Computer Vision

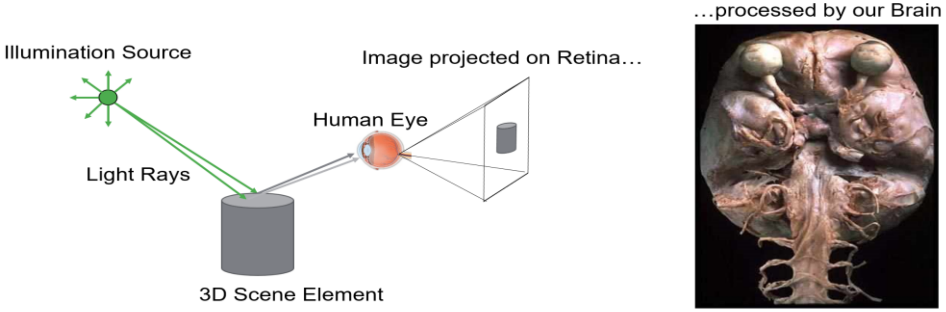
Lecture 2: Capturing Digital Images

Oliver Bimber

JOHANNES KEPLER
UNIVERSITY LINZ

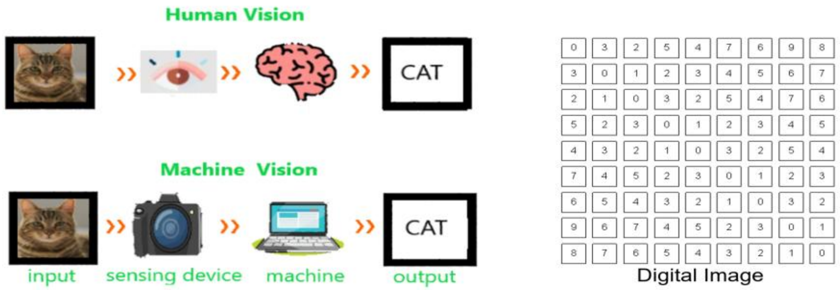
Last Week: Introduction and Course Overview

How Humans see the World (in Principle)



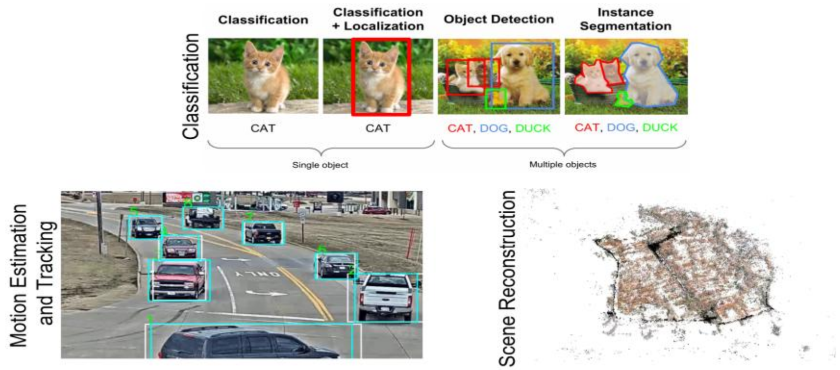
JOHANNES KEPLER UNIVERSITY LINZ

How Computers see the World (in Principle)



