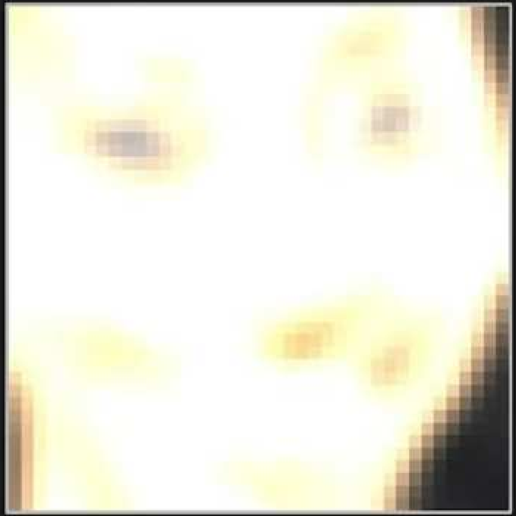
$*$



"Box Filter"
5 x 5

$=$

Output



$f(x,y)$ $\quad$ $a(x,y)$ $\quad$ $g(x,y)$

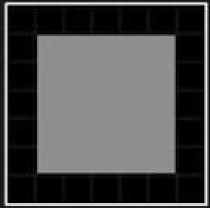
Result Image is saturated. Why?

Example: Averaging

Input

$f(x,y)$

$*$



"Box Filter"
5 x 5

$a(x,y)$

$=$

Output

$g(x,y)$

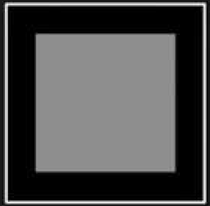
Sum of all the filter (kernel) weights should be 1.

Smoothing With Box Filter

Input

$f(x, y)$

$*$



"Box Filter"
21 x 21

$a(x, y)$

$=$

Output

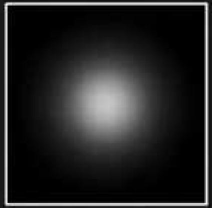
$g(x, y)$

Image smoothed with a box filter does not look "natural." Has blocky artifacts.

Smoothing With “Fuzzy” Filter

Input

$*$



“Fuzzy Filter”
21 x 21

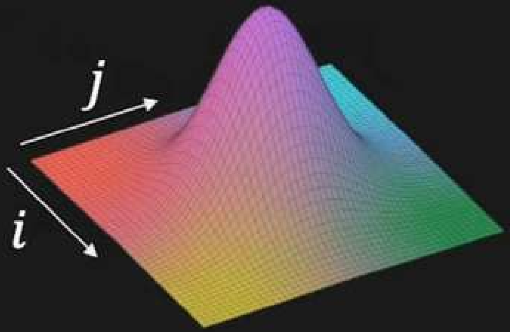
$=$

Output

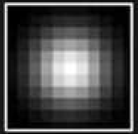
$f(x, y)$ $\quad$ $b(x, y)$ $\quad$ $g(x, y)$

Gaussian Kernel: A Fuzzy Filter

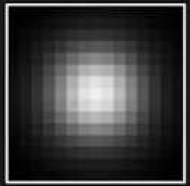
$$n_\sigma[i,j] = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}\left(\frac{i^2+j^2}{\sigma^2}\right)}$$



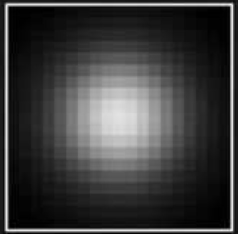
σ^2: Variance



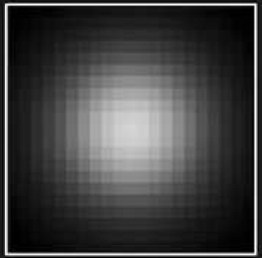
$\sigma = 2$



$\sigma = 3$



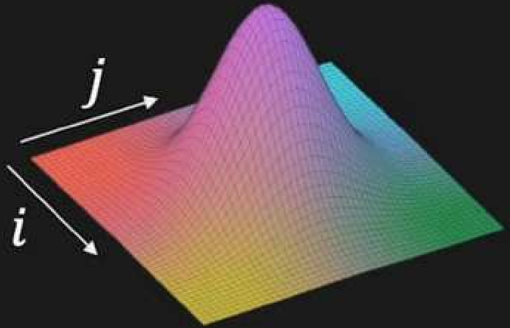
$\sigma = 4$



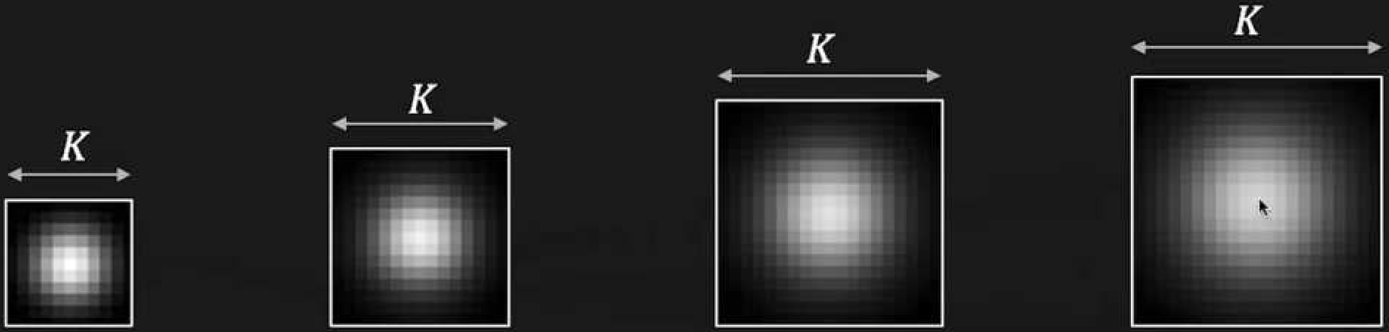
$\sigma = 5$

Gaussian Kernel: A Fuzzy Filter

$$n_\sigma[i,j] = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}\left(\frac{i^2+j^2}{\sigma^2}\right)}$$



σ^2: Variance



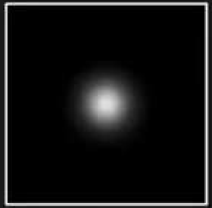
$\sigma = 2$ $\quad$ $\sigma = 3$ $\quad$ $\sigma = 4$ $\quad$ $\sigma = 5$

Rule of thumb: Set kernel size $K \approx 2\pi\sigma$

Gaussian Smoothing

Input

$*$



$\sigma = 8$

$=$

Output

$f(x,y)$ $\quad$ $n_8(x,y)$ $\quad$ $g(x,y)$

Larger the kernel (or σ), more the blurring

Gaussian Smoothing is Separable

$$g[i,j] = \frac{1}{2\pi\sigma^2} \sum_{m=1}^{K} \sum_{n=1}^{K} e^{-\frac{1}{2}\left(\frac{m^2+n^2}{\sigma^2}\right)} f[i-m, j-n]$$

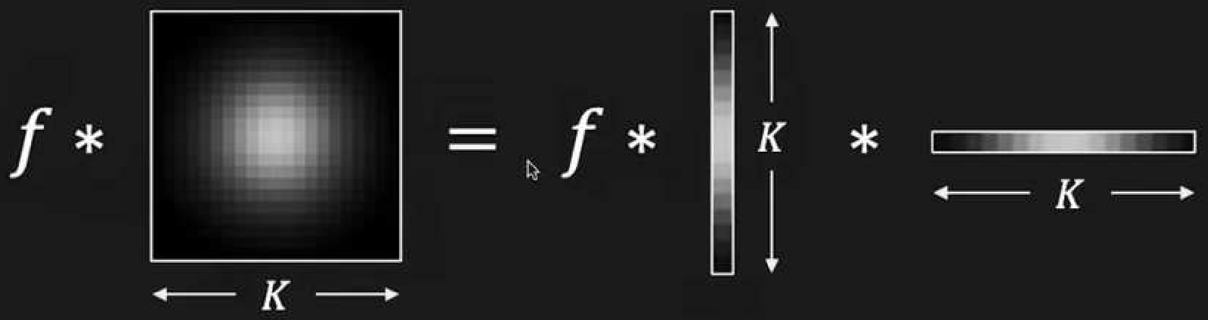
$$g[i,j] = \frac{1}{2\pi\sigma^2} \sum_{m=1}^{K} e^{-\frac{1}{2}\left(\frac{m^2}{\sigma^2}\right)} . \sum_{n=1}^{K} e^{-\frac{1}{2}\left(\frac{n^2}{\sigma^2}\right)} f[i-m, j-n]$$

Gaussian Smoothing is Separable

$$g[i,j] = \frac{1}{2\pi\sigma^2} \sum_{m=1}^{K} \sum_{n=1}^{K} e^{-\frac{1}{2}\left(\frac{m^2+n^2}{\sigma^2}\right)} f[i-m, j-n]$$

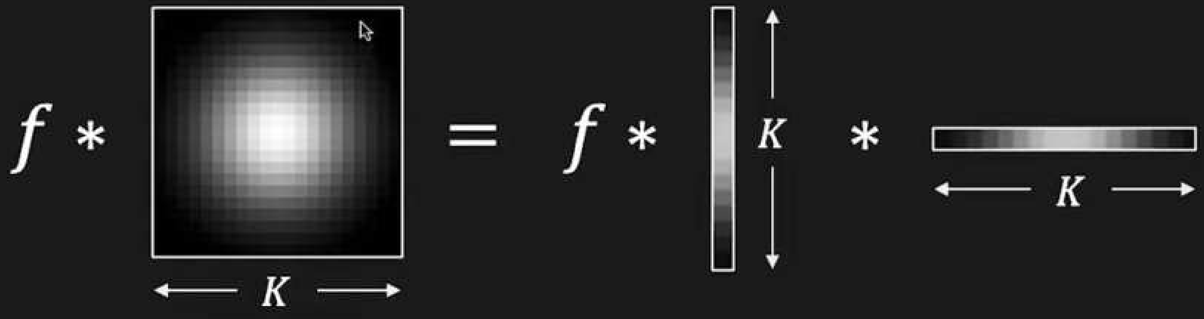
$$g[i,j] = \frac{1}{2\pi\sigma^2} \sum_{m=1}^{K} e^{-\frac{1}{2}\left(\frac{m^2}{\sigma^2}\right)} . \sum_{n=1}^{K} e^{-\frac{1}{2}\left(\frac{n^2}{\sigma^2}\right)} f[i-m, j-n]$$

Using One 2D Gaussian Filter ≡ Using Two 1D Gaussian Filters



Gaussian Smoothing is Separable

Using One 2D Gaussian Filter ≡ Using Two 1D Gaussian Filters

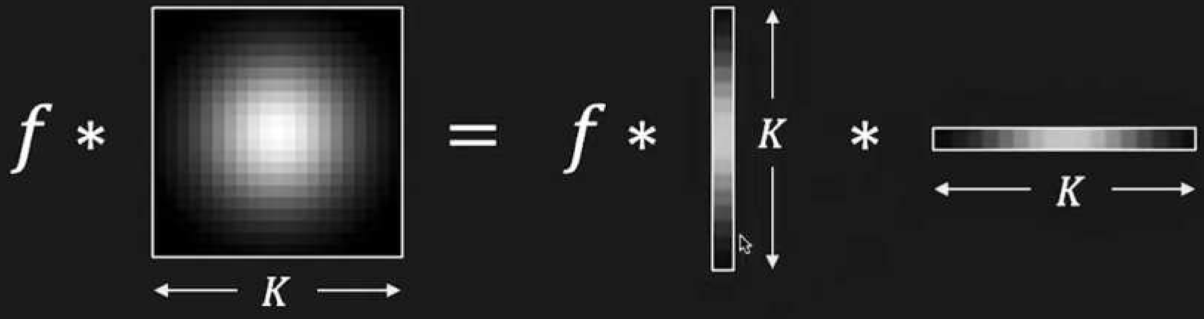


Which one is faster? Why?

K^2 Multiplications	$2K$ Multiplications
$K^2 - 1$ Additions	$2(K-1)$ Additions

Gaussian Smoothing is Separable

Using One 2D Gaussian Filter ≡ Using Two 1D Gaussian Filters



Which one is faster? Why?

K^2 Multiplications	$2K$ Multiplications
$K^2 - 1$ Additions	$2(K-1)$ Additions

Smoothing to Remove Image Noise

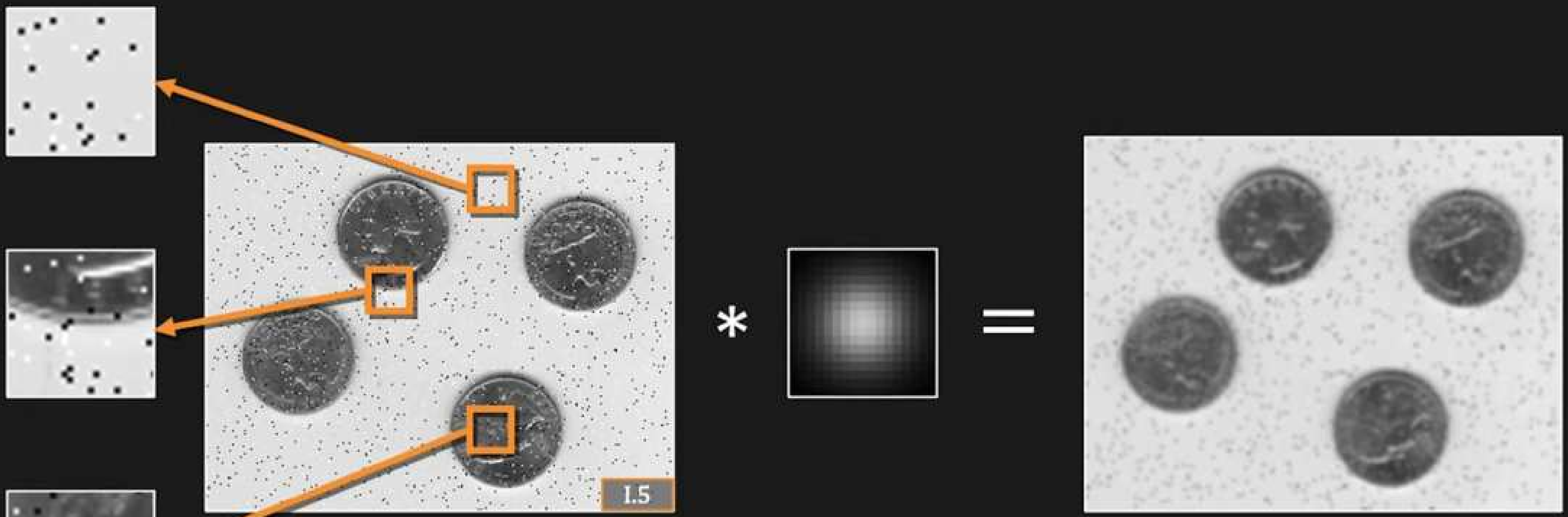


Image with Salt and Pepper Noise

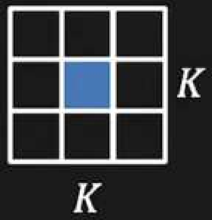
Gaussian Blurred Image

Problem with Smoothing:

- Does not remove outliers (Noise)
- Smooths edges (Blur)

Median Filtering

1. Sort the K^2 values in window centered at the pixel
2. Assign the Middle Value (Median) to pixel



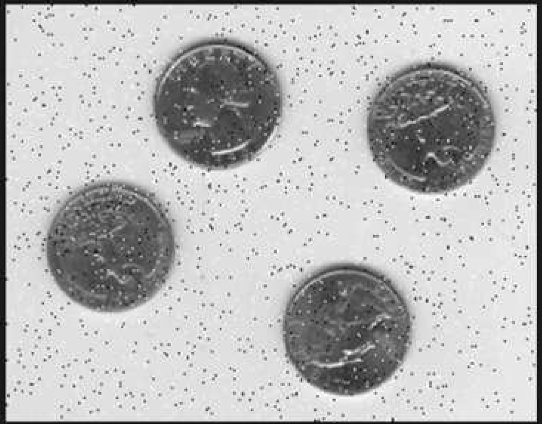


Image with
Salt and Pepper Noise

Median Filtered
Image ($K = 3$)

Non-linear Operation
(Cannot be implemented using convolution)

Median Filtering

Not Effective when Image Noise is not a Simple Salt and Pepper Noise.

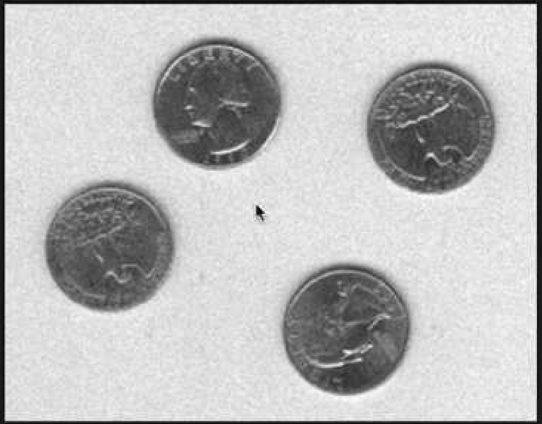
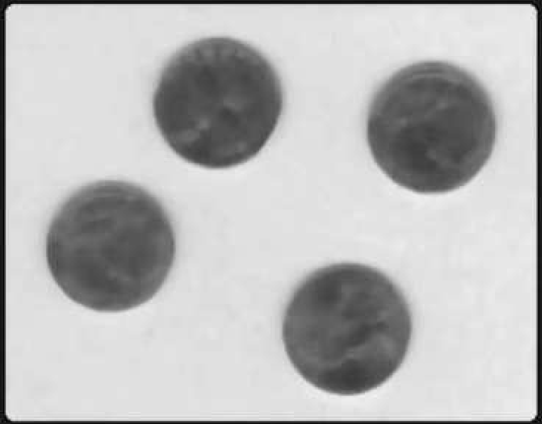


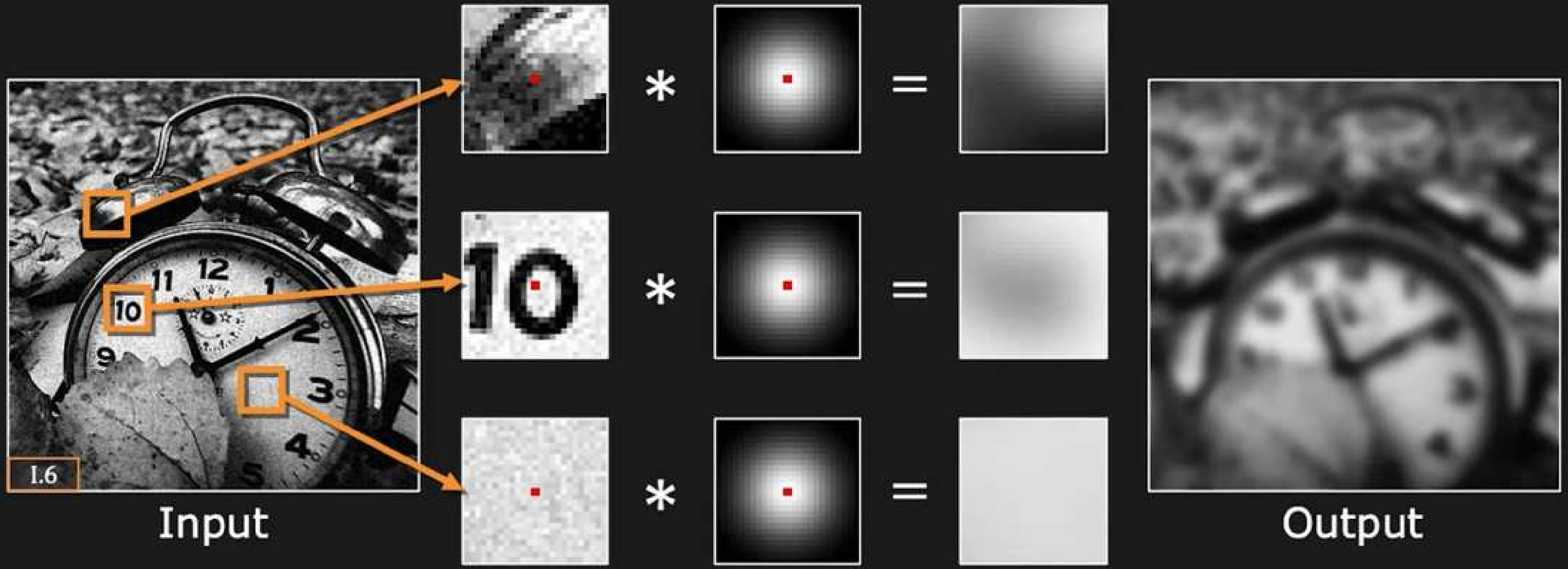
Image with Noise



Median Filtered Image ($K = 7$)

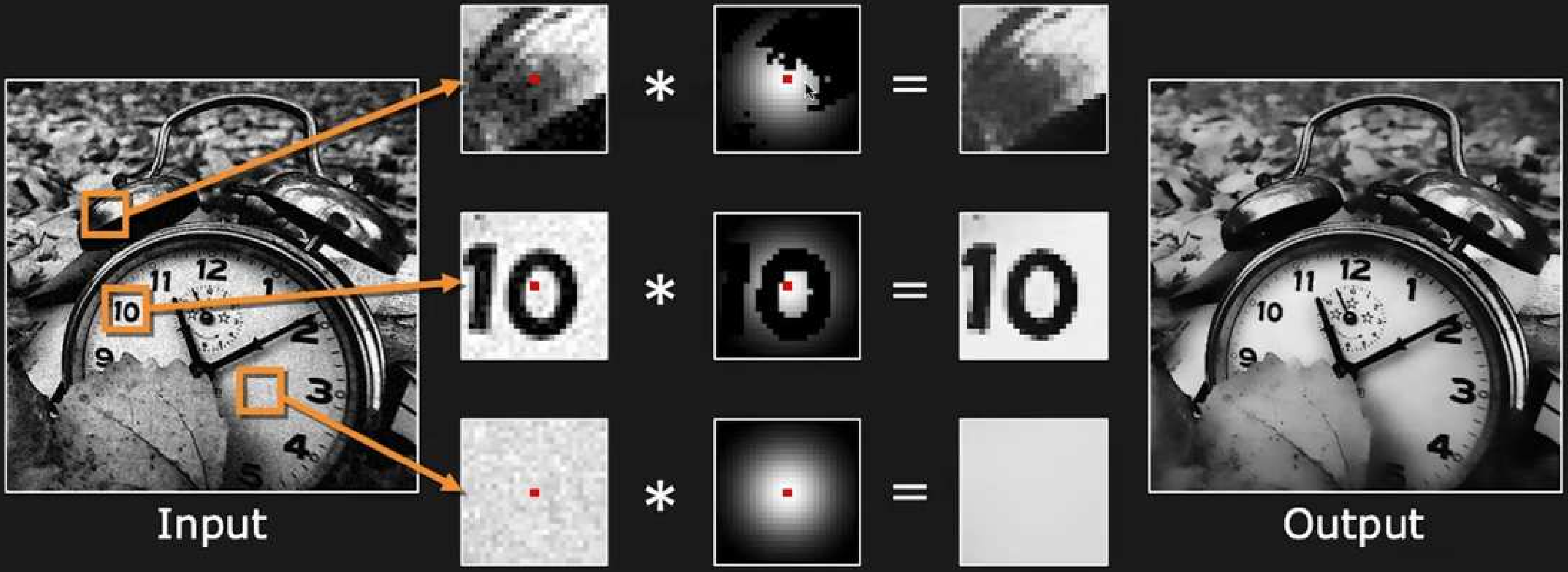
Larger K causes blurring of image detail

Revisiting Gaussian Smoothing



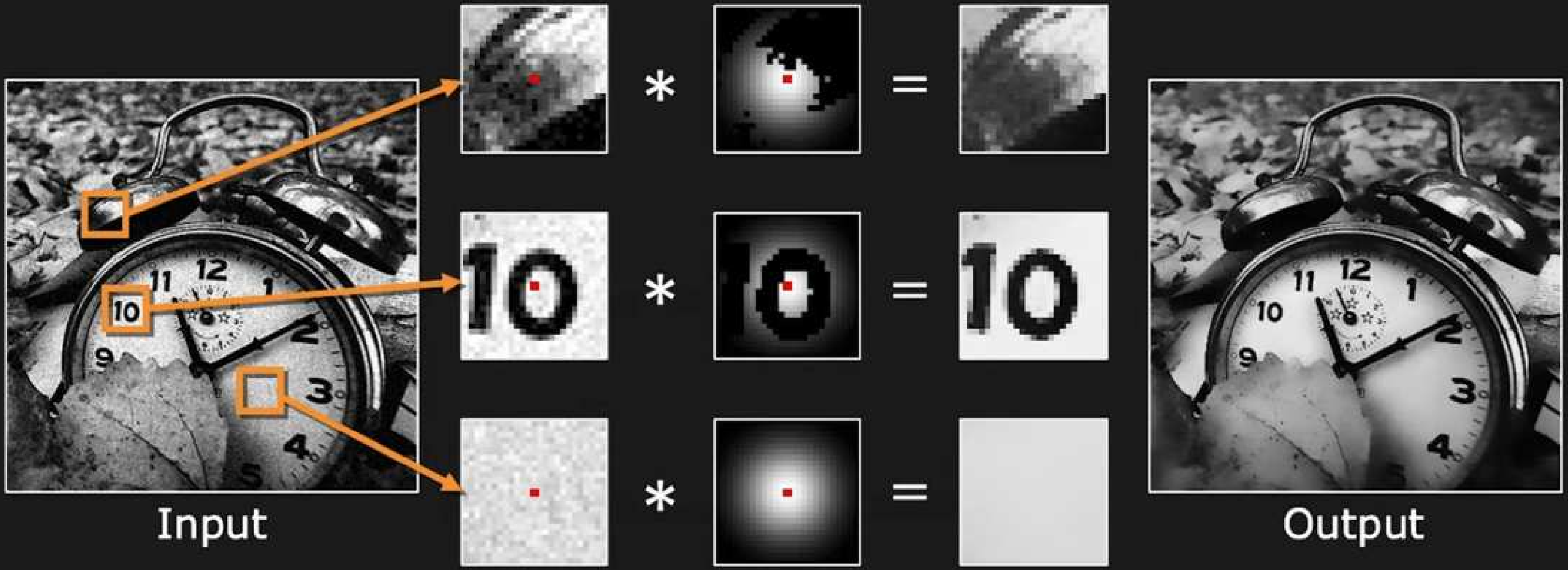
Same Gaussian kernel is used everywhere.
Blurs across edges.

Blur Similar Pixels Only



"Bias" Gaussian Kernel such that pixels not similar in intensity to the center pixel receive a lower weight.

Blur Similar Pixels Only

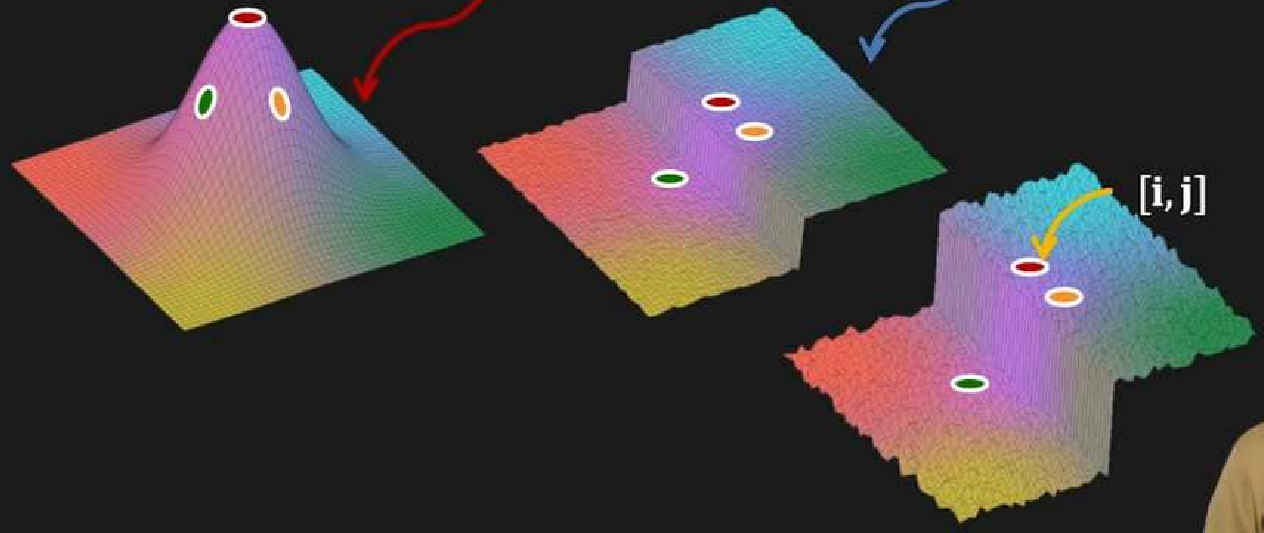


“Bias” Gaussian Kernel such that pixels not similar in intensity to the center pixel receive a lower weight.

Bilateral Filter: Add Bias to Gaussian

Spatial Gaussian Brightness Gaussian

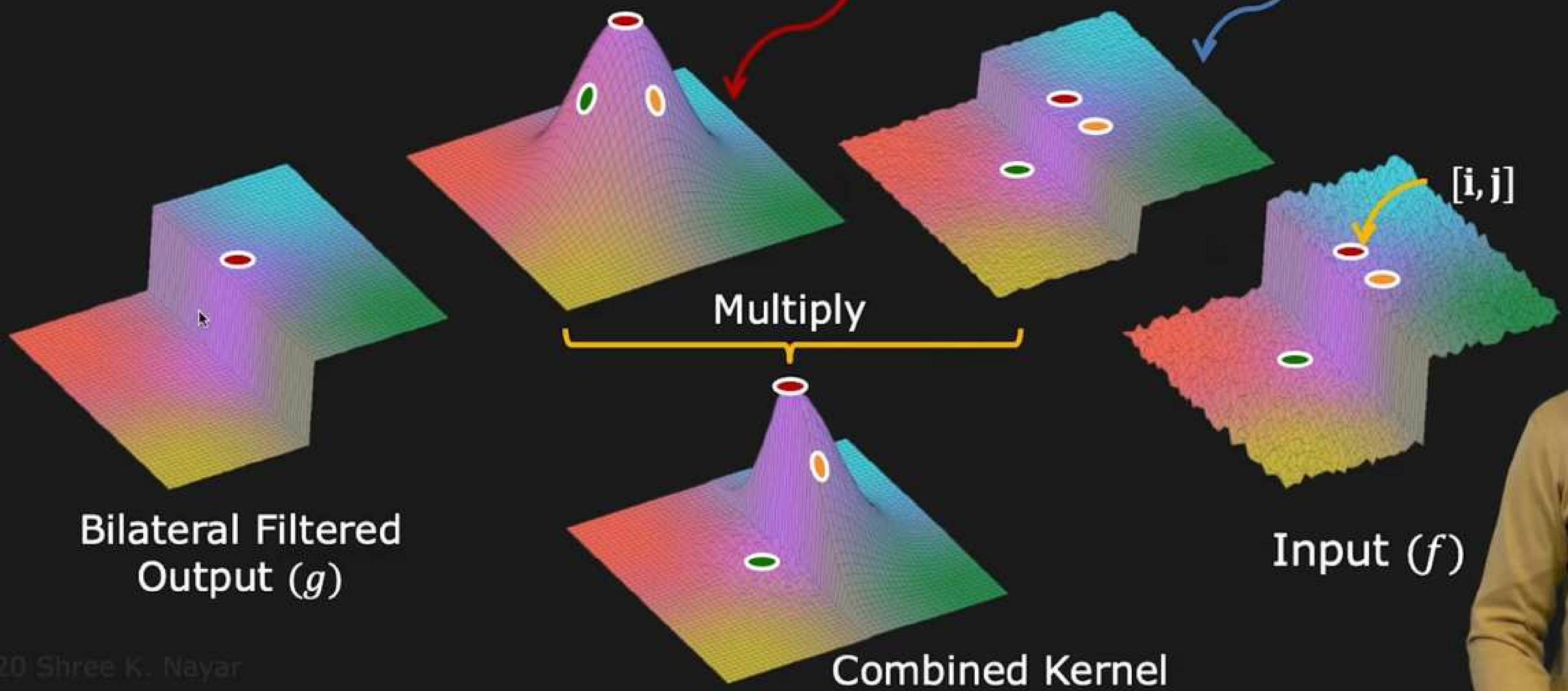
$$g[i,j] = \frac{1}{W_{sb}} \sum_{m} \sum_{n} f[m,n]\, \underbrace{n_{\sigma_s}[i-m,j-n]}\, \underbrace{n_{\sigma_b}(f[m,n]-f[i,j])}$$



Input (f)

Bilateral Filter: Add Bias to Gaussian

$$g[i,j] = \frac{1}{W_{sb}} \sum_{m} \sum_{n} f[m,n]\, \overbrace{n_{\sigma_s}[i-m, j-n]}^{\text{Spatial Gaussian}}\, \overbrace{n_{\sigma_b}(f[m,n] - f[i,j])}^{\text{Brightness Gaussian}}$$



Bilateral Filter: Summary

$$\boxed{g[i,j] = \frac{1}{W_{sb}} \sum_{m} \sum_{n} f[m,n]\, n_{\sigma_s}[i-m, j-n]\, n_{\sigma_b}(f[m,n] - f[i,j])}$$

Where:

$$n_{\sigma_s}[m,n] = \frac{1}{2\pi{\sigma_s}^2} e^{-\frac{1}{2}\left(\frac{m^2+n^2}{\sigma_s^2}\right)} \qquad n_{\sigma_b}(k) = \frac{1}{\sqrt{2\pi}\sigma_b} e^{-\frac{1}{2}\left(\frac{k^2}{\sigma_b^2}\right)}$$

$$W_{sb} = \sum_{m} \sum_{n} n_{\sigma_s}[i-m, j-n]\, n_{\sigma_b}(f[m,n] - f[i,j])$$

Gaussian vs. Bilateral Filtering: Example

Original

Gaussian
$\sigma_s = 2$

Bilateral
$\sigma_s = 2, \sigma_b = 10$

Bilateral Filtering: Changing σ_b

Bilateral
$\sigma_s = 6, \sigma_b = 10$

Bilateral
$\sigma_s = 6, \sigma_b = 20$

Bilateral
$\sigma_s = 6, \sigma_b = \infty$
(Gaussian Smoothin

Template Matching

Template

How do we locate the template in the image?

Template Matching

Template

How do we locate the template in the image?

Minimize:

$$E[i,j] = \sum_{m}\sum_{n}(f[m,n] - t[m-i,n-j])^2$$

$$E[i,j] = \sum_{m}\sum_{n}(f^2[m,n] + t^2[m-i,n-j] - \underbrace{2f[m,n]t[m-i,n-j]}_{\text{Maximize}})$$

Template Matching

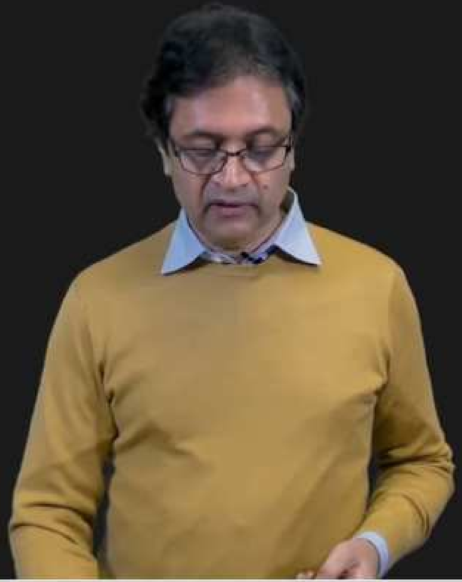
Template

How do we locate the template in the image?

Maximize:

$$R_{tf}[i,j] = \sum_{m}\sum_{n} f[m,n]\,t[m-i,n-j] = t \otimes f$$

(Cross-Correlation)



Convolution vs. Correlation

Convolution:

$$g[i,j] = \sum_{m}\sum_{n} f[m,n]\underline{t[i-m,j-n]} = t * f$$

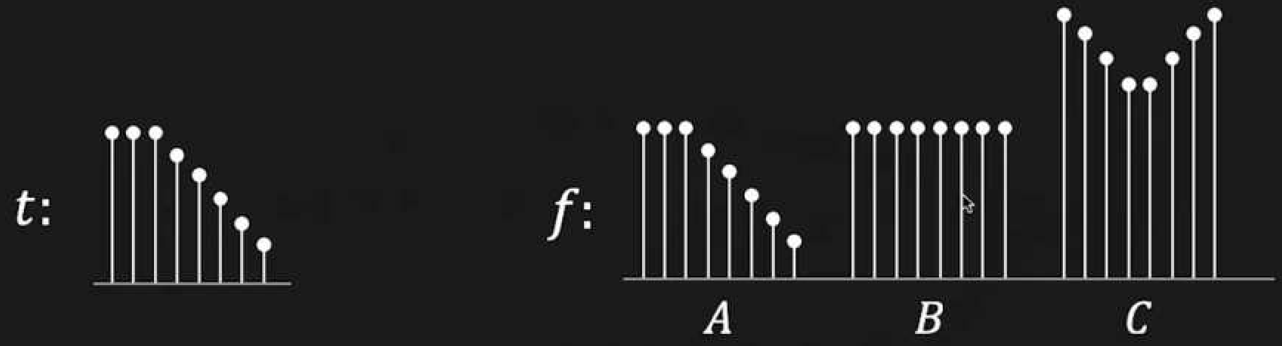
Correlation:

$$R_{tf}[i,j] = \sum_{m}\sum_{n} f[m,n]\underline{t[m-i,n-j]} = t \otimes f$$

No Flipping in Correlation

Problem with Cross-Correlation

$$R_{tf}[i,j] = \sum_m \sum_n f[m,n]t[m-i,n-j] = t \otimes f$$



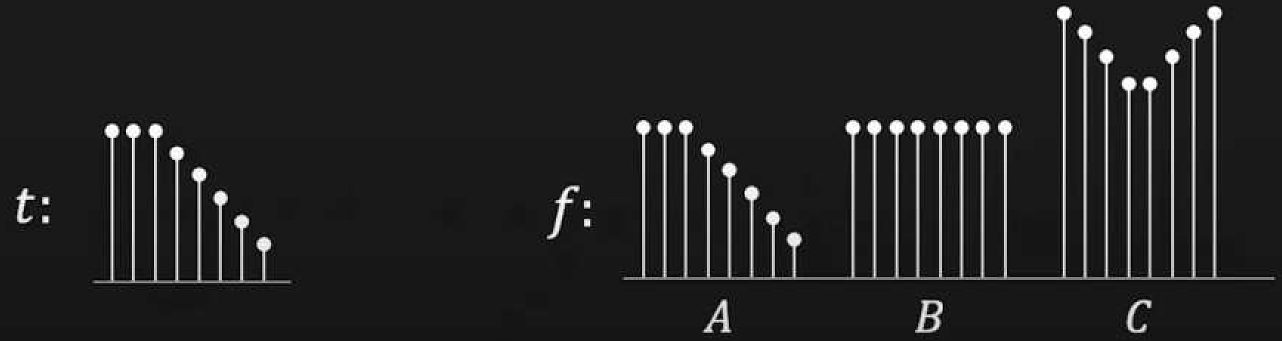
$$R_{tf}(C) > R_{tf}(B) > R_{tf}(A)$$

We need $R_{tf}(A)$ to be the maximum!

Normalized Cross-Correlation

Account for energy differences

$$N_{tf}[i,j] = \frac{\sum_m \sum_n f[m,n]t[m-i,n-j]}{\sqrt{\sum_m \sum_n f^2[m,n]}\ \sqrt{\sum_m \sum_n t^2[m-i,n-j]}}$$



$$N_{tf}(A) > N_{tf}(B) > N_{tf}(C)$$

Normalized Cross-Correlation

Account for energy differences

$$N_{tf}[i,j] = \frac{\sum_m \sum_n f[m,n]\,t[m-i,n-j]}{\sqrt{\sum_m \sum_n f^2[m,n]}\,\sqrt{\sum_m \sum_n t^2[m-i,n-j]}}$$

Recap: Image Processing I

Transform image to new one that is clearer or easier to analyze.