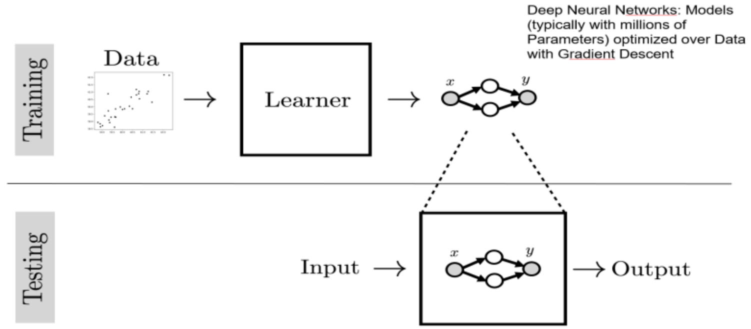
JOHANNES KEPLER UNIVERSITY LINZ

Classical Computer Vision Tasks



JOHANNES KEPLER UNIVERSITY LINZ

Learning from Data

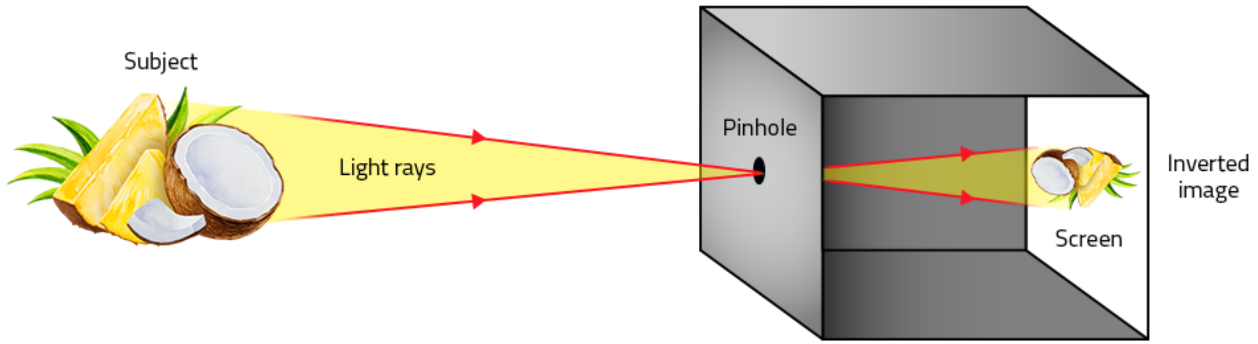


JOHANNES KEPLER UNIVERSITY LINZ

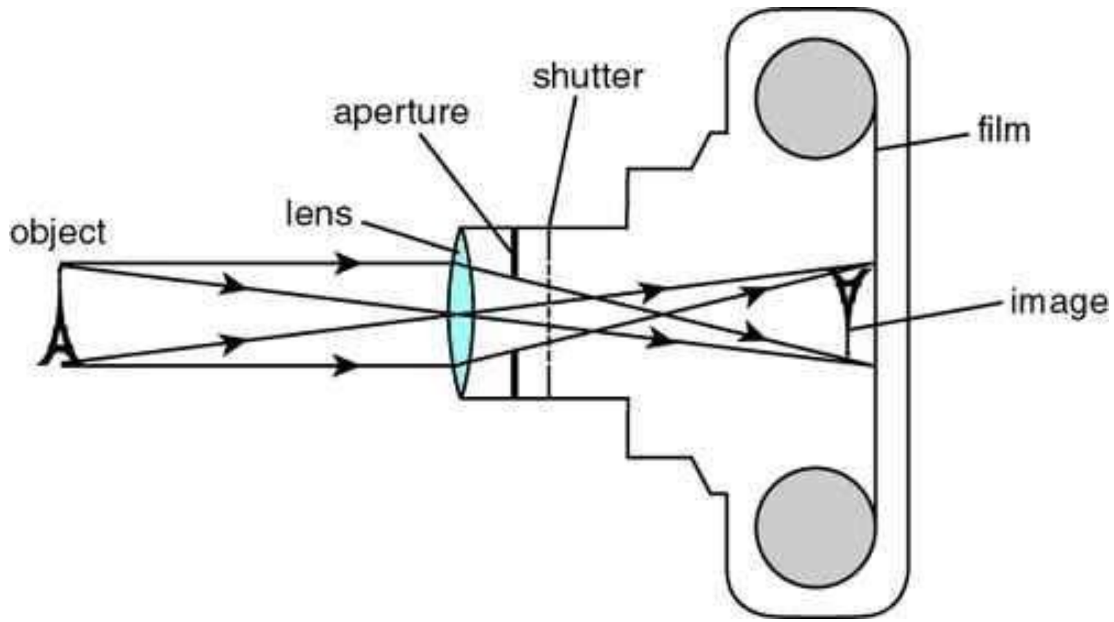
Course Overview

CW	Topic	Date	Place	Lab
41	Introduction and Course Overview	07.10.2025	Zoom	Lab 1
42	Capturing Digital Images	14.10.2025	Zoom	Lab 2
43	Digital Image Processing	21.10.2025	Zoom	Assignment 1
44	Machine Learning	28.10.2025	Zoom	