Topics:

(1) Pixel Processing

(2) LSIS and Convolution

(3) Linear Image Filters

(4) Non-Linear Image Filters

(5) Template Matching by Correlation

Image Processing II

Transform image to new one that is clearer or easier to analyze.

Topics:

(1) Fourier Transform

(2) Convolution Theorem

Image Processing II

Transform image to new one that is clearer or easier to analyze.

Topics:

(1) Fourier Transform

(2) Convolution Theorem

(3) Deconvolution in Frequency Domain

(4) Sampling Theory and Aliasing

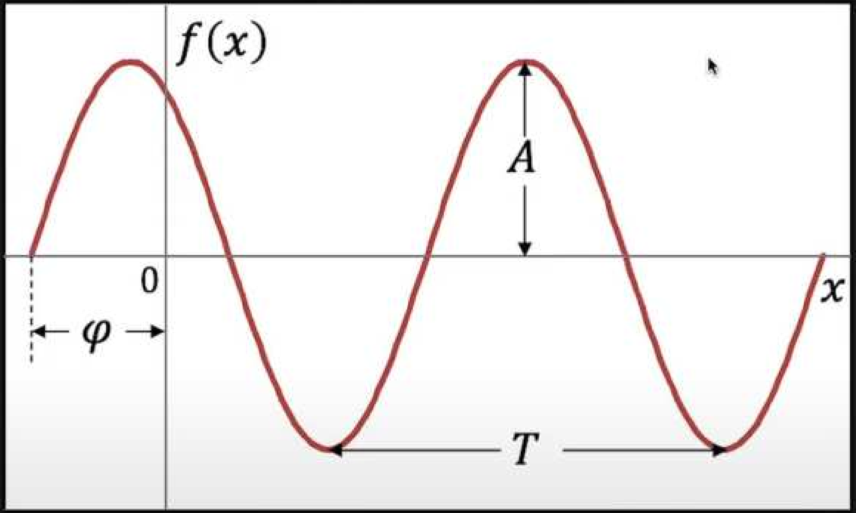
Jean Baptiste Joseph Fourier

(1768-1830)

Any Periodic Function can be rewritten as a Weighted Sum of Infinite Sinusoids of Different Frequencies.

Sinusoid

$$f(x) = A\sin(2\pi u x + \varphi)$$



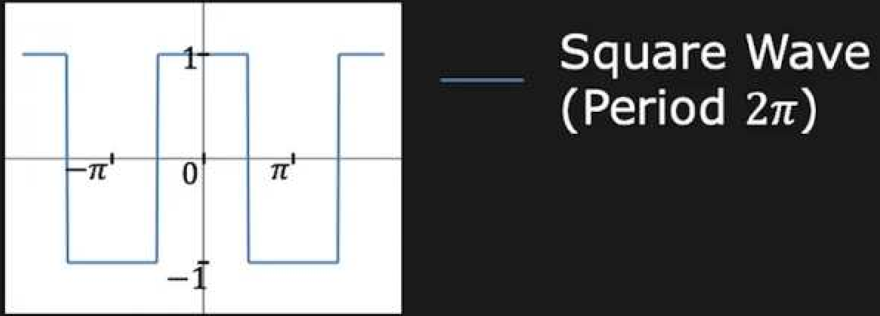
A: Amplitude

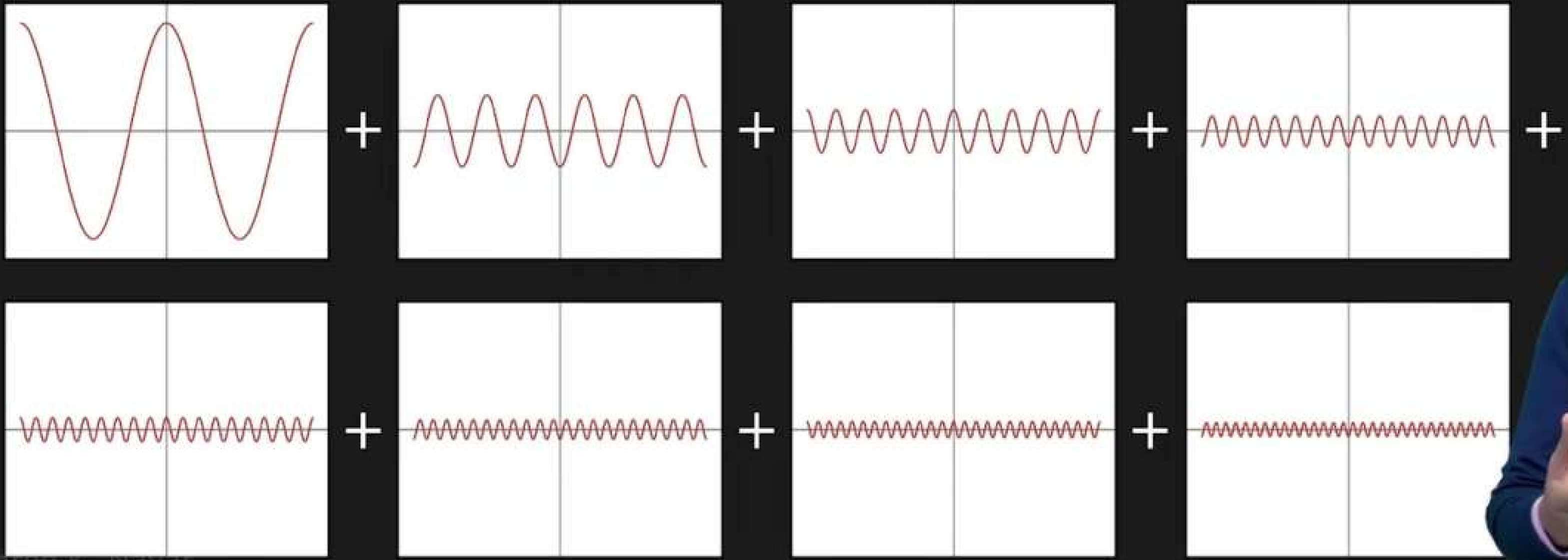
T: Period

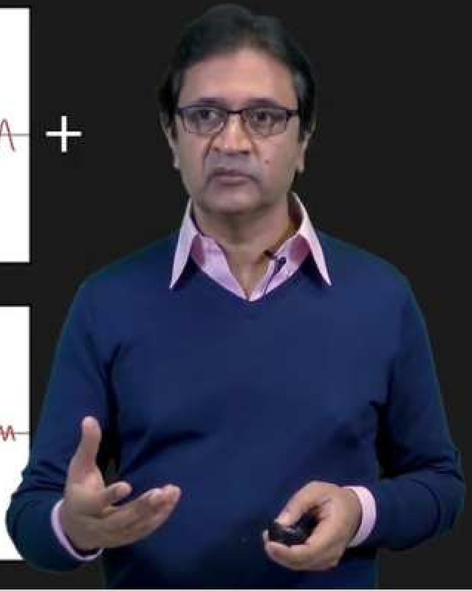
φ: Phase

u: Frequency $(1/T)$

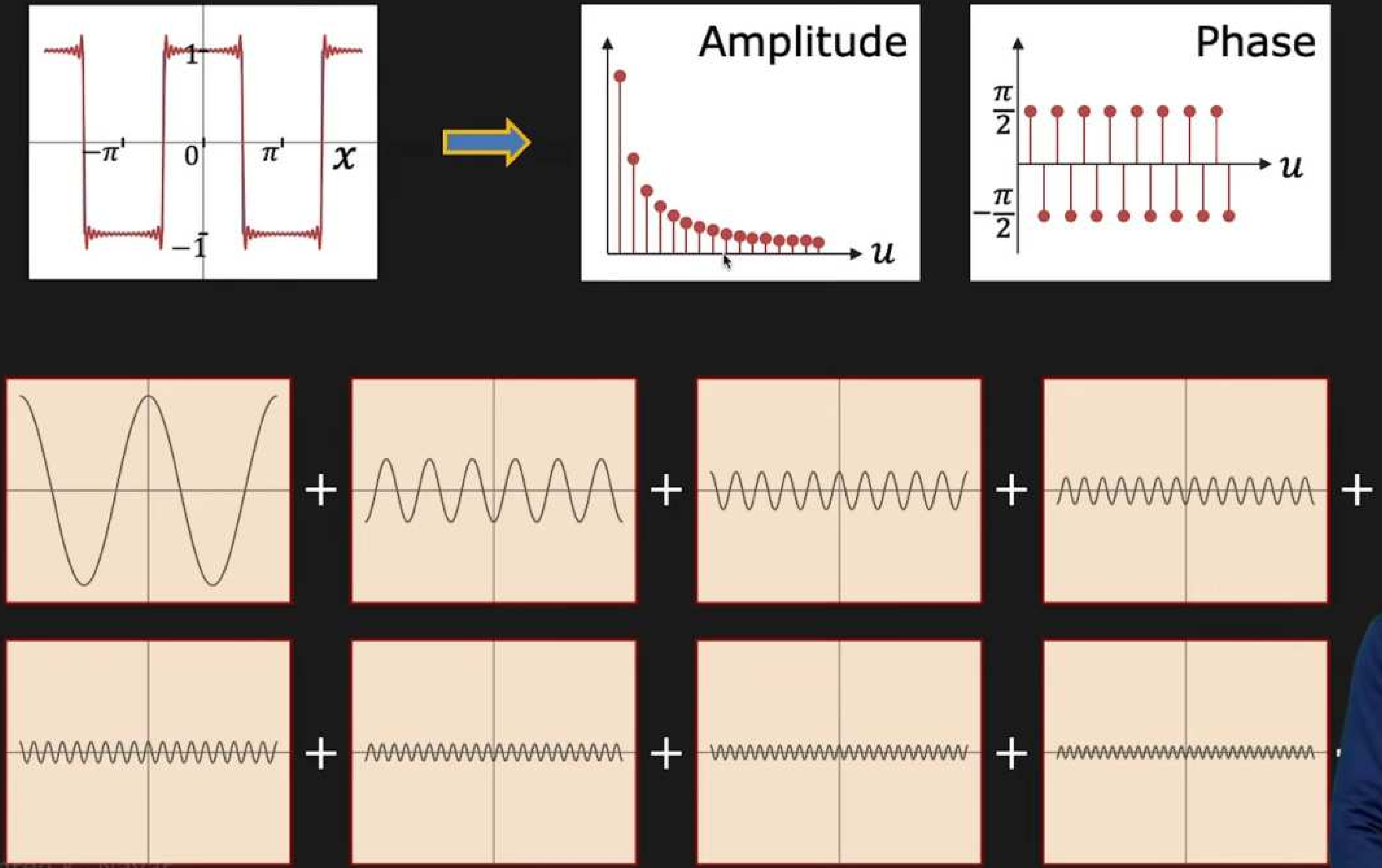
Fourier Series



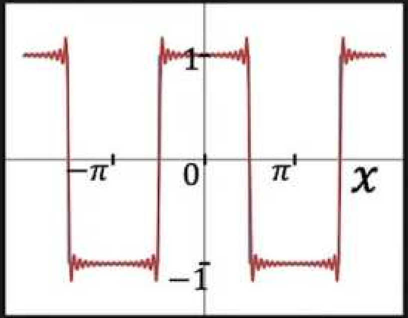


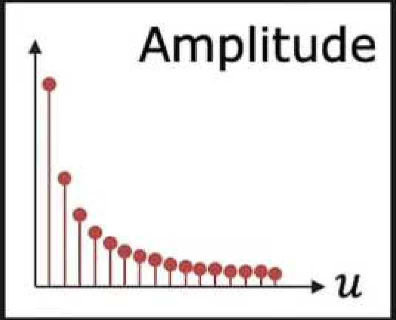


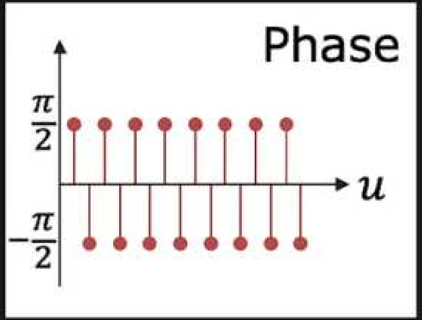
Frequency Representation of Signal



Fourier Transform (FT)



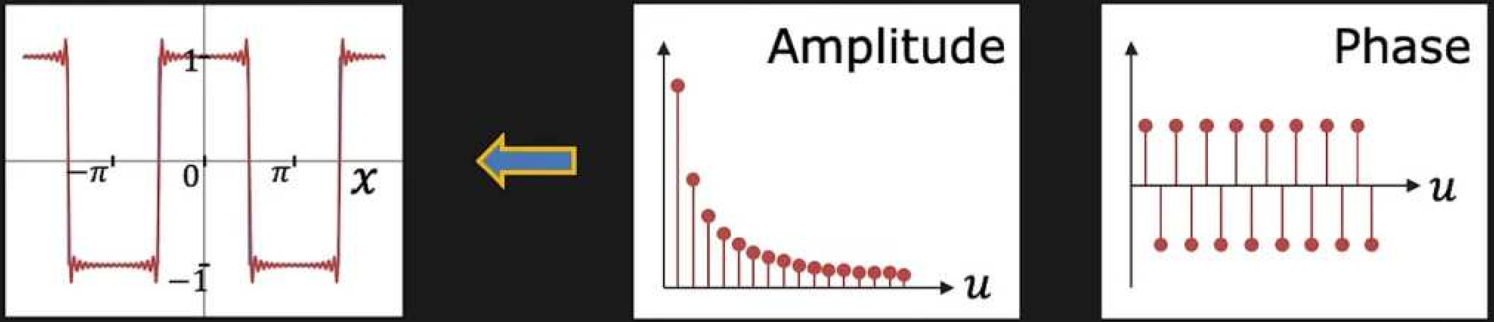




Represents a signal $f(x)$ in terms of Amplitudes and Phases of its Constituent Sinusoids.

$$f(x) \longrightarrow \boxed{\mathrm{FT}} \longrightarrow F(u)$$

Inverse Fourier Transform (IFT)



Computes the signal $f(x)$ from the Amplitudes and Phases of its Constituent Sinusoids.

$$f(x) \longleftarrow \boxed{\text{IFT}} \longleftarrow F(u)$$

Finding FT and IFT

Fourier Transform:

$$\boxed{F(u) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi ux}\, dx}$$

x: space

u: frequency

$e^{i\theta} = \cos\theta + i\sin\theta$

$i = \sqrt{-1}$

Inverse Fourier Transform:

$$\boxed{f(x) = \int_{-\infty}^{\infty} F(u) e^{i2\pi ux}\, du}$$

Complex Exponential (Euler Formula)

$$\boxed{e^{i\theta} = \cos\theta + i\sin\theta} \qquad i = \sqrt{-1}$$

Expand $e^{i\theta}$ using Taylor Series:

$$e^{i\theta} = 1 + i\theta + \frac{(i\theta)^2}{2!} + \frac{(i\theta)^3}{3!} + \frac{(i\theta)^4}{4!} + \frac{(i\theta)^5}{5!} + \frac{(i\theta)^6}{6!} + \ldots$$

$$e^{i\theta} = \underbrace{\left(1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \cdots\right)}_{\cos\theta} + i\underbrace{\left(\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \frac{\theta^7}{7!} + \cdots\right)}_{\sin\theta}$$

Fourier Transform is Complex!

$F(u)$ holds the Amplitude and Phase of the sinusoid of frequency u.

$$\boxed{F(u) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi ux} dx}$$

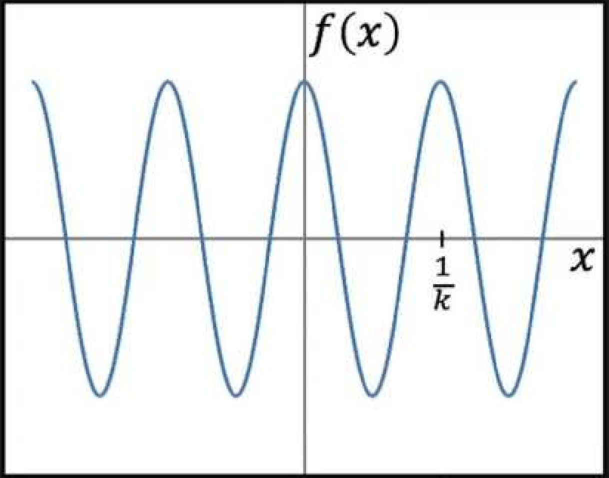
$$F(u) = \Re e\{F(u)\} + i\,\Im m\{F(u)\}$$

Amplitude: $A(u) = \sqrt{\Re e\{F(u)\}^2 + \Im m\{F(u)\}^2}$

Phase: $\varphi(u) = \mathrm{atan2}(\Im m\{F(u)\}, \Re e\{F(u)\})$

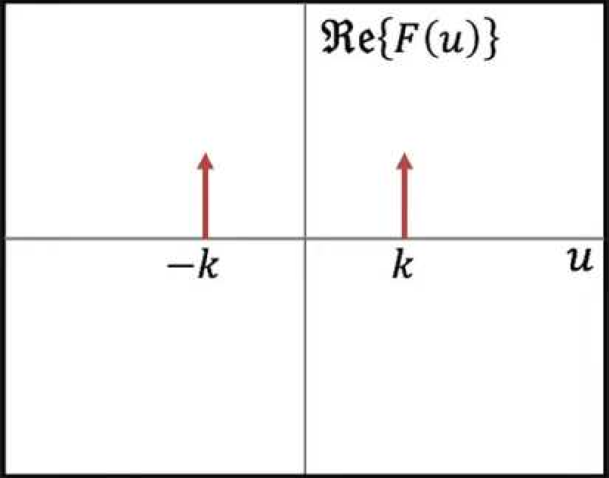
Fourier Transform Examples

Signal $f(x)$



$$f(x) = \cos 2\pi kx$$

Fourier Transform $F(u)$

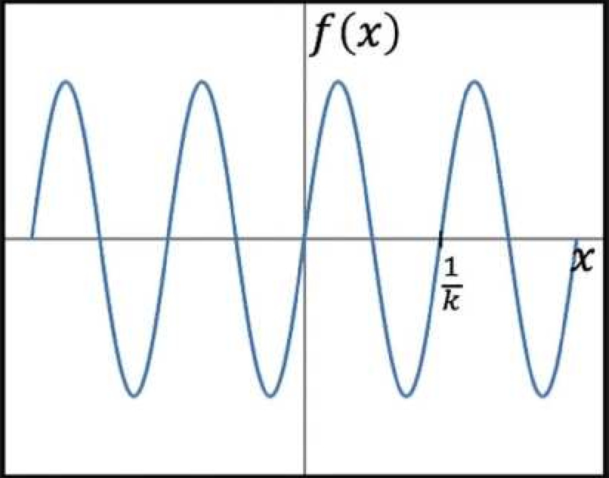


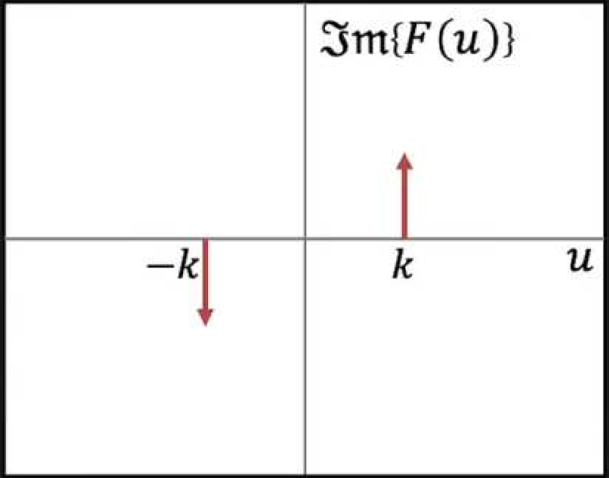
$$F(u) = \frac{1}{2}[\delta(u+k) + \delta(u-k)]$$