45	Feature Extraction	04.11.2025	Zoom	Open Lab 1
46	Segmentation	11.11.2025	Zoom	Assignment 2
47	Optical Flow	18.11.2025	Zoom	Open Lab 2
48	Object Detection	25.11.2025	Zoom	Assignment 3
49	Multi-View Geometry	02.12.2025	Zoom	Open Lab 3
50	3D Vision	09.12.2025	Zoom	Assignment 4
3	Trends in Computer Vision	13.01.2026	Zoom	
4	Q&A	20.01.2026	Zoom	Open Lab 4
5	Exam	27.01.2026	HS1 (Linz), S1/S3 (Vienna), S5 (Bregenz)	
9	Retry Exam	24.02.2026	tba	

2D Imaging

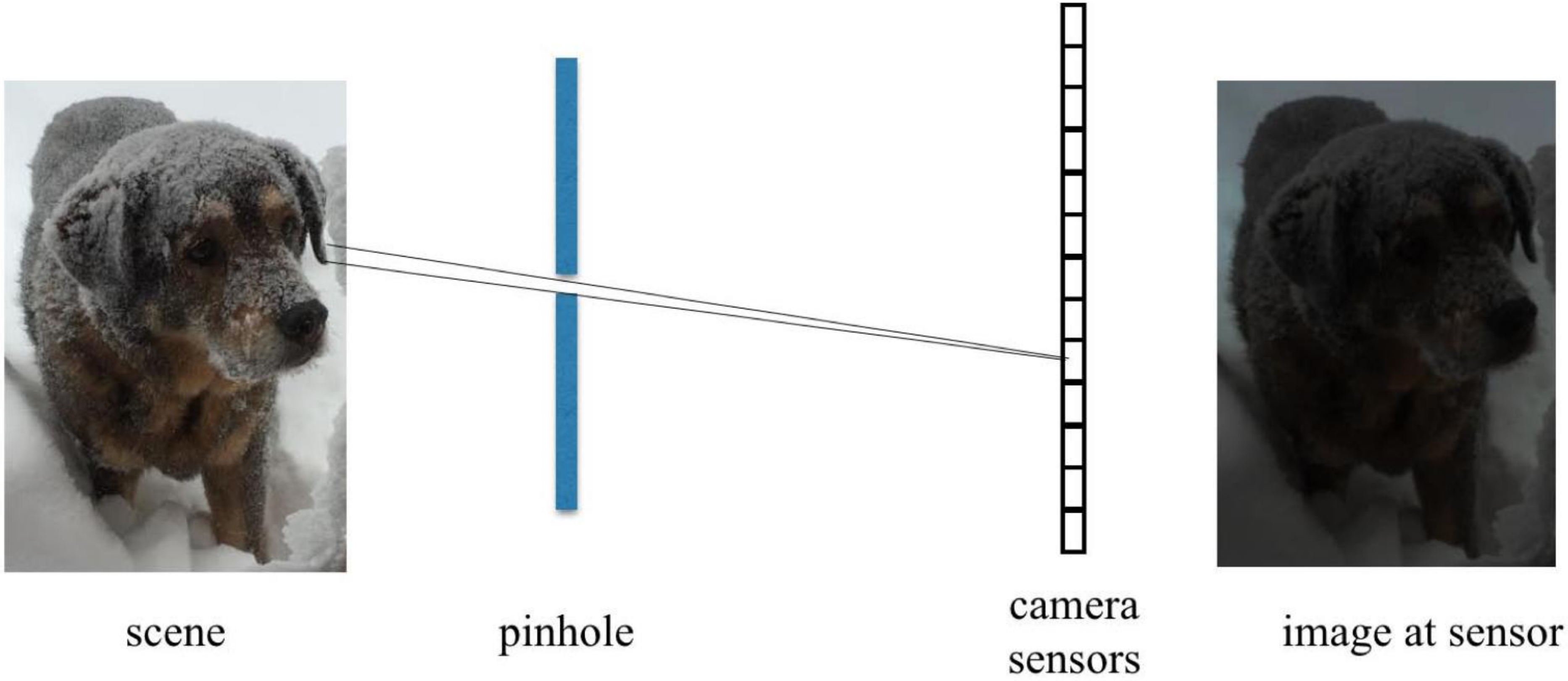


Pinhole Camera (Camera Obscure)

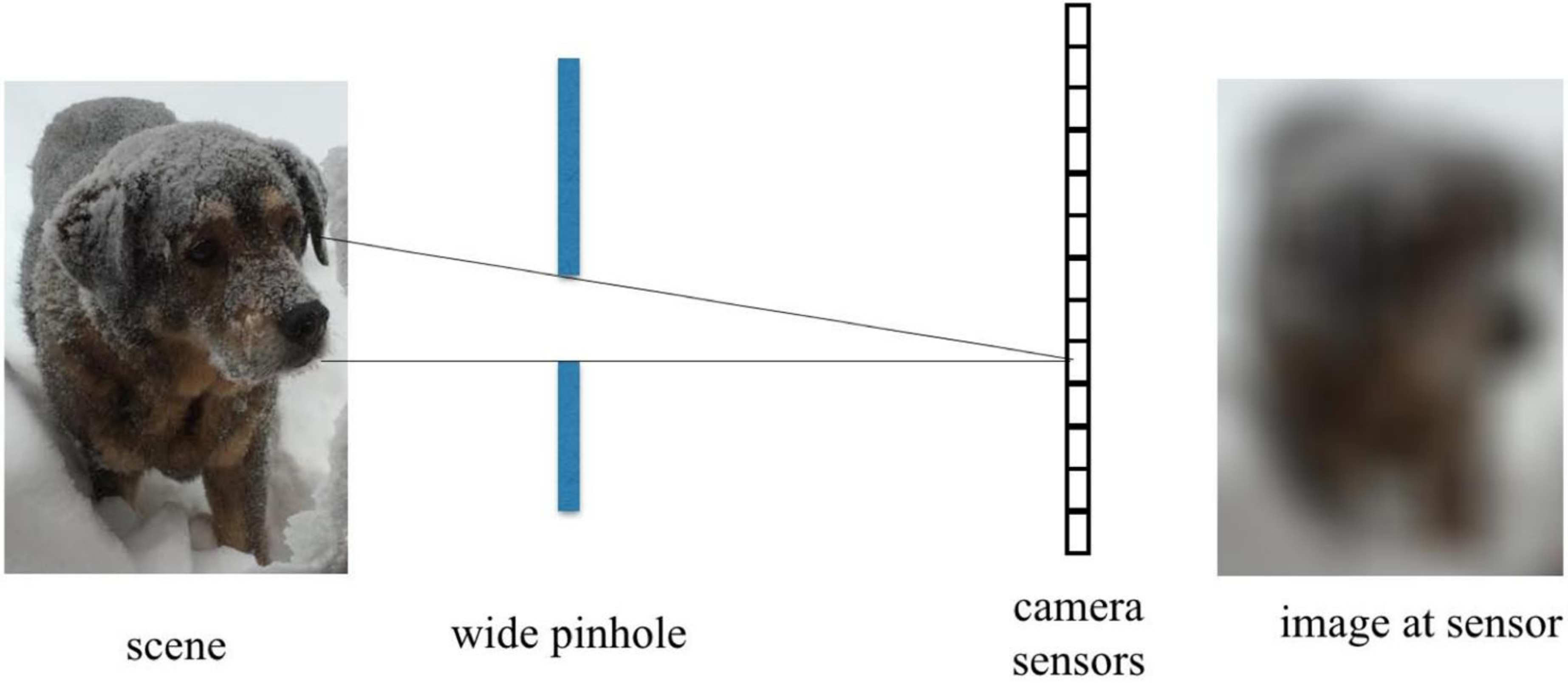


Analog Camera

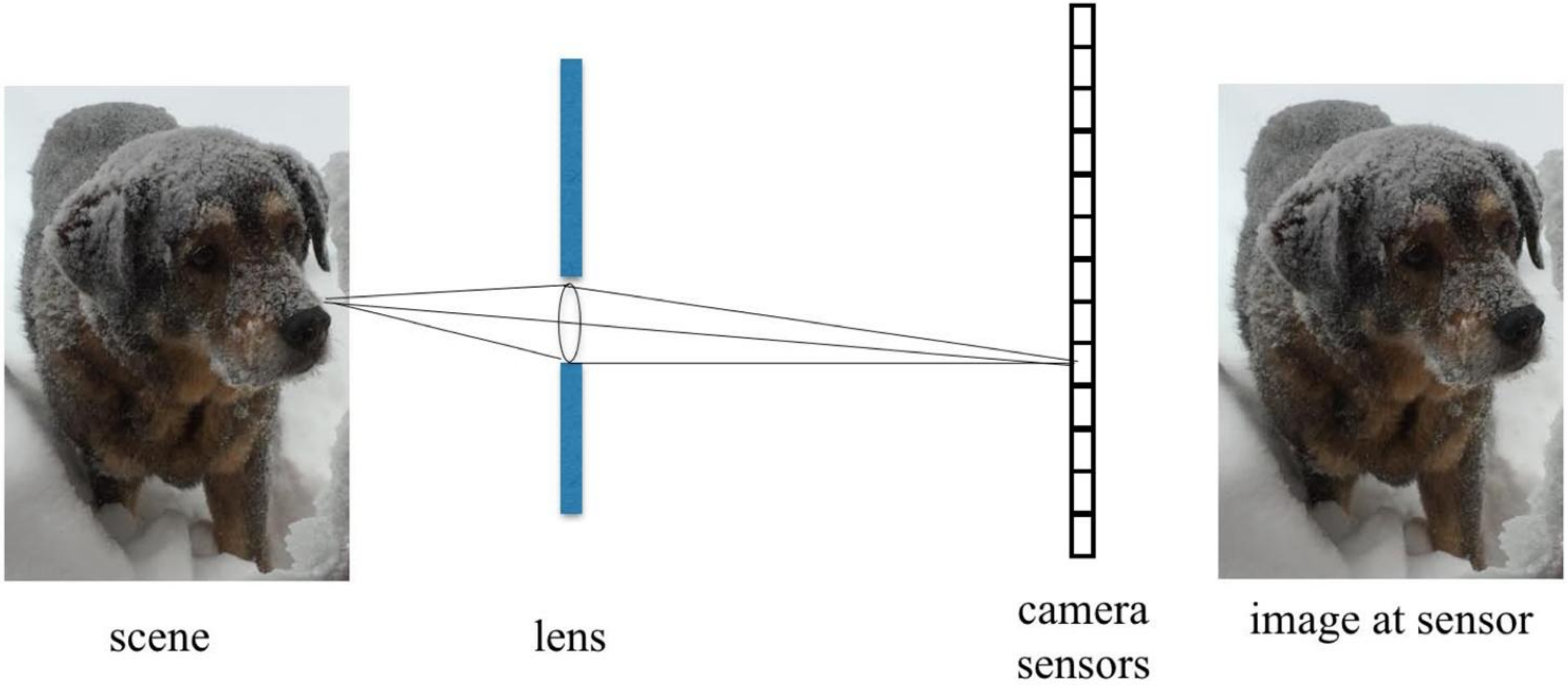
Drawbacks of Pinhole Cameras