Fourier Transform Examples

Signal $f(x)$

Fourier Transform $F(u)$



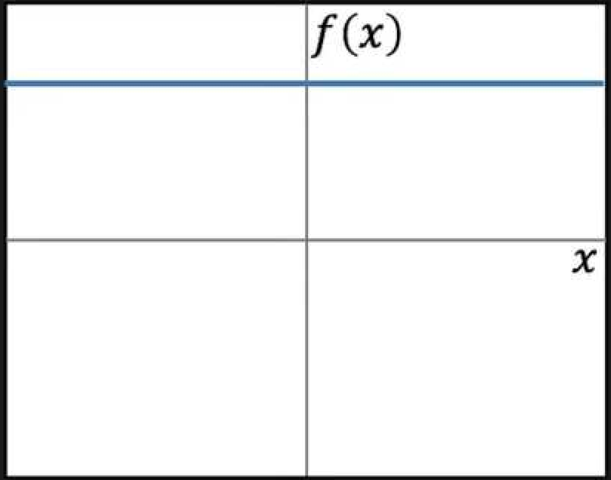


$$f(x) = \sin 2\pi kx$$

$$F(u) = \tfrac{1}{2} i [\delta(u+k) - \delta(u-k)]$$

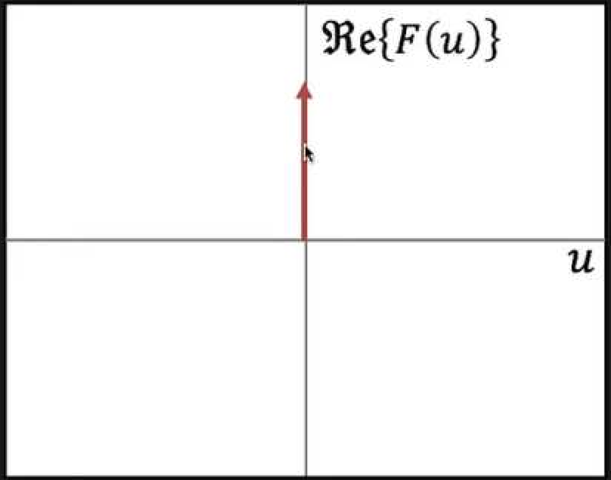
Fourier Transform Examples

Signal $f(x)$



$$f(x) = 1$$

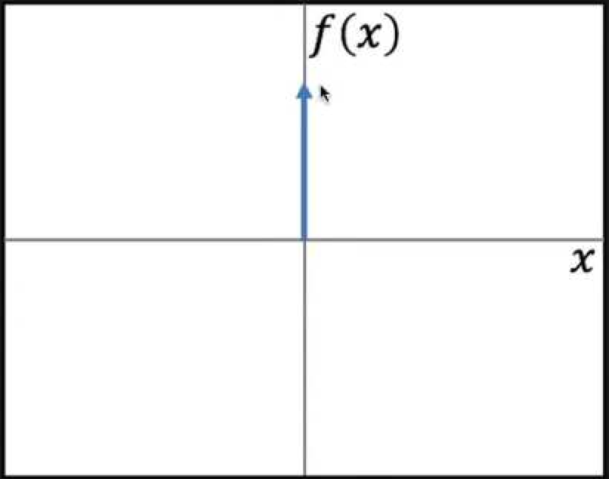
Fourier Transform $F(u)$



$$F(u) = \delta(u)$$

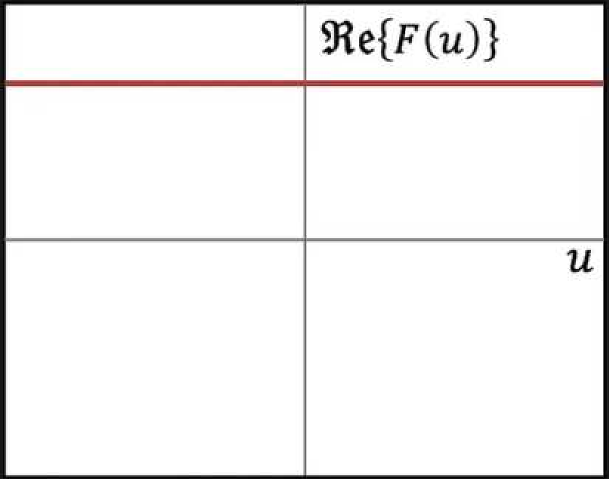
Fourier Transform Examples

Signal $f(x)$



$$f(x) = \delta(x)$$

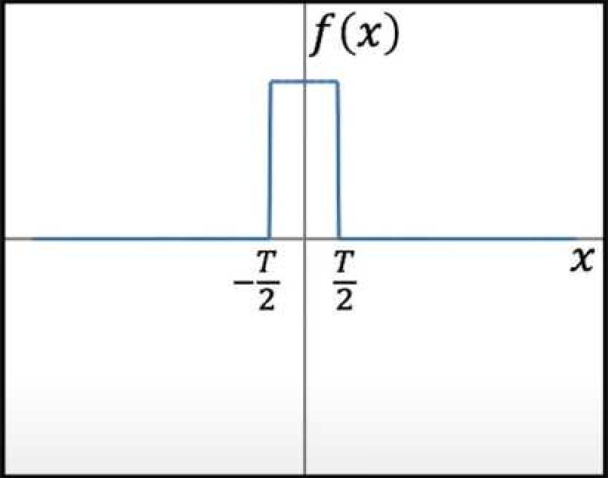
Fourier Transform $F(u)$



$$F(u) = 1$$

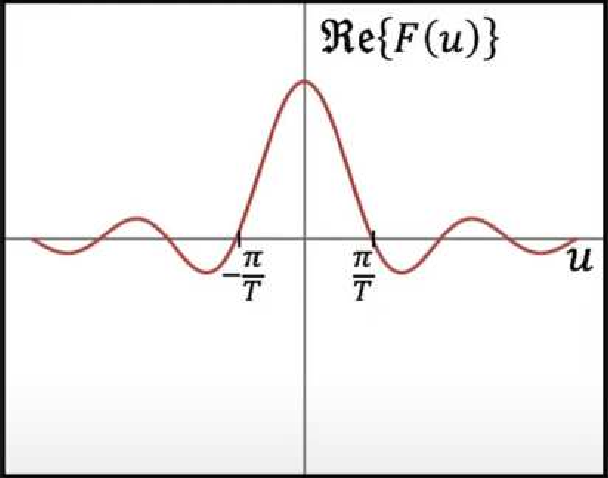
Fourier Transform Examples

Signal $f(x)$



$$f(x) = \mathrm{Rect}\left(\frac{x}{T}\right)$$

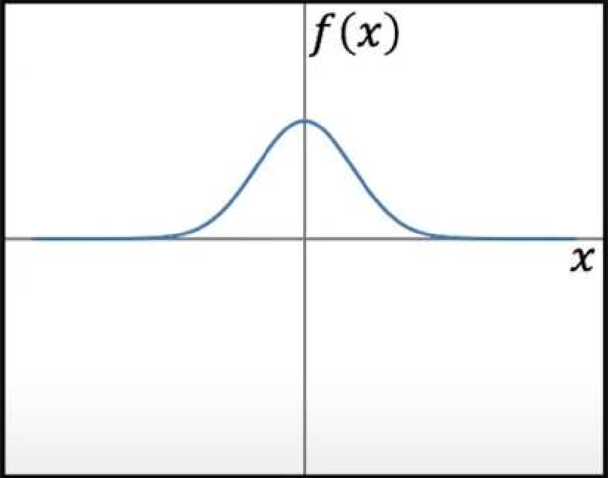
Fourier Transform $F(u)$



$$F(u) = T\,\mathrm{sinc}\,Tu$$

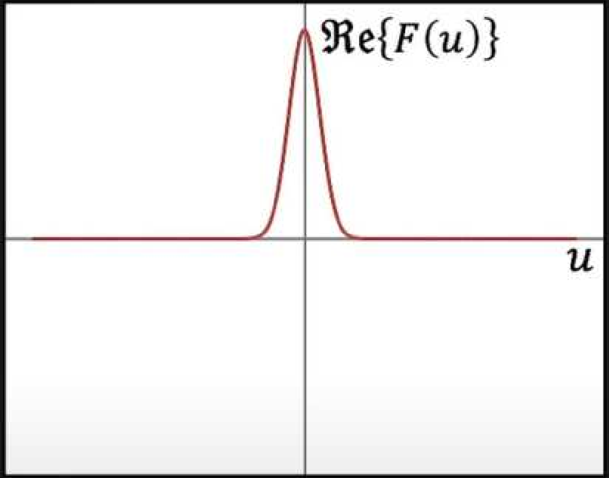
Fourier Transform Examples

Signal $f(x)$



$$f(x) = e^{-ax^2}$$

Fourier Transform $F(u)$



$$F(u) = \sqrt{\pi/a}\; e^{-\pi^2 u^2/a}$$

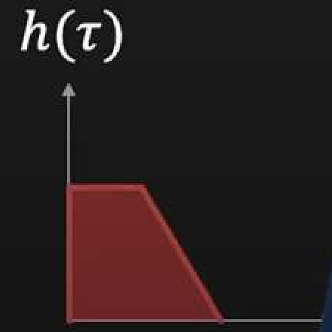
Properties of Fourier Transform

Property	Spatial Domain	Frequency Domain
Linearity	$\alpha f_1(x) + \beta f_2(x)$	$\alpha F_1(u) + \beta F_2(u)$
Scaling	$f(ax)$	$\frac{1}{\lvert a \rvert} F\left(\frac{u}{a}\right)$
Shifting	$f(x-a)$	$e^{-i2\pi ua} F(u)$
Differentiation	$\frac{d^n}{dx^n}(f(x))$	$(i2\pi u)^n F(u)$

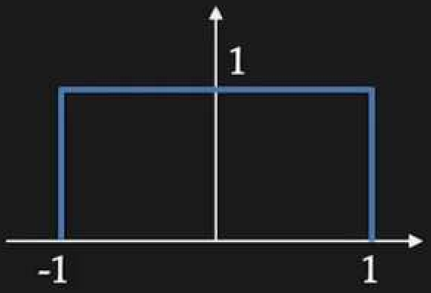
Convolution

Convolution of two functions $f(x)$ and $h(x)$

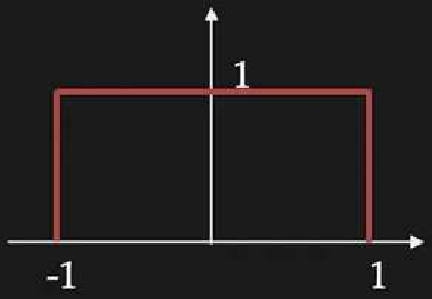
$$\boxed{g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau}$$



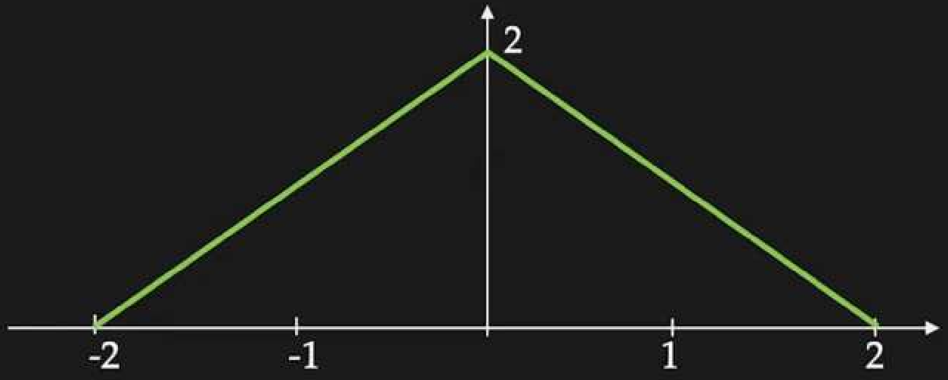
Convolution: Example



$f(x)$



$h(x)$



$f(x) * h(x)$

REVIEW

Convolution and Fourier Transform

Convolution: $g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau) h(x-\tau)\, d\tau$

Fourier Transform of $g(x)$:

$$G(u) = \int_{-\infty}^{\infty} g(x) e^{-i2\pi ux} dx$$

$$G(u) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\tau) h(x-\tau) e^{-i2\pi ux} d\tau dx$$

$$G(u) = \int_{-\infty}^{\infty} f(\tau) e^{-i2\pi u\tau} d\tau \int_{-\infty}^{\infty} h(x-\tau) e^{-i2\pi u(x-\tau)}\, dx$$

Convolution and Fourier Transform

Convolution: $g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau$

Fourier Transform of $g(x)$:

$$G(u) = \int_{-\infty}^{\infty} g(x)e^{-i2\pi ux}dx$$

$$G(u) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\tau)h(x-\tau)e^{-i2\pi ux}d\tau dx$$

$$G(u) = \underbrace{\int_{-\infty}^{\infty} f(\tau)e^{-i2\pi u\tau}d\tau}_{F(u)} \; \underbrace{\int_{-\infty}^{\infty} h(x-\tau)e^{-i2\pi u(x-\tau)}\,dx}_{H(u)}$$

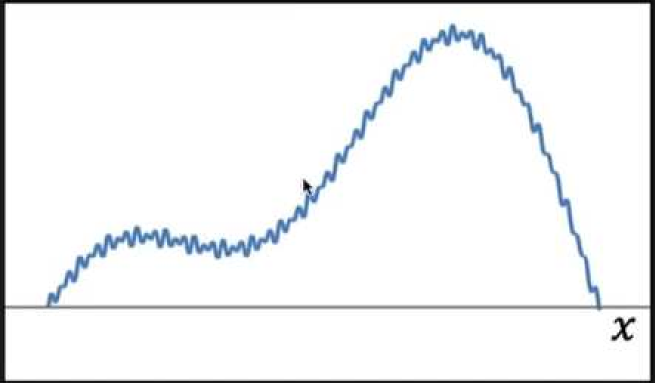
Convolution and Fourier Transform

Spatial Domain		Frequency Domain
$g(x) = f(x) * h(x)$ Convolution	$\longleftrightarrow$	$G(u) = F(u)\,H(u)$ Multiplication
$g(x) = f(x)\,h(x)$ Multiplication	$\longleftrightarrow$	$G(u) = F(u) * H(u)$ Convolution

Convolution Using Fourier Transform

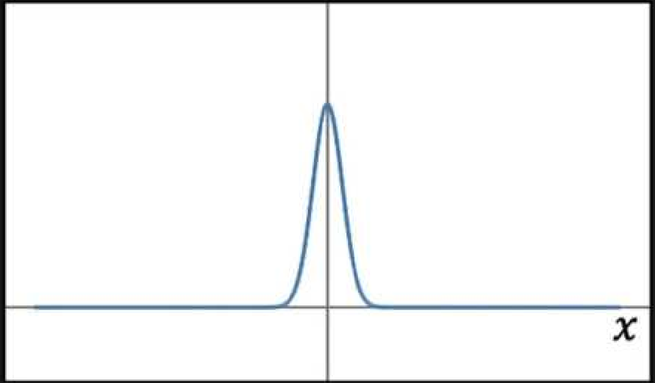
$$\begin{tikzcd}
g(x) & = & f(x) \arrow[d, "\mathrm{FT}"] & * & h(x) \arrow[d, "\mathrm{FT}"] \\
G(u) \arrow[u, "\mathrm{IFT}"] & = & F(u) & \times & H(u)
\end{tikzcd}$$

Gaussian Smoothing in Fourier Domain



Noisy Signal $f(x)$

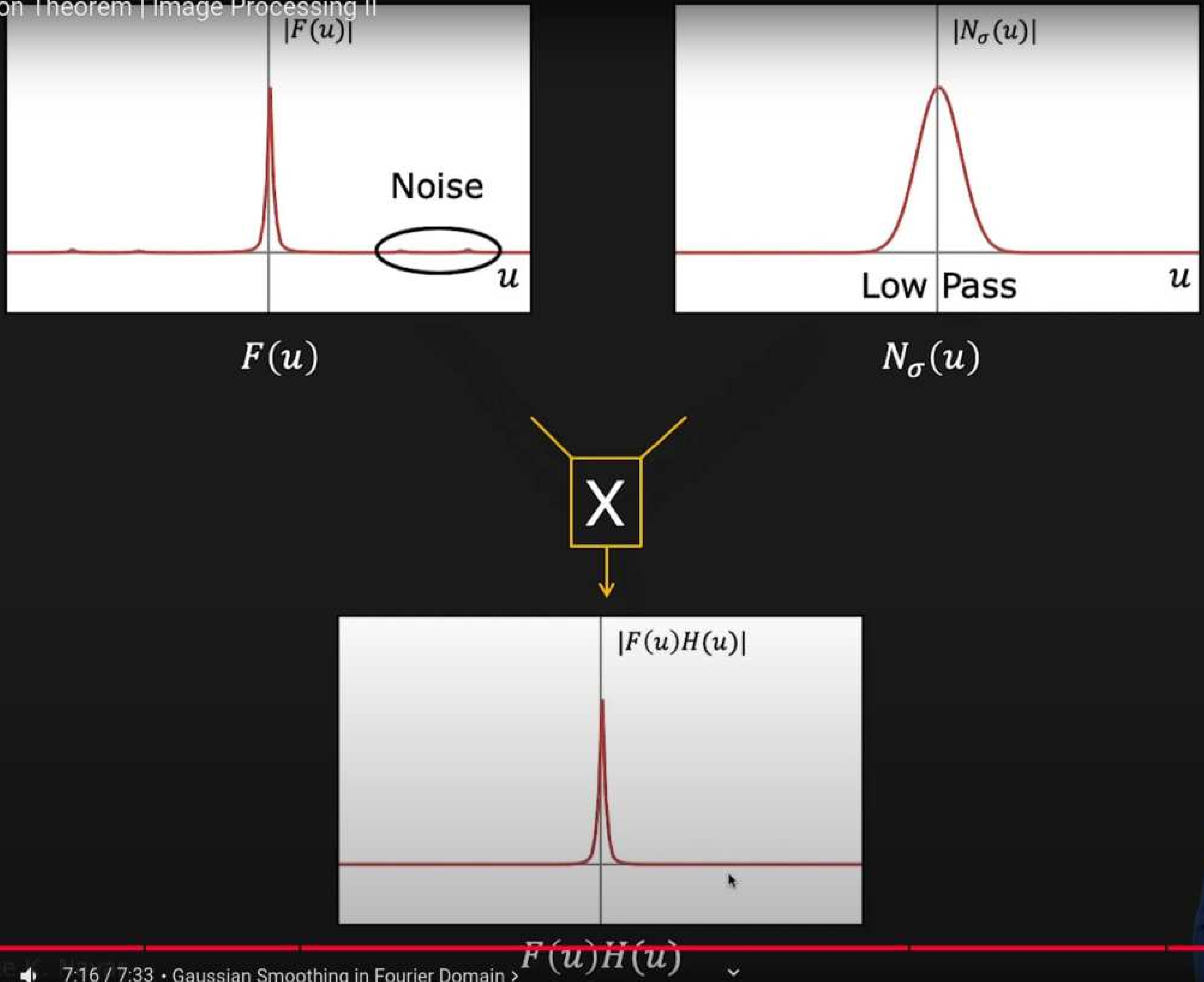
$*$



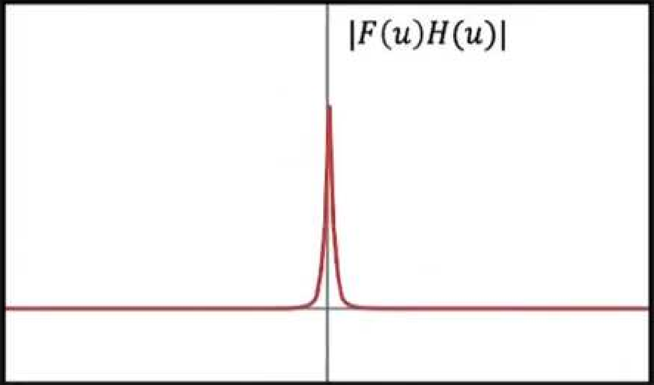
Gaussian Kernel $n_\sigma(x)$

Convolve the Noisy Signal with a Gaussian Kernel

Convolution Theorem | Image Processing II

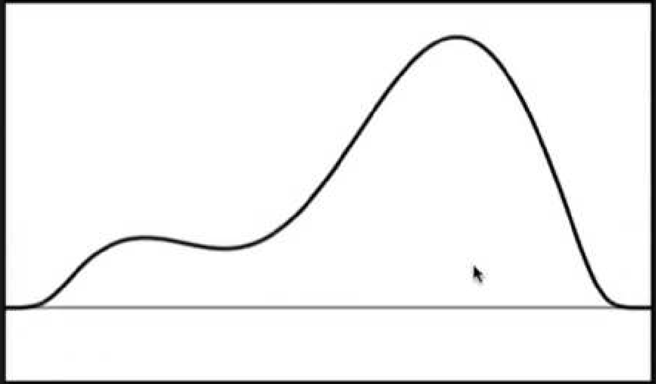


7:16 / 7:33 • Gaussian Smoothing in Fourier Domain ›

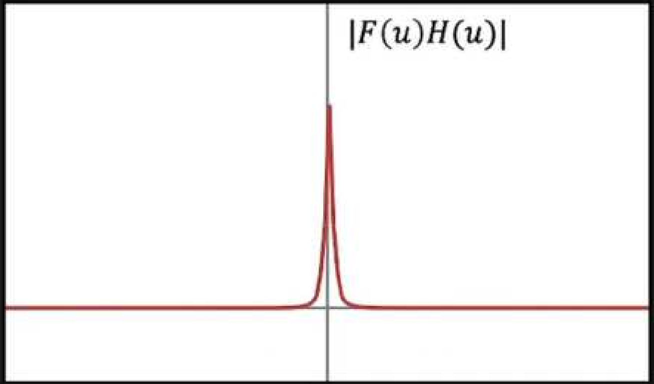


$F(u)H(u)$

IFT

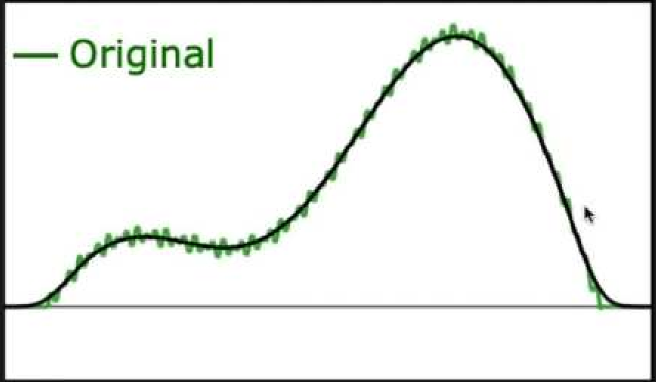


Gaussian Blurred Signal $g(x)$



$F(u)H(u)$

IFT



Gaussian Blurred Signal $g(x)$

2D Fourier Transform

Fourier Transform:

$$F(u,v) = \iint_{-\infty}^{\infty} f(x,y) e^{-i2\pi(ux+vy)} dx dy$$

u and v are frequencies along x and y, respectively

Inverse Fourier Transform:

$$f(x,y) = \iint_{-\infty}^{\infty} F(u,v) e^{i2\pi(xu+yv)} du dv$$

2D Fourier Transform: Discrete Images

Discrete Fourier Transform (DFT):

$$F[p,q] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[m,n] e^{-i2\pi pm/M} e^{-i2\pi qn/N}$$