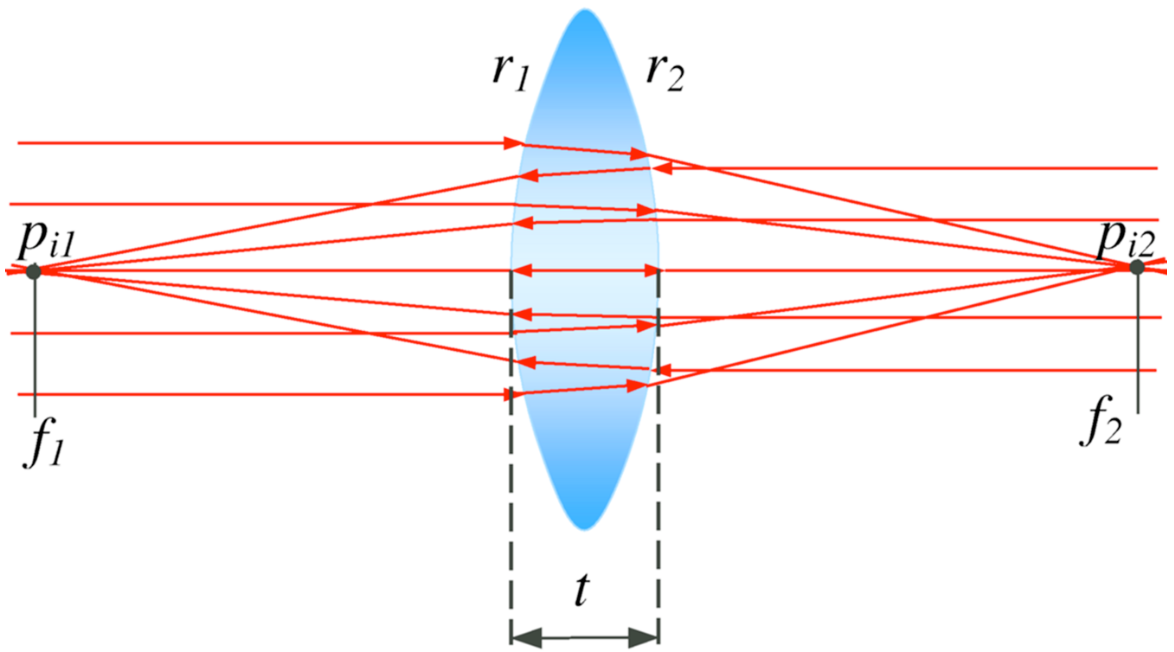
Drawbacks of Pinhole Cameras



Using Lenses to collect and focus Light



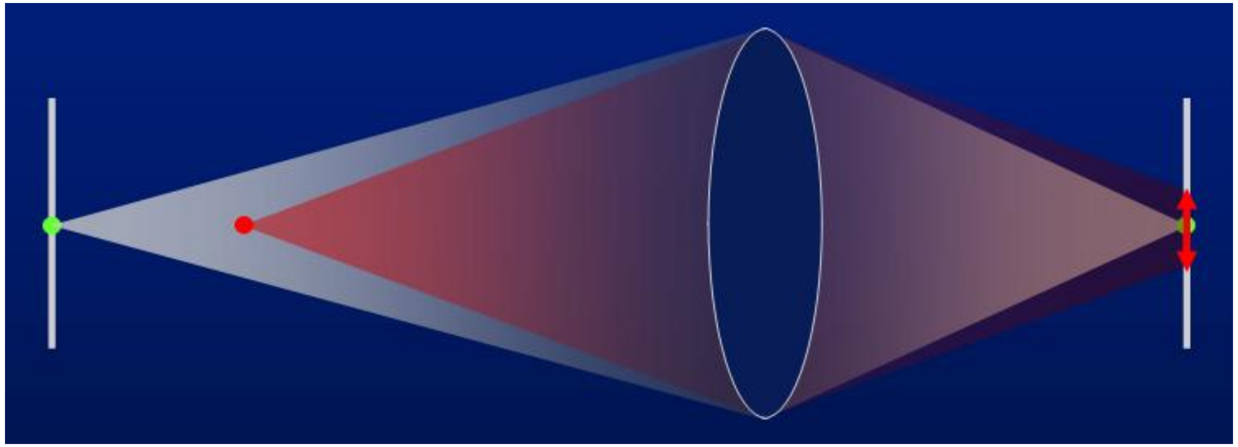
Lenses

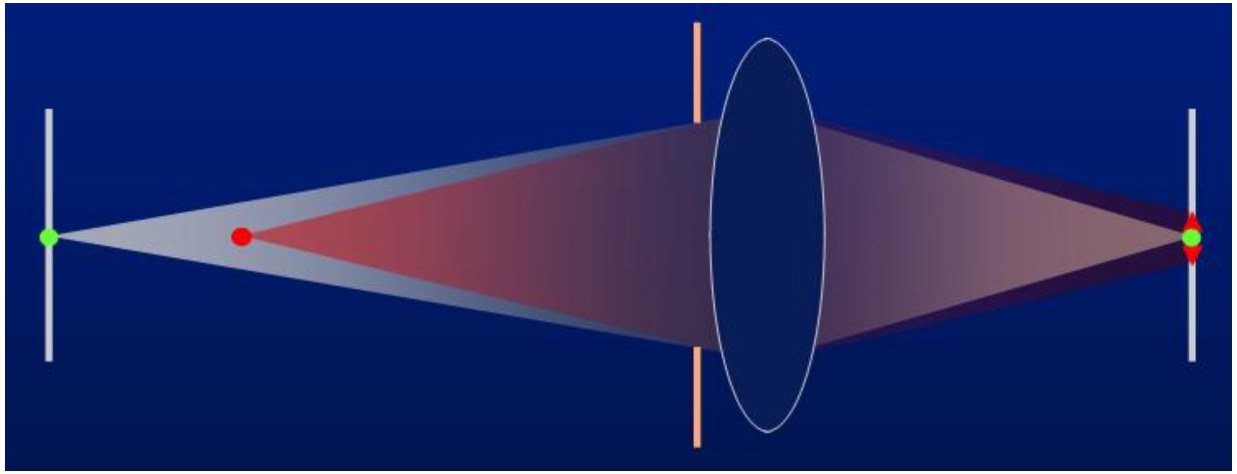


$$\frac{1}{f} = (\eta_2 - 1)\left(\frac{1}{r_1} - \frac{1}{r_2}\right) + \frac{(\eta_2 - 1)^2}{\eta_2}\frac{t}{r_1 r_2}$$