$p = 0 \ldots M-1$

$q = 0 \ldots N-1$

p and q are frequencies along m and n, respectively

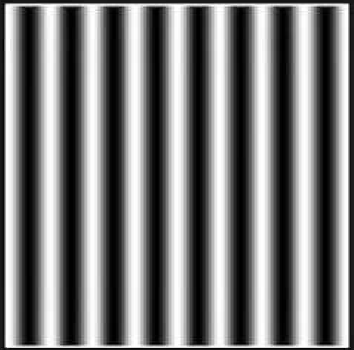
Inverse Discrete Fourier Transform (IDFT):

$$f[m,n] = \frac{1}{MN} \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} F[p,q] e^{i2\pi pm/M} e^{i2\pi qn/N}$$

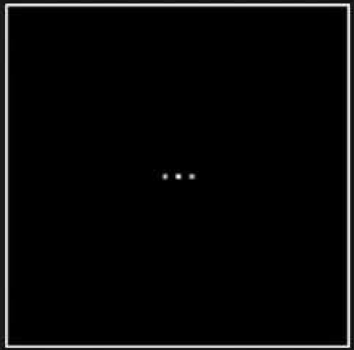
$m = 0 \ldots M -$

$n = 0 \ldots N -$

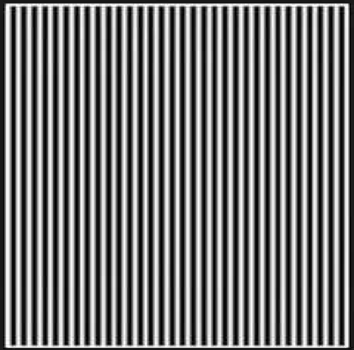
2D Fourier Transform: Example 1



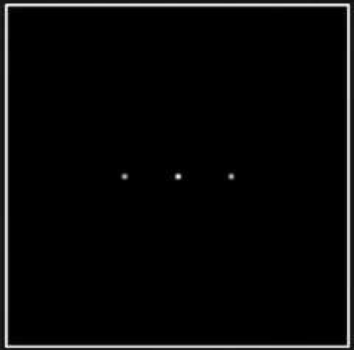
$f(m,n)$



$\log(|F(p,q)|)$



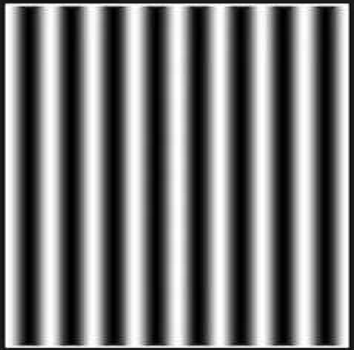
$g(m,n)$



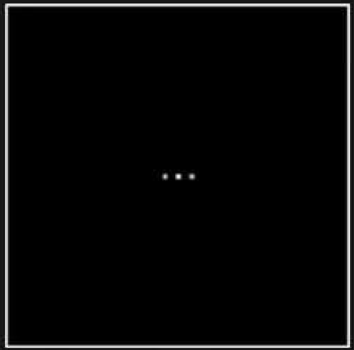
$\log(|G(p,q)|)$

Note: $\log(|F|)$ is used just for display

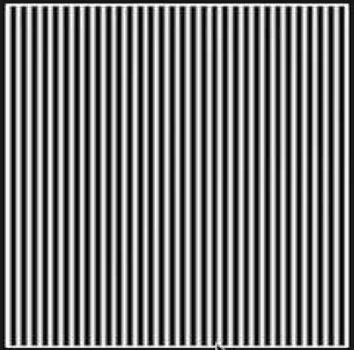
2D Fourier Transform: Example 1



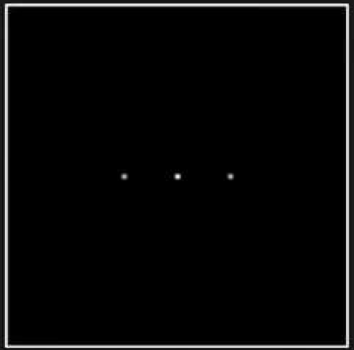
$f(m,n)$



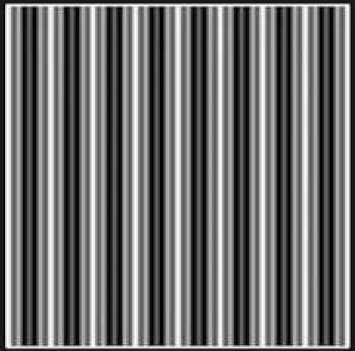
$\log(|F(p,q)|)$



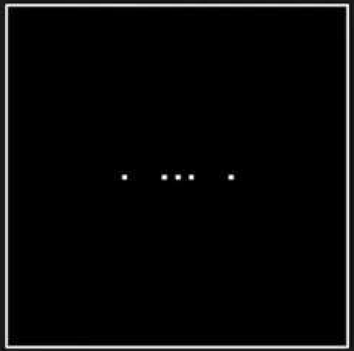
$g(m,n)$



$\log(|G(p,q)|)$



$f(m,n) + g(m,n)$

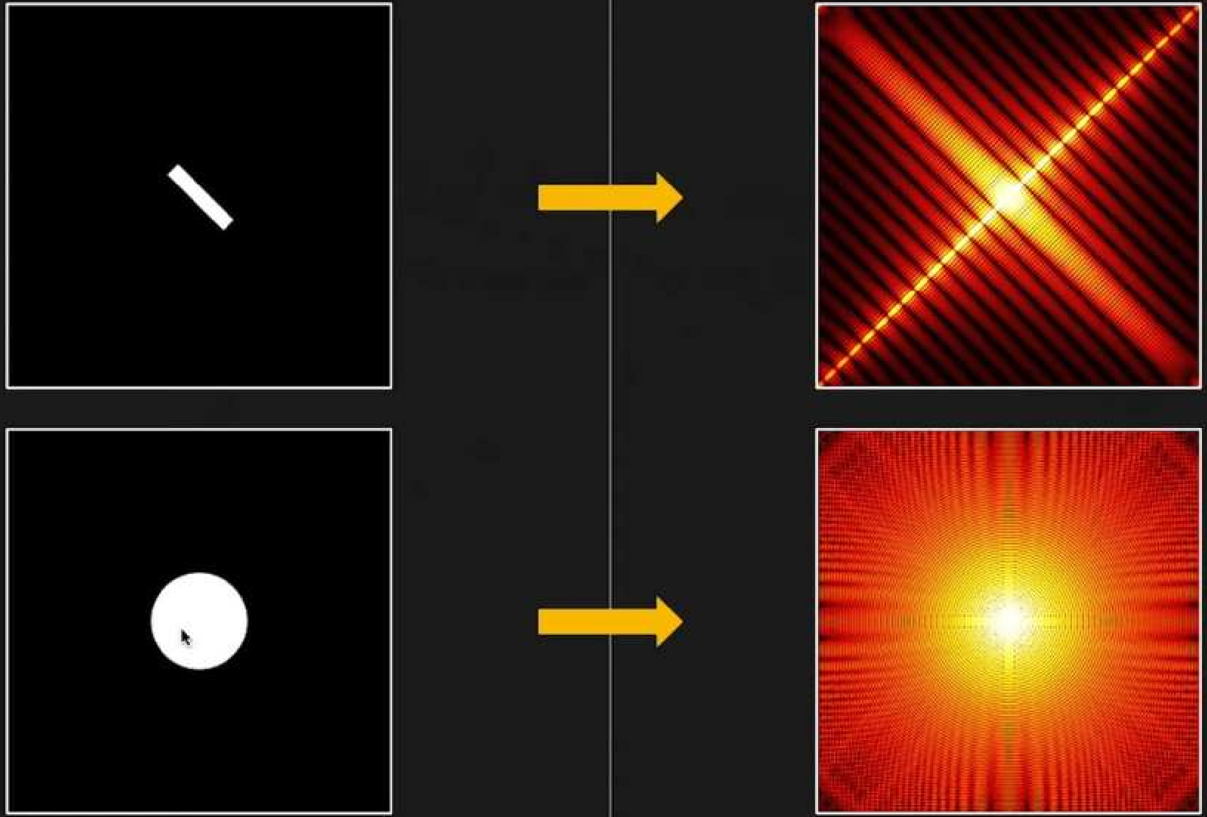


$\log(|F(p,q) + G(p,q)|)$

Note: $\log(|F|)$ is used just for dis

2D Fourier Transform: Example 2

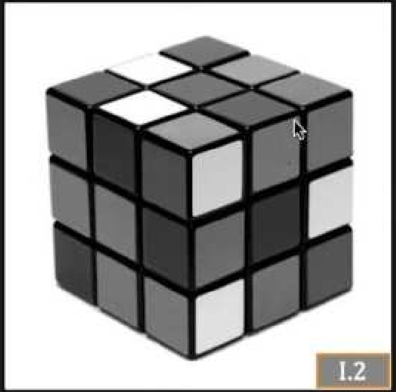
$f(m,n)$ $\qquad$ $\log(|F(p,q)|)$

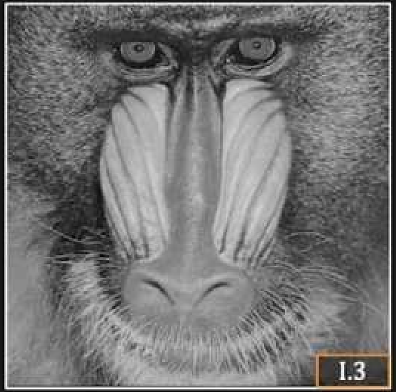


Min  Max

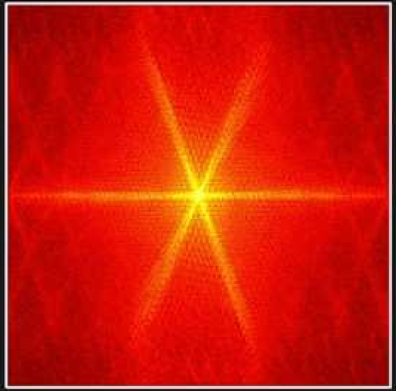
2D Fourier Transform: Example 3

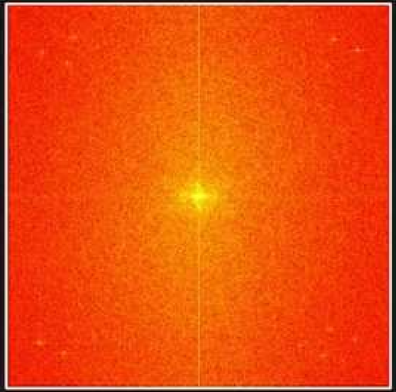
$f(m,n)$





$\log(|F(p,q)|)$

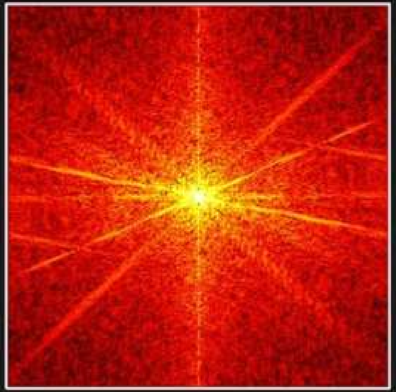


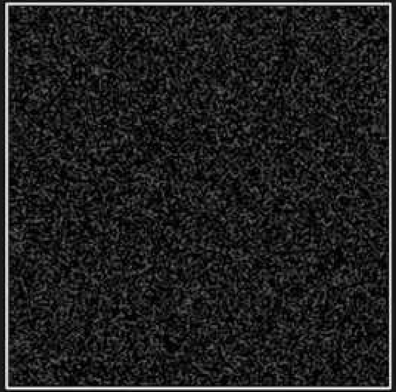


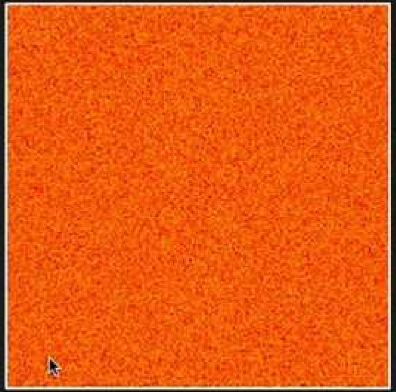
Min  Max

2D Fourier Transform: Example 4

$f(m,n)$ → $\log(|F(p,q)|)$



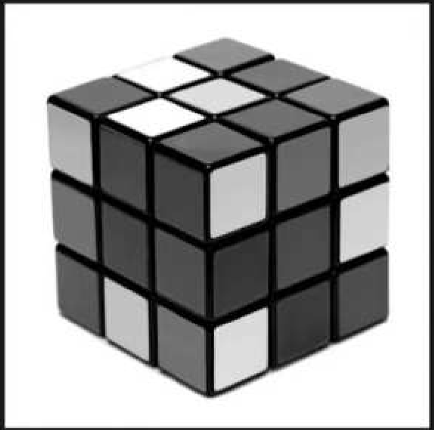




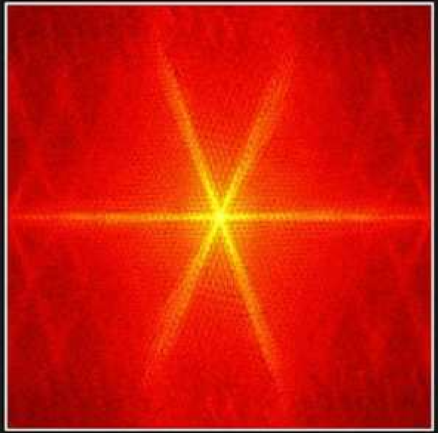
Min  Max

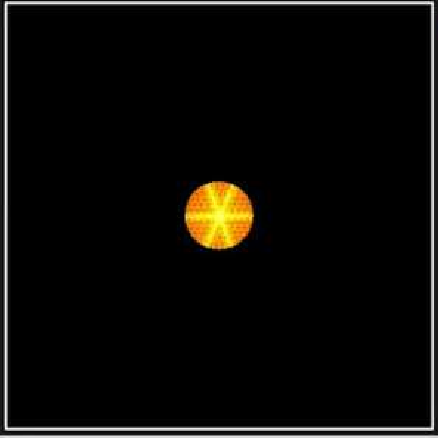
Low Pass Filtering

$f(m,n)$



$\log(|F(p,q)|)$

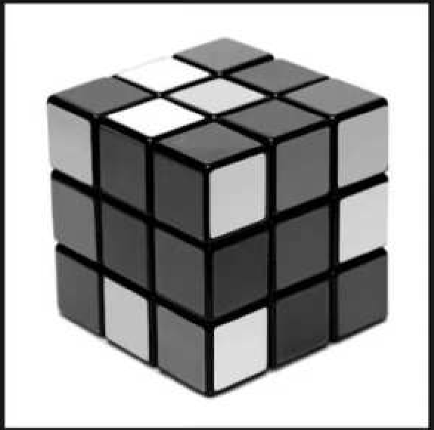


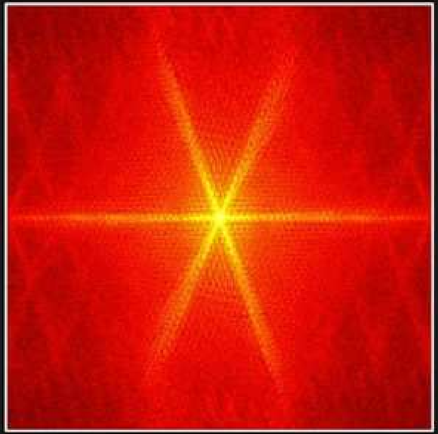


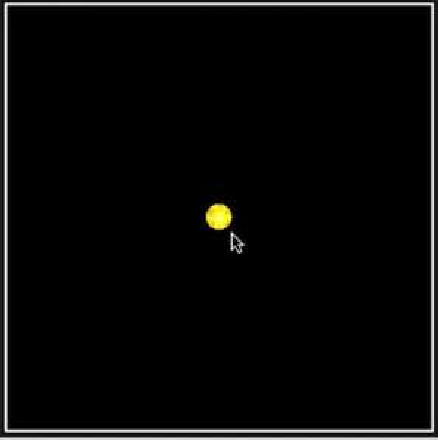
Low Pass Filtering

$f(m,n)$

$\log(|F(p,q)|)$

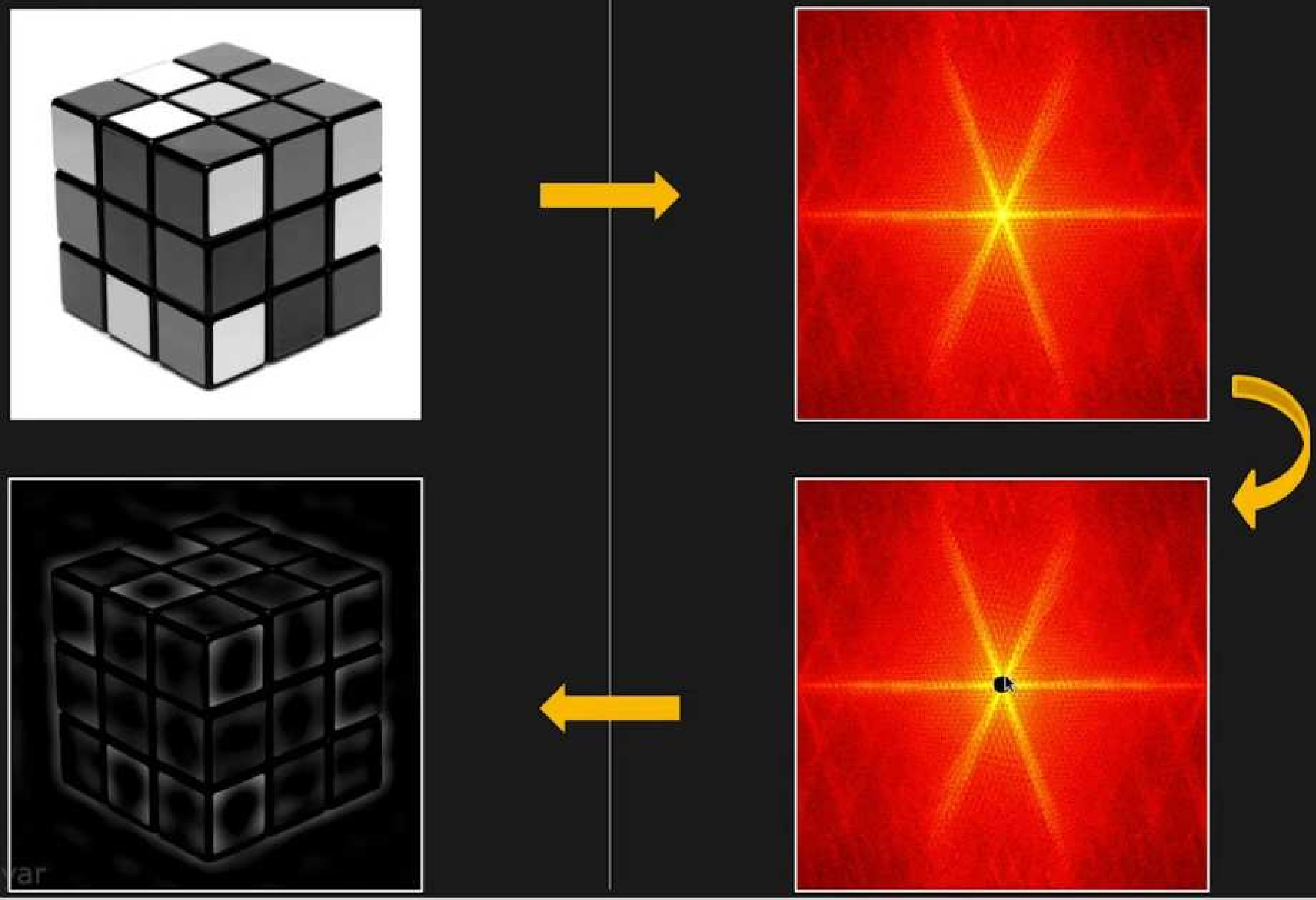




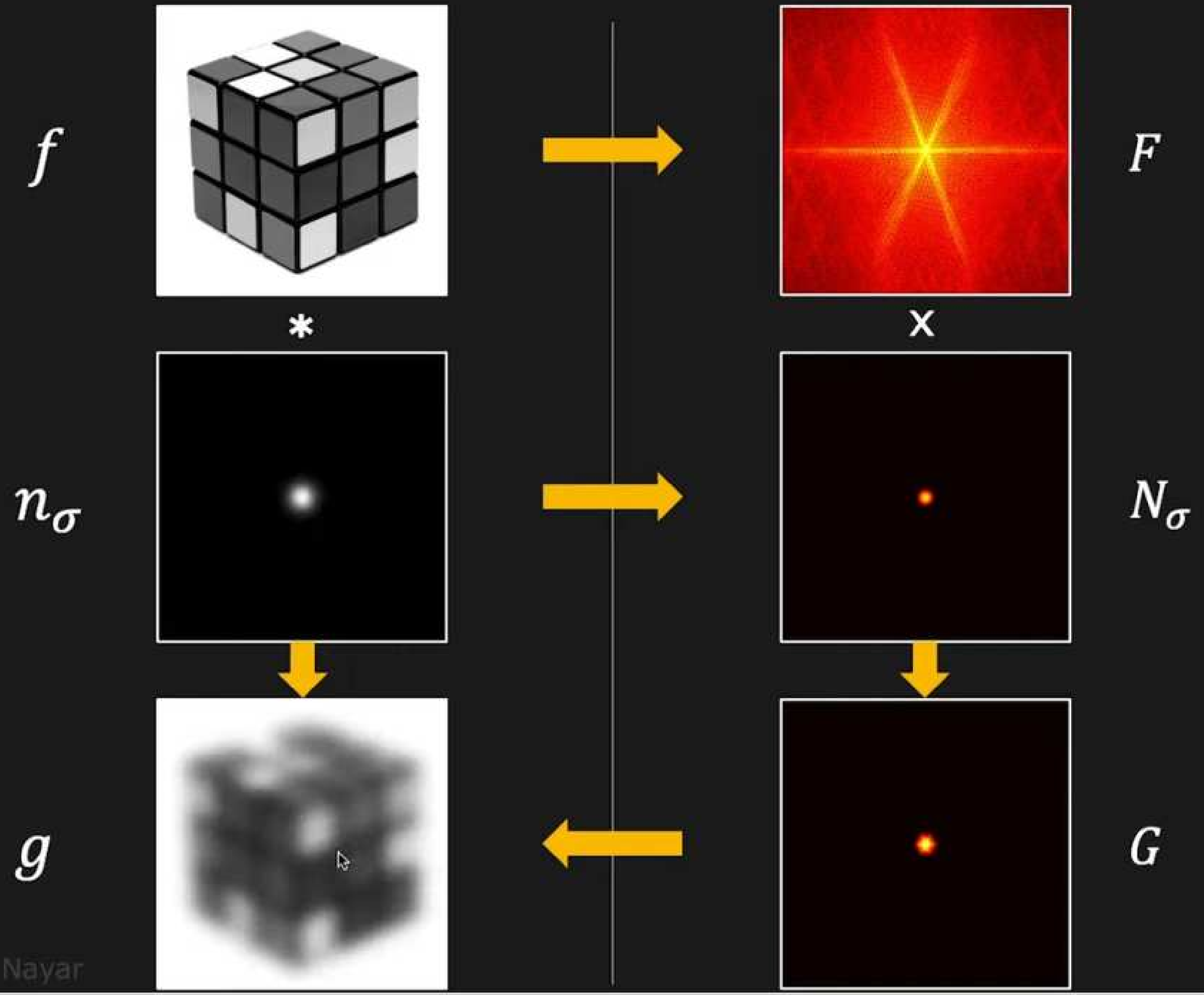


High Pass Filtering

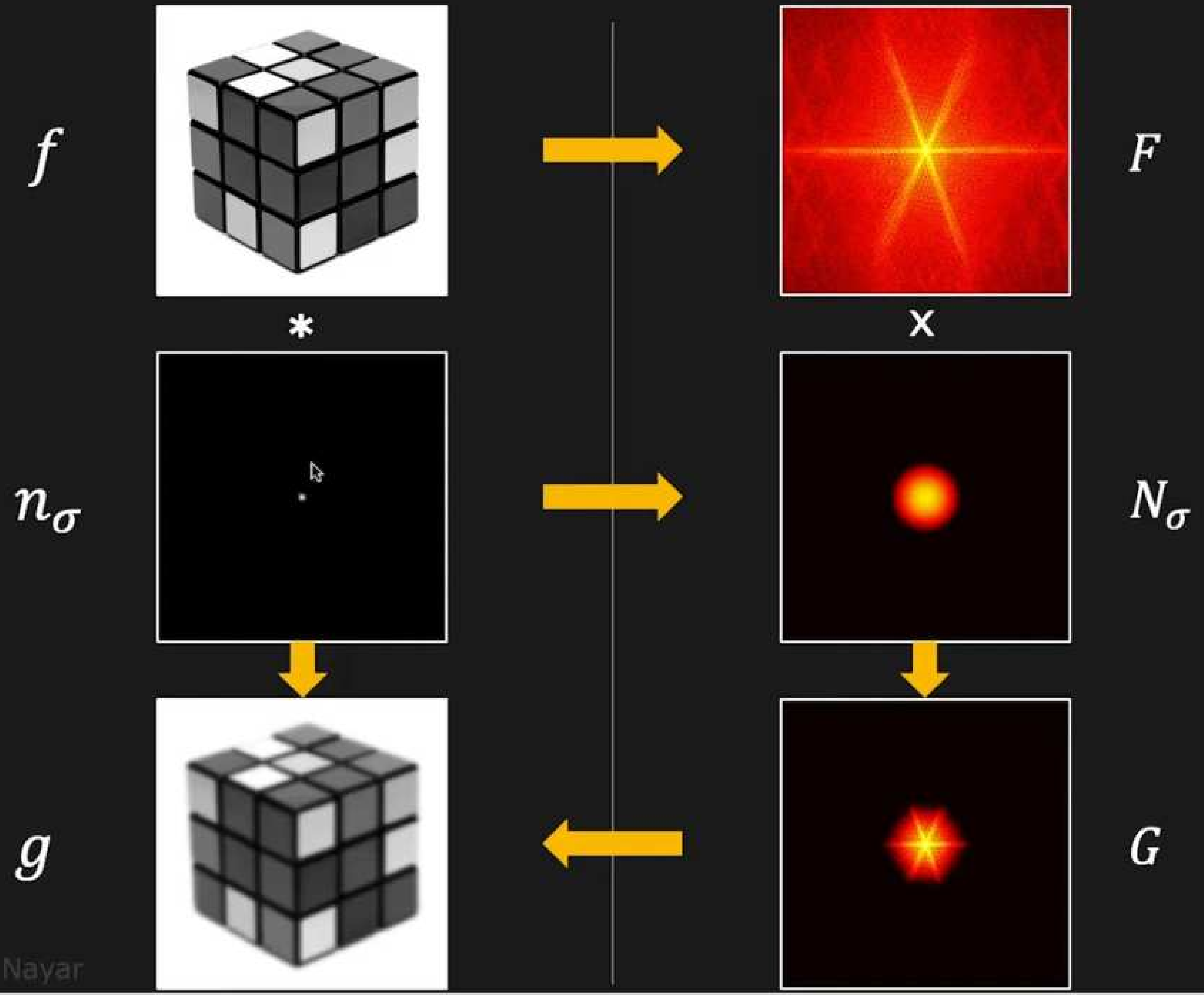
$f(m,n)$ $\log(|F(p,q)|)$



Gaussian Smoothing

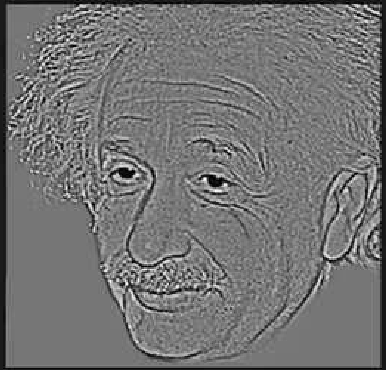


Gaussian Smoothing



Hybrid Images

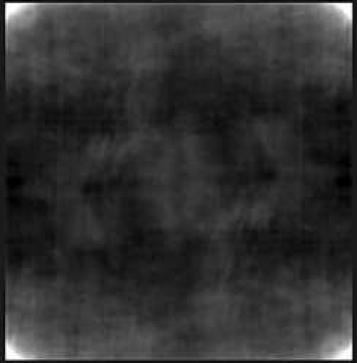
Low Freq Only



High Freq Only

[Oliva 2

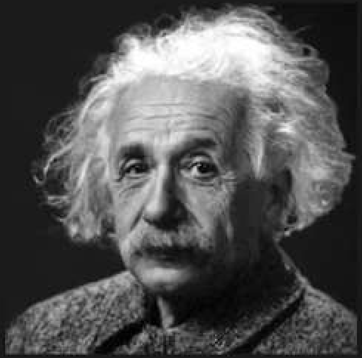
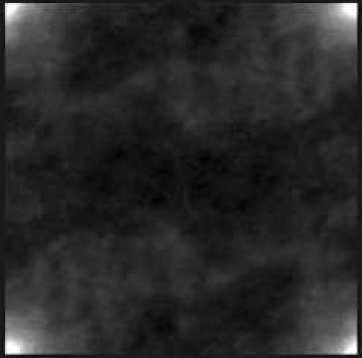
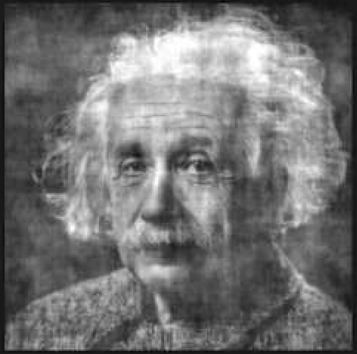
Importance of Phase


Original Image

Magnitude Preserved, Phase Set to Zero

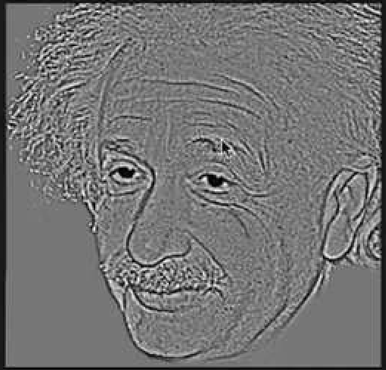
Phase Preserved, Magnitude Set to Average of Natural Images

[Oppenheim

Hybrid Images

Low Freq Only

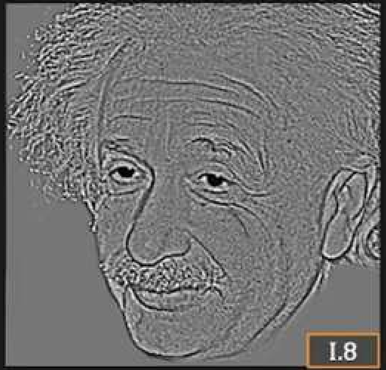


High Freq Only

[Oliva 20

Hybrid Images

Low Freq Only



High Freq Only

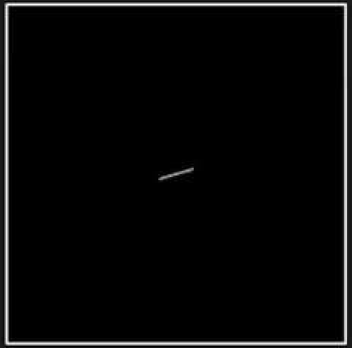
Hybrid (Sum) Image

[Oliva 20

Motion Blur

Scene $f(x, y)$

$*$



PSF $h(x, y)$
(Camera Shake)

$=$

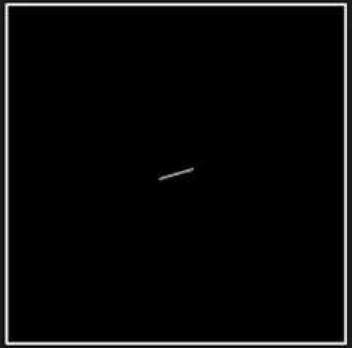
Image $g(x, y)$

$$f(x, y) * h(x, y) = g(x, y)$$

Given captured image $g(x, y)$ and PSF $h(x, y)$,
can we estimate actual scene $f(x, y)$?

Motion Deblur: Deconvolution

*



=

Scene $f(x, y)$ PSF $h(x, y)$ (Camera Shake) Image $g(x, y)$

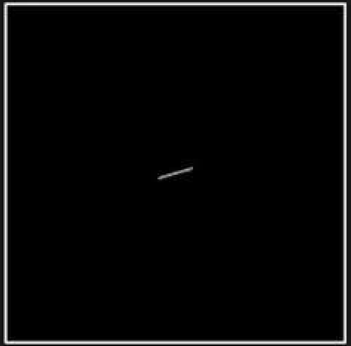
Let f' be the recovered scene.

$$f'(x, y) * h(x, y) = g(x, y)$$

Motion Deblur: Deconvolution

Scene $f(x,y)$

$*$



PSF $h(x,y)$
(Camera Shake)

$=$

Image $g(x,y)$

Let f' be the recovered scene.

$$f'(x,y) * h(x,y) = g(x,y)$$

$$F'(u,v)H(u,v) \quad = G(u,v)$$

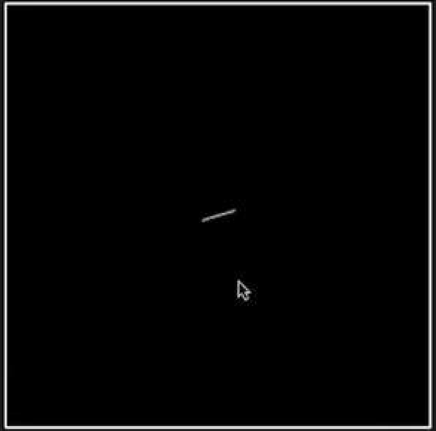
$$F'(u,v) = \frac{G(u,v)}{H(u,v)} \longrightarrow \boxed{\text{IFT}} \longrightarrow f'(x,y)$$

Motion Deblur: Deconvolution

$$F'(u,v) = \frac{G(u,v)}{H(u,v)} \longrightarrow \boxed{\text{IFT}} \longrightarrow f'(x,y)$$

Image $g(x,y)$

deconvolve



PSF $h(x,y)$

=

Recovered $f'(x,y)$

Motion Deblur: Deconvolution

$$F'(u,v) = \frac{G(u,v)}{H(u,v)} \longrightarrow \boxed{\text{IFT}} \longrightarrow f'(x,y)$$

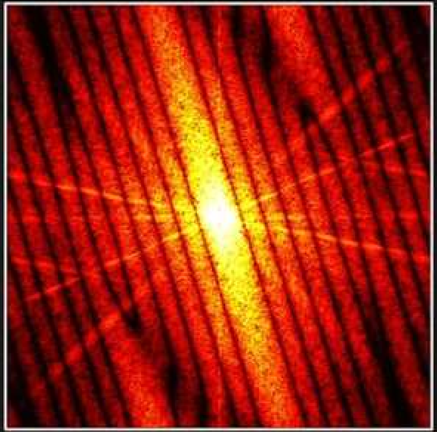
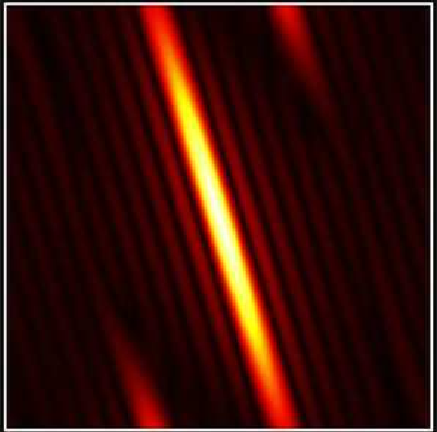


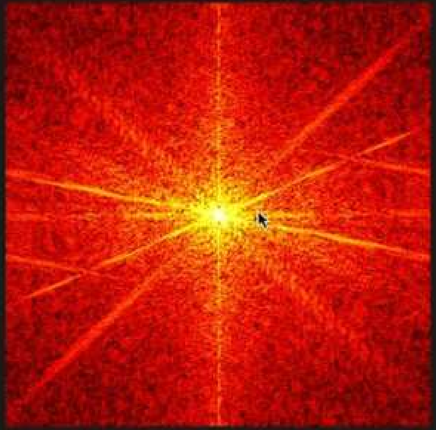
Image $G(u,v)$

/



PSF $H(u,v)$

=



Recovered $F'(u,v)$

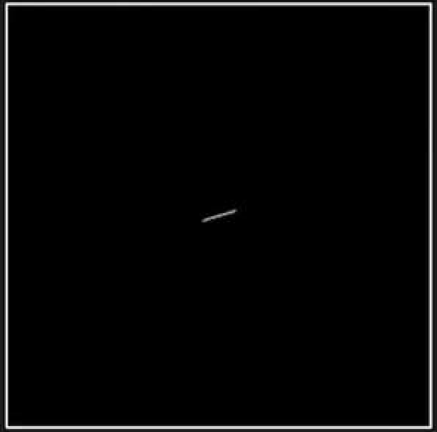
Step 1: Recover $F'(u,v)$ in Fourier Domain

Motion Deblur: Deconvolution

$$F'(u,v) = \frac{G(u,v)}{H(u,v)} \longrightarrow \boxed{\mathrm{IFT}} \longrightarrow f'(x,y)$$

Image $g(x,y)$

deconvolve



PSF $h(x,y)$

=

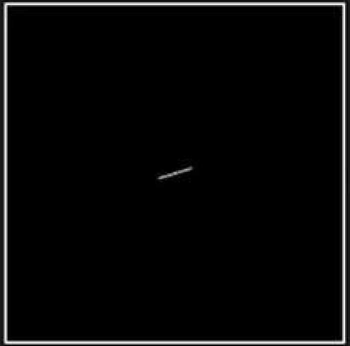
Recovered $f'(x,y)$

Step 2: Compute IFT of $F'(u,v)$ to recover scene

Adding Noise to the Problem

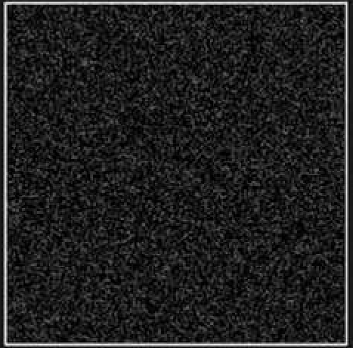
Scene $f(x, y)$

$*$



PSF $h(x, y)$
(Camera Shake)

$+$



Noise $\eta(x, y)$

$=$

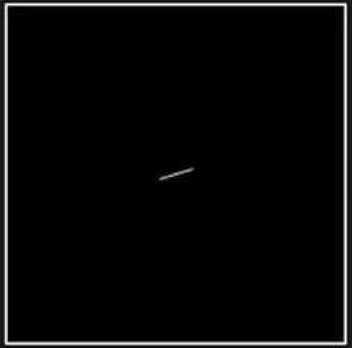
Image $g(x, y)$

$$\boxed{f(x, y) * h(x, y) + \eta(x, y) = g(x, y)}$$

Adding Noise to the Problem

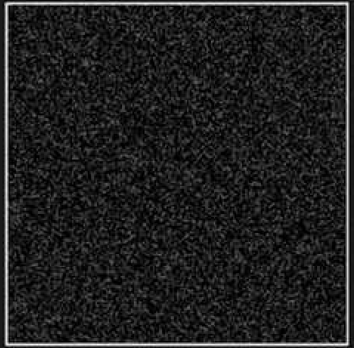
Scene $f(x,y)$

$*$



PSF $h(x,y)$
(Camera Shake)

$+$



Noise $\eta(x,y)$

$=$

Image $g(x,y)$

$$\boxed{f(x,y) * h(x,y) + \eta(x,y) = g(x,y)}$$

Can we afford to ignore noise?

Motion Deblur: Deconvolution

If we ignore the noise ($\eta(x,y)$):

$$\frac{G(u,v)}{H(u,v)} = F'(u,v) \longrightarrow \boxed{\text{IFT}} \longrightarrow f'(x,y)$$

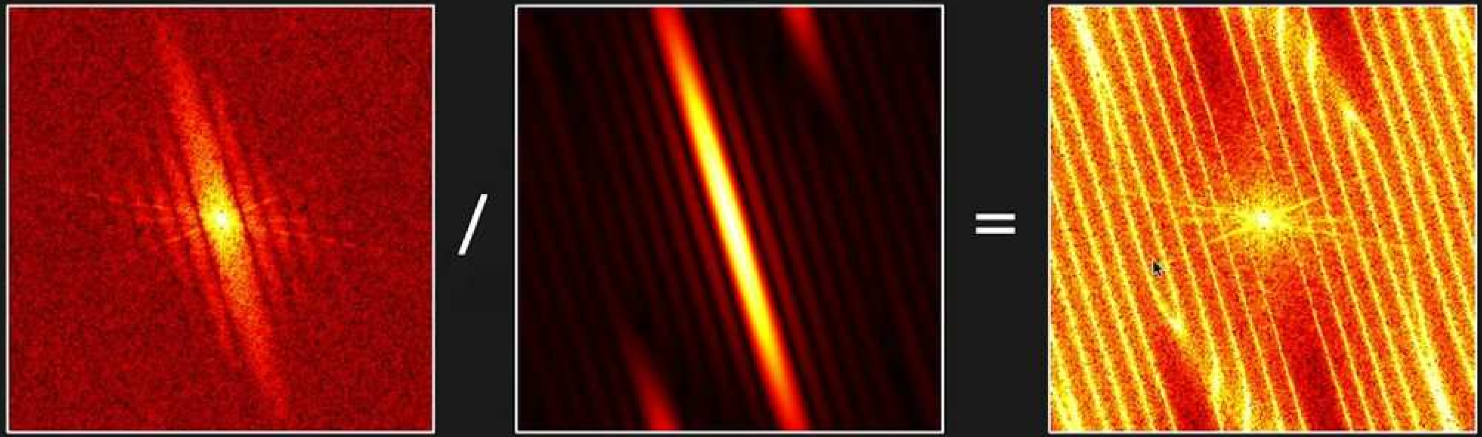


Image $G(u,v)$ / PSF $H(u,v)$ = Recovered $F'(u,v)$

Higher frequencies in $F'(u,v)$ are amplified

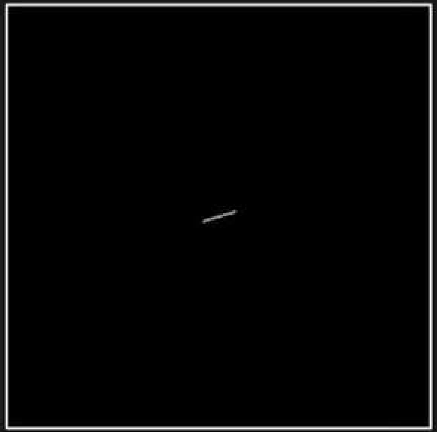
Motion Deblur: Deconvolution

If we ignore the noise ($\eta(x,y)$):

$$\frac{G(u,v)}{H(u,v)} = F'(u,v) \longrightarrow \boxed{\text{IFT}} \longrightarrow f'(x,y)$$

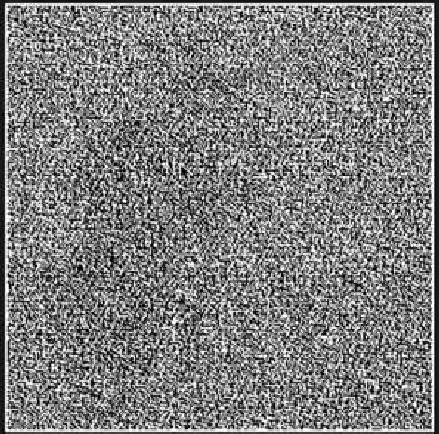
Image $g(x,y)$
(with noise)

deconvolve



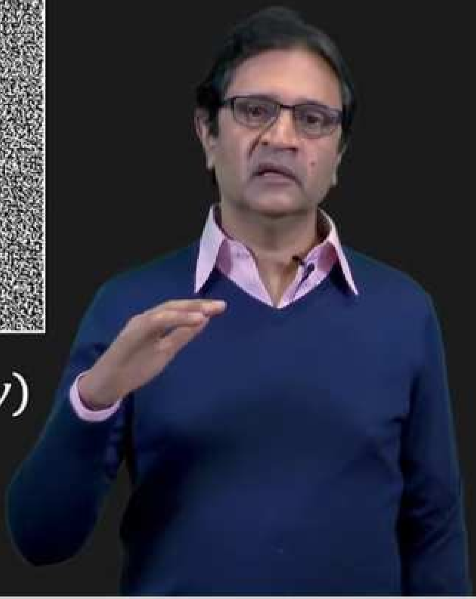
PSF $h(x,y)$

=



Recovered $f'(x,y)$

Noise is significantly amplified



Deconvolution: Issues

$$\frac{G(u,v)}{H(u,v)} = F'(u,v) \longrightarrow \boxed{\text{IFT}} \longrightarrow f'(x,y)$$

1. Where $H(u,v) = 0$, $F'(u,v) = \infty \rightarrow$ Not recoverable

2. Motion blur filter $H(u,v)$ is a low pass filter.

   For high frequencies (u,v):
   - Noise $N(u,v)$ in $G(u,v)$ is high
   - Filter $H(u,v) \approx 0$

Deconvolution: Issues

$$\frac{G(u,v)}{H(u,v)} = F'(u,v) \longrightarrow \boxed{\text{IFT}} \longrightarrow f'(x,y)$$

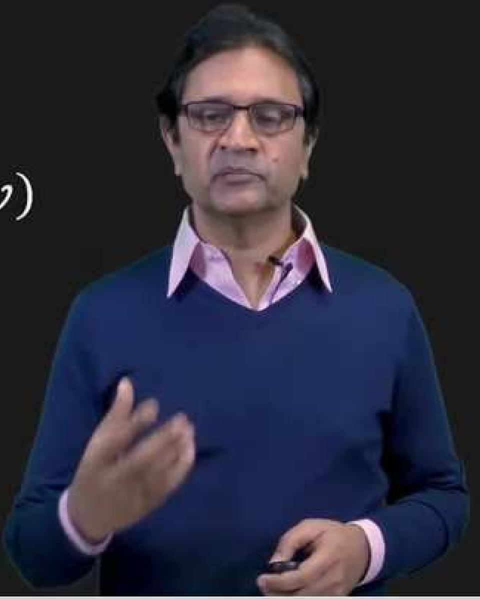
1. Where $H(u,v) = 0$, $F'(u,v) = \infty \rightarrow$ Not recoverable

2. Motion blur filter $H(u,v)$ is a low pass filter.

   For high frequencies (u,v):
   - Noise $N(u,v)$ in $G(u,v)$ is high
   - Filter $H(u,v) \approx 0$

   } Noise in $G(u,v)$ is amplified

We need some kind of Noise Suppression.



Noise Suppression: Weiner Deconvolution

$$\boxed{F'(u,v) = \frac{G(u,v)}{H(u,v)}\left[\frac{1}{1+\dfrac{NSR(u,v)}{|H(u,v)|^2}}\right]}$$

Noise-to-Signal Ratio, $NSR(u,v)$:

$$NSR(u,v) = \frac{\text{Power of Noise at } (u,v)}{\text{Power of Signal (Scene) at } (u,v)} = \frac{|N(u,v)|^2}{|F(u,v)|^2}$$

Noise Suppression: Weiner Deconvolution

$$\boxed{F'(u,v) = \frac{G(u,v)}{H(u,v)} \left[\frac{1}{1 + \frac{NSR(u,v)}{|H(u,v)|^2}} \right]}$$

Where:

Weiner Filter $\stackrel{\text{def}}{=}$ $$\boxed{W(u,v) = \frac{1}{H(u,v)} \left[\frac{1}{1 + \frac{NSR(u,v)}{|H(u,v)|^2}} \right]}$$

Noise-to-Signal Ratio, $NSR(u,v)$:

$$NSR(u,v) = \frac{\text{Power of Noise at } (u,v)}{\text{Power of Signal (Scene) at } (u,v)} = \frac{|N(u,v)|^2}{|F(u,v)|^2}$$

Noise Suppression: Weiner Deconvolution

$$\boxed{F'(u,v) = \frac{G(u,v)}{H(u,v)} \left[\frac{1}{1 + \frac{NSR(u,v)}{|H(u,v)|^2}} \right]}$$

- Determining NSR requires us to have prior knowledge of the noise "pattern" and the scene (or of a similar scene).

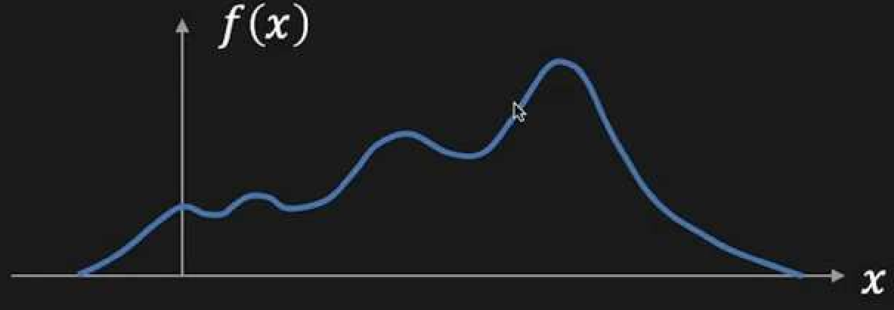
$$NSR(u,v) = \frac{|N(u,v)|^2}{|F(u,v)|^2}$$

- Often NSR is set to a single suitable constant λ.

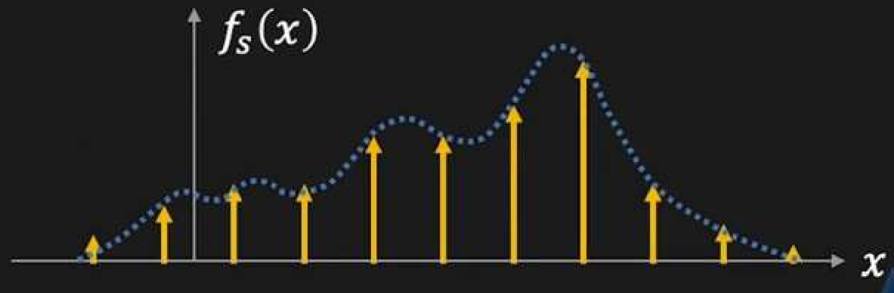
$$\boxed{NSR(u,v) = \lambda}$$

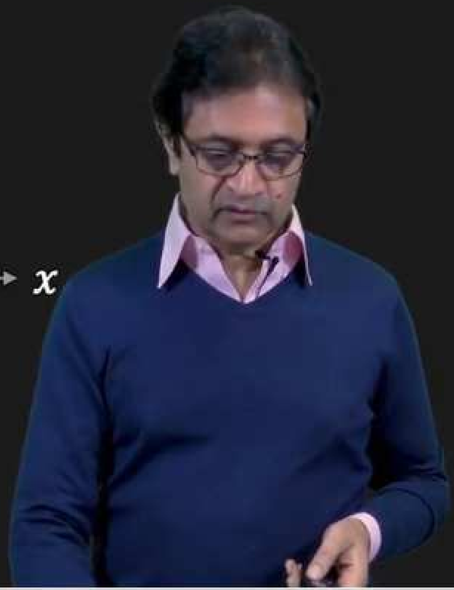
From Continuous to Digital Image

Continuous Signal:



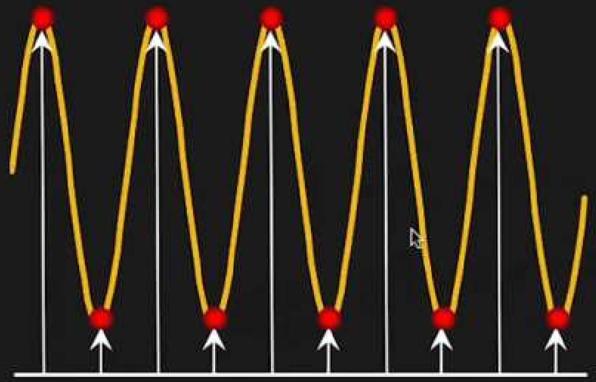
Digital Signal:



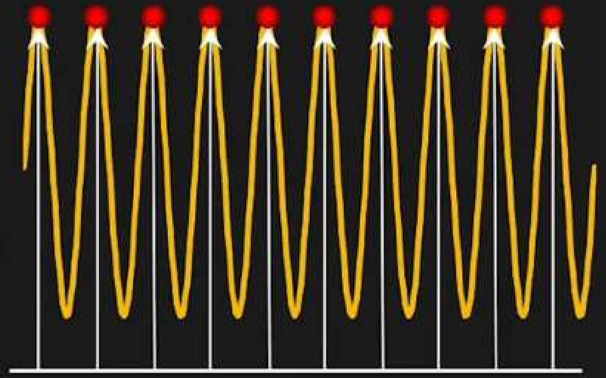


How "dense" should the samples be?

Sampling Problem

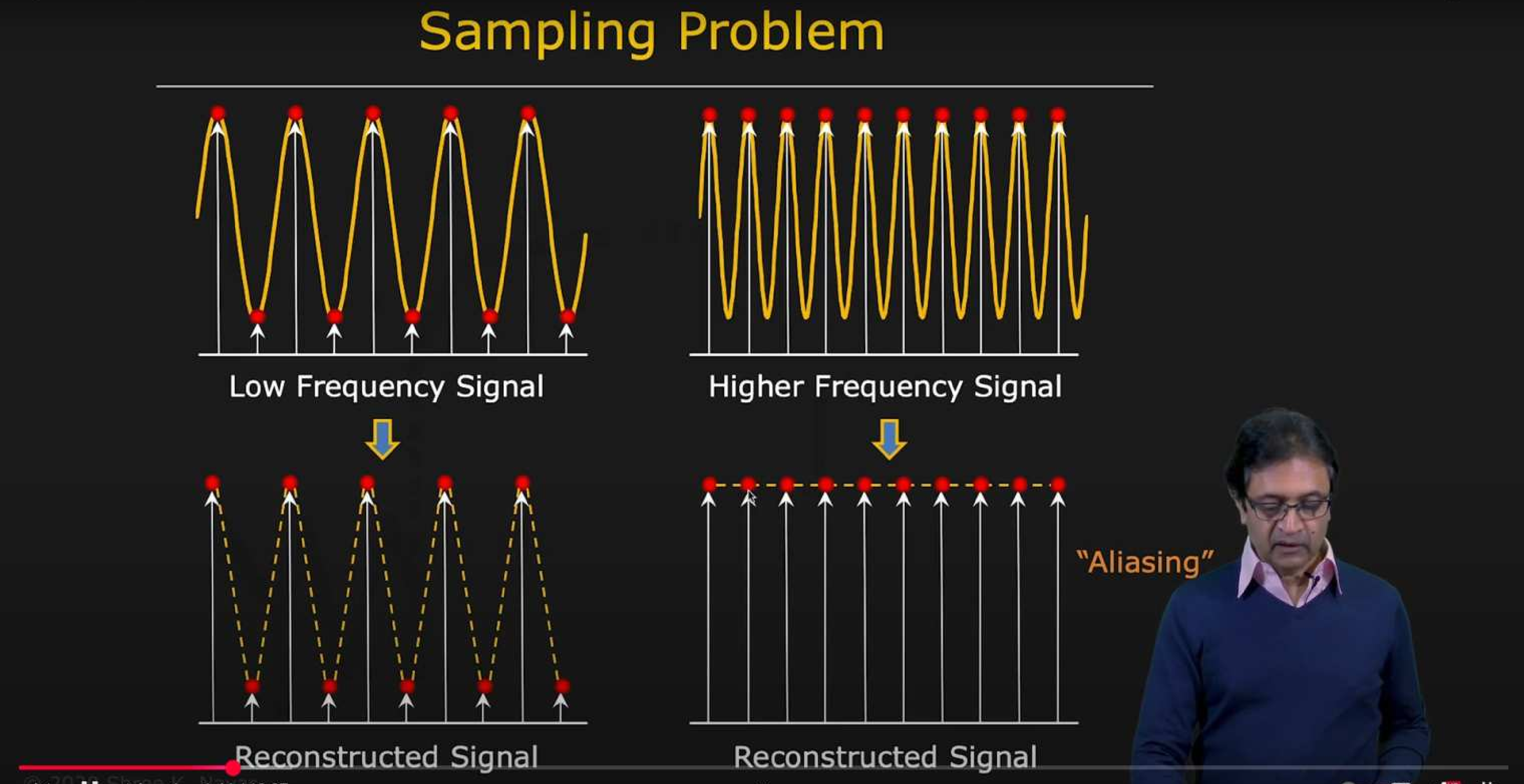


Low Frequency Signal



Higher Frequency Signal

Sampling Theory and Aliasing | Image Processing II



1:44 / 12:07

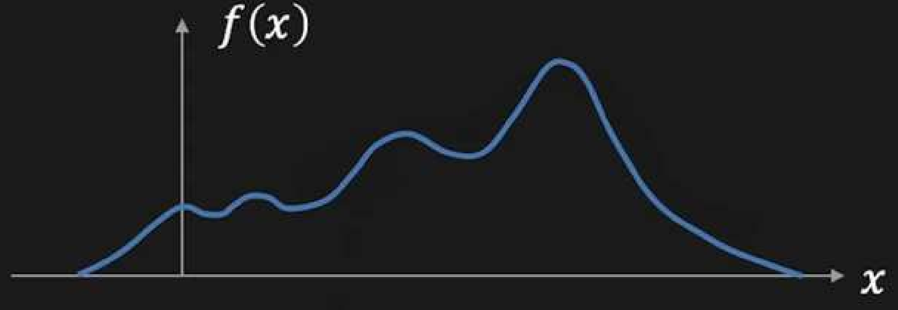
Sampling Problem

"Well sampled" image

"Under sampled" image
(visible aliasing artifacts)

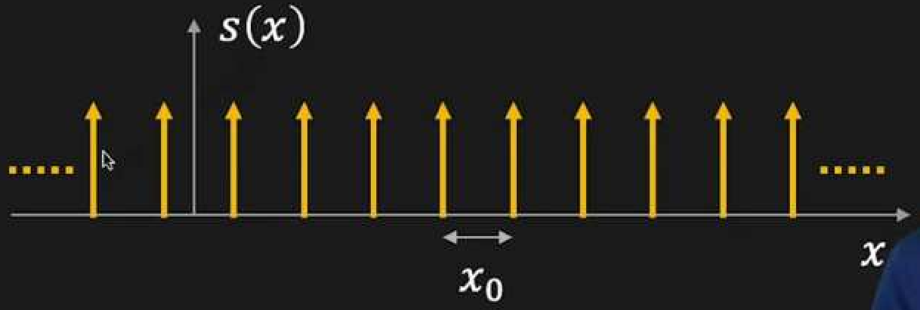
Sampling Theory

Continuous Signal:

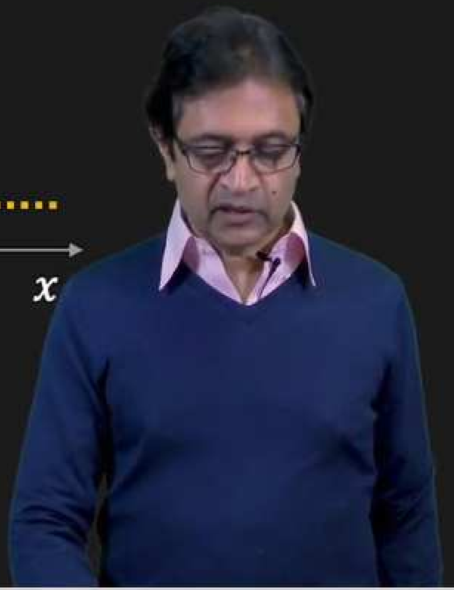


Shah Function (Impulse Train):

$$s(x) = \sum_{n=-\infty}^{\infty} \delta(x - nx_0)$$



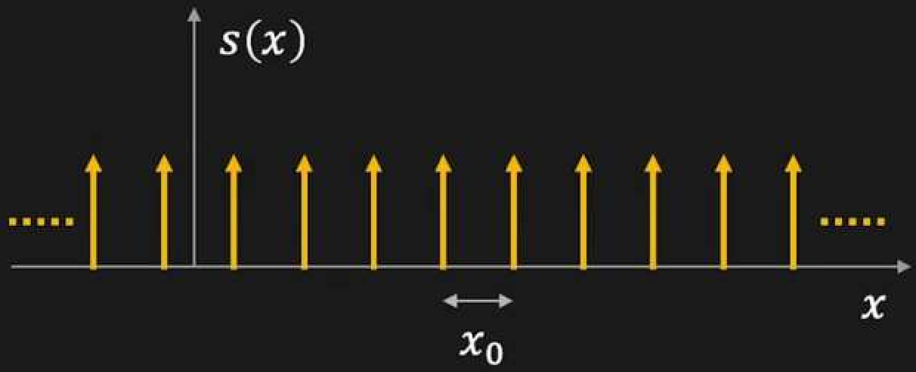
Sampled Function: $\boxed{f_s(x) = f(x)s(x)}$



Shah Function (Impulse Train)

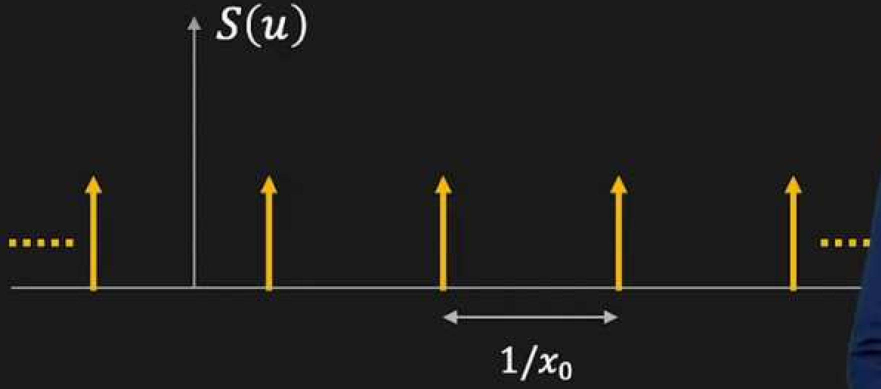
Shah Function (Spatial Domain):

$$s(x) = \sum_{n=-\infty}^{\infty} \delta(x - n x_0)$$



Shah Function (Fourier Domain):

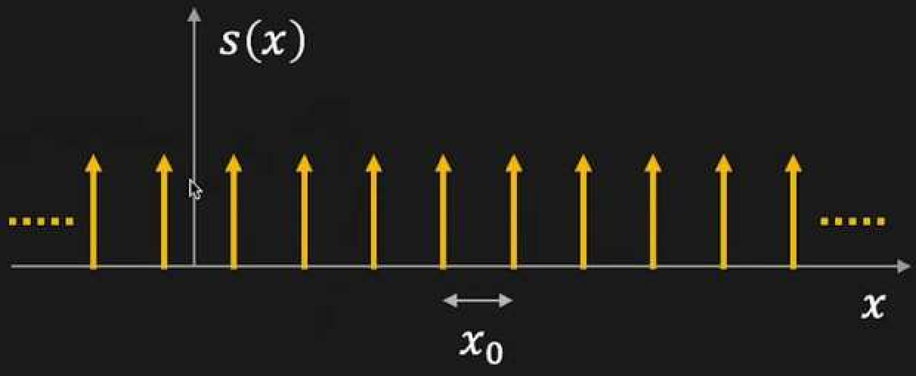
$$S(u) = \frac{1}{x_0} \sum_{n=-\infty}^{\infty} \delta\left(x - \frac{n}{x_0}\right)$$



Shah Function (Impulse Train)

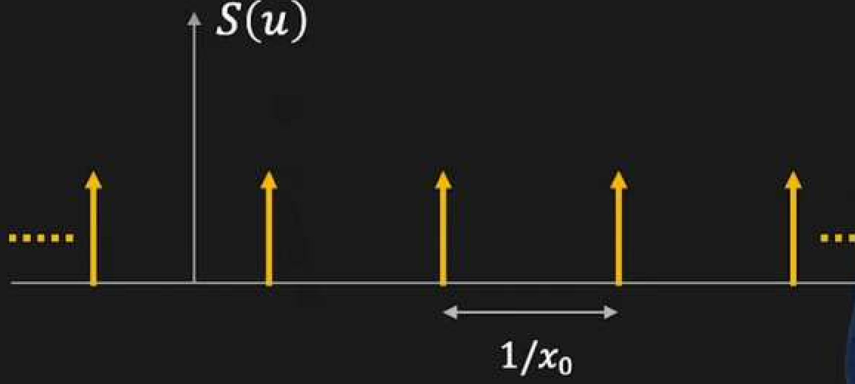
Shah Function (Spatial Domain):

$$s(x) = \sum_{n=-\infty}^{\infty} \delta(x - nx_0)$$



Shah Function (Fourier Domain):

$$S(u) = \frac{1}{x_0} \sum_{n=-\infty}^{\infty} \delta\left(x - \frac{n}{x_0}\right)$$



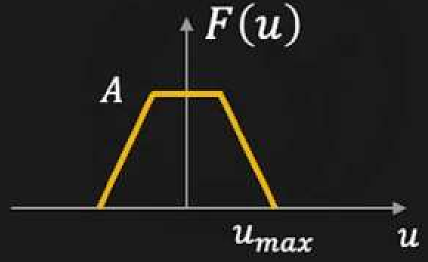
Fourier Analysis of Sampled Signal

Sampled Signal:

$$f_s(x) = f(x)s(x) = f(x)\sum \delta(x - nx_0)$$

$$F_s(u) = F(u) * S(u) = F(u) * \frac{1}{x_0}\sum \delta(u - n/x_0)$$

For example:



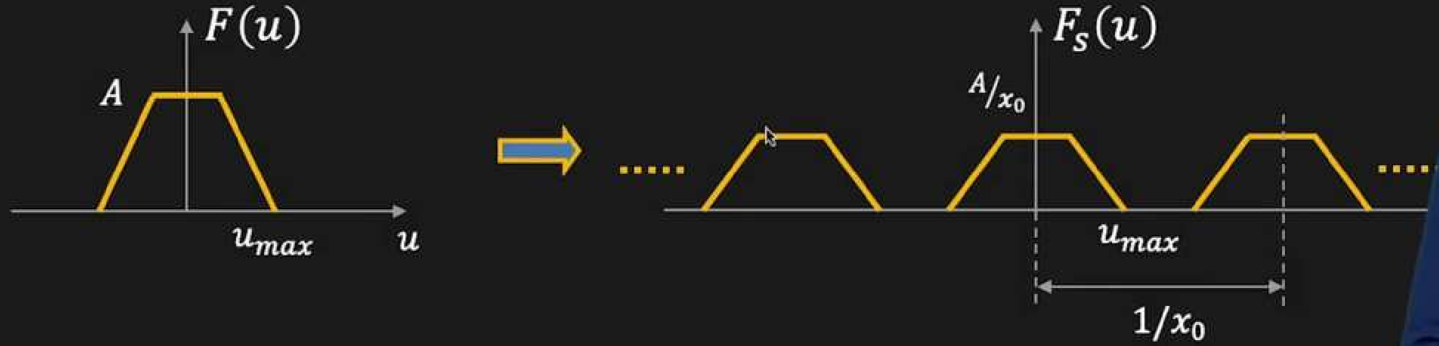
Fourier Analysis of Sampled Signal

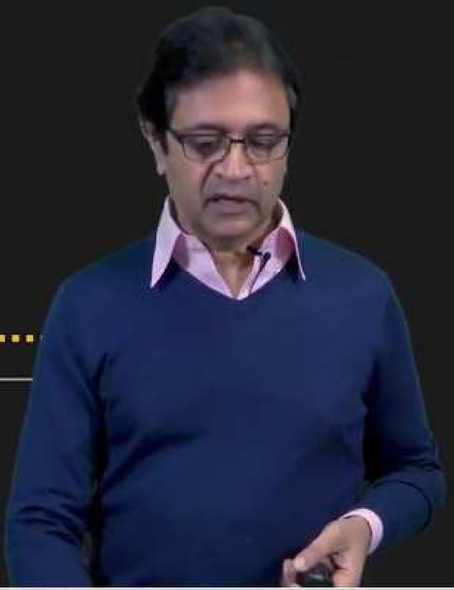
Sampled Signal:

$$f_s(x) = f(x)s(x) = f(x)\sum \delta(x - nx_0)$$

$$F_s(u) = F(u) * S(u) = F(u) * \frac{1}{x_0}\sum \delta(u - n/x_0)$$

If $u_{max} \leq \frac{1}{2x_0}$





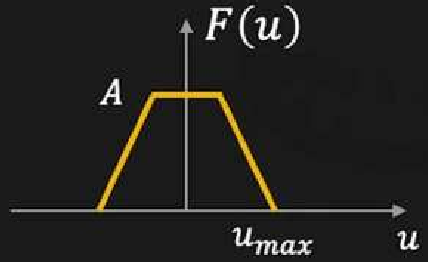
Aliasing

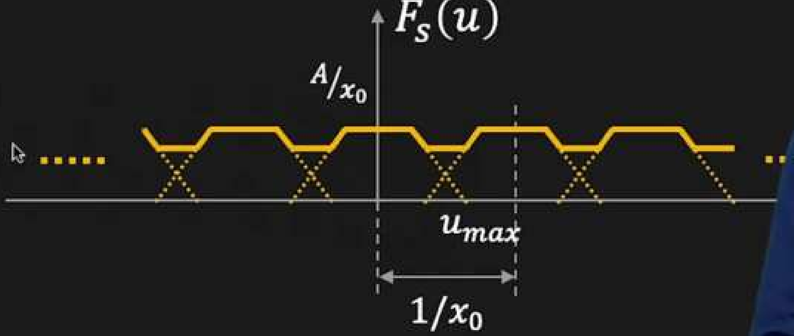
Sampled Signal:

$$f_s(x) = f(x)s(x) = f(x)\sum \delta(x - nx_0)$$

$$F_s(u) = F(u) * S(u) = F(u) * \frac{1}{x_0}\sum \delta(u - n/x_0)$$

If $u_{max} > \frac{1}{2x_0}$ Aliasing

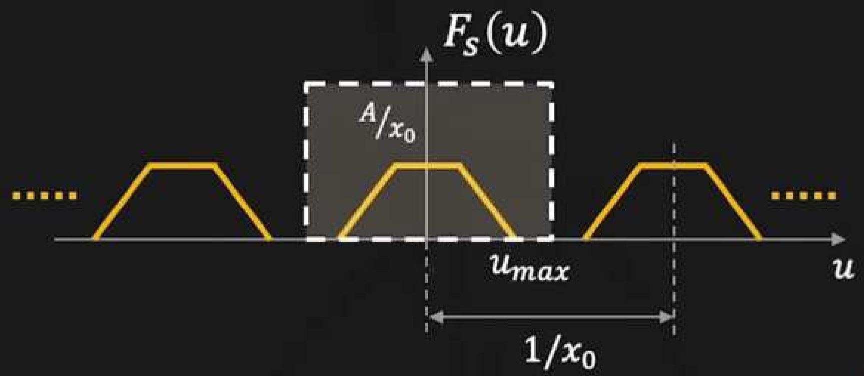




Nyquist Theorem

Can we recover $f(x)$ from $f_s(x)$? In other words, can we recover $F(u)$ from $F_s(u)$?

Only if $u_{max} \leq \frac{1}{2x_0}$ (Nyquist Frequency)



$$C(u) = \begin{cases} x_0, & |u| < 1/2x_0 \\ 0, & Otherwise \end{cases}$$

Nyquist Theorem

Can we recover $f(x)$ from $f_s(x)$? In other words, can we recover $F(u)$ from $F_s(u)$?

Only if $u_{max} \leq \dfrac{1}{2x_0}$ (Nyquist Frequency)

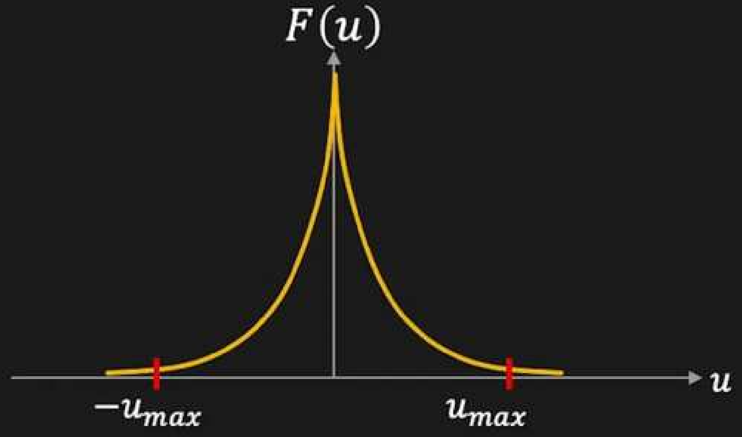
$$F(u) = F_s(u)C(u)$$

$$C(u) = \begin{cases} x_0, & |u| < 1/2x_0 \\ 0, & Otherwise \end{cases}$$

$$\boxed{f(x) = IFT\big(F(u)\big)}$$

Aliasing in Digital Imaging

Aliasing occurs when imaging a scene (signal) that has frequencies above the image sensor's Nyquist Frequency

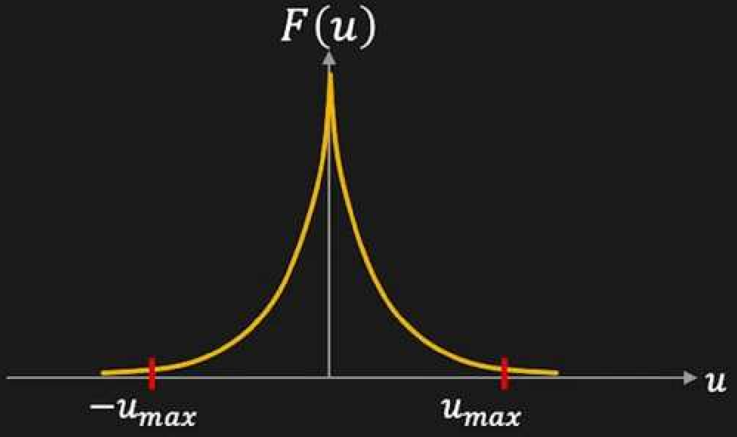


Typical Power Spectrum of Natural Scenes

Aliasing artifacts usually occur
the form of Moiré patterns

Aliasing in Digital Imaging

Aliasing occurs when imaging a scene (signal) that has frequencies above the image sensor's Nyquist Frequency



Typical Power Spectrum of Natural Scenes

Aliasing artifacts usually occur in the form of Moiré patterns

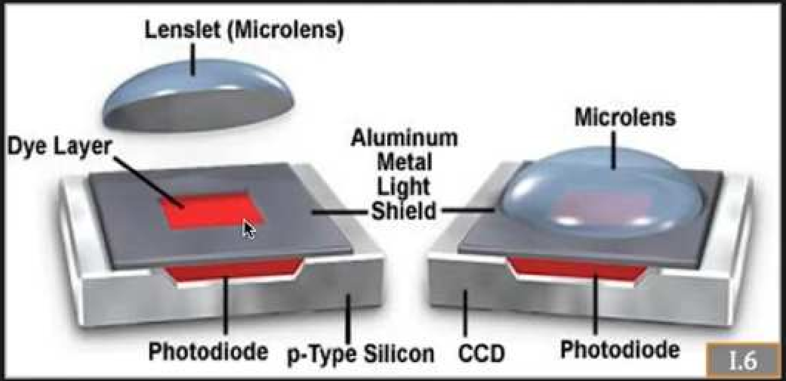
How do sensors combat aliasing?

Minimizing the Effects of Aliasing

Band Limit: Clip the signal above the Nyquist frequency.

Effectively, "blur" the scene before sampling.

Sensors use two strategies.



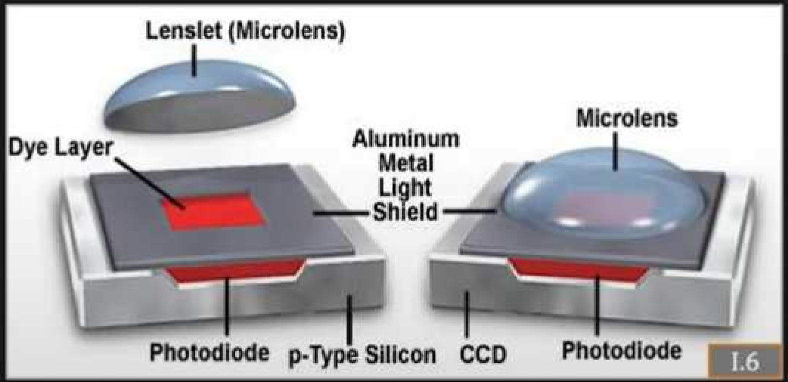
Pixels are area-samplers
(box-averaging filter)

Minimizing the Effects of Aliasing

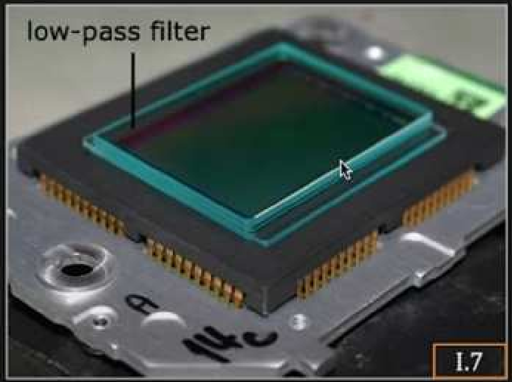
Band Limit: Clip the signal above the Nyquist frequency.

Effectively, "blur" the scene before sampling.

Sensors use two strategies.



Pixels are area-samplers
(box-averaging filter)



Use optical low-pass filter
(anti-aliasing filter)