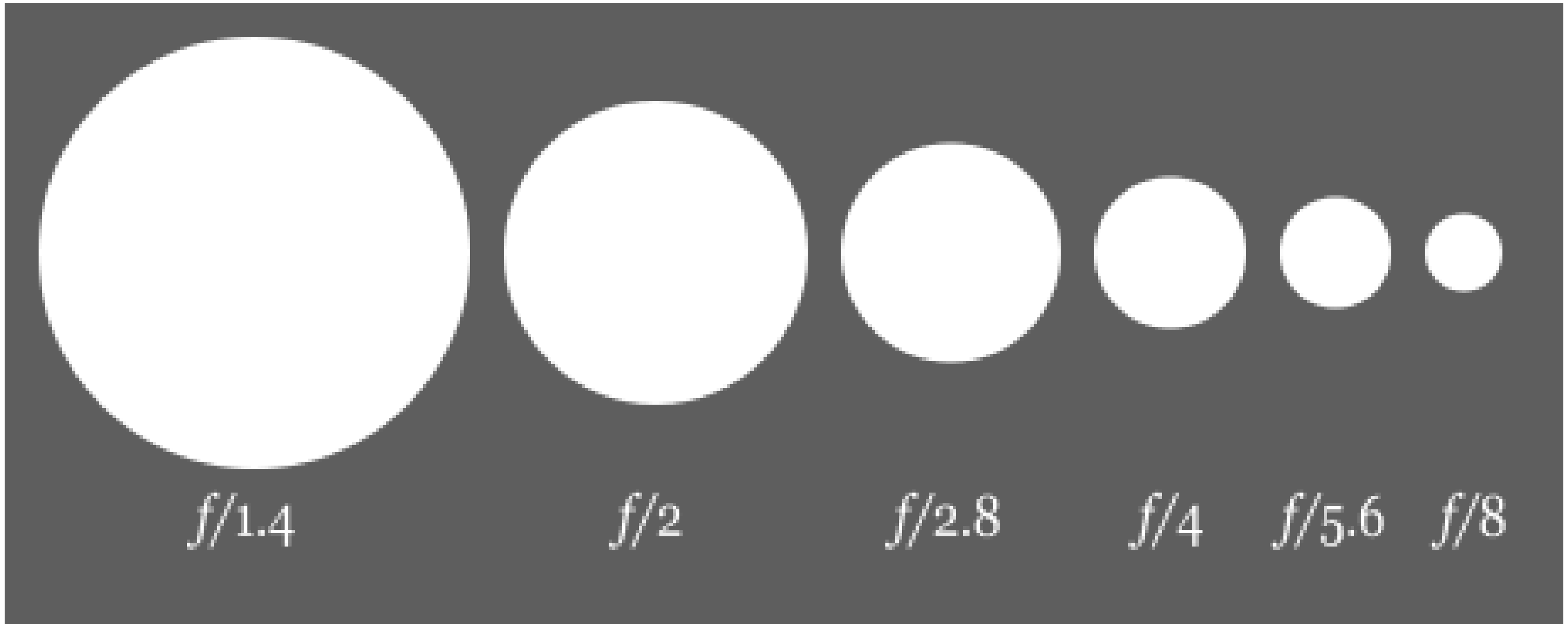
r_1 and r_2 are the two Surface Radii, and t the central Thickness

Depth of Field





f-number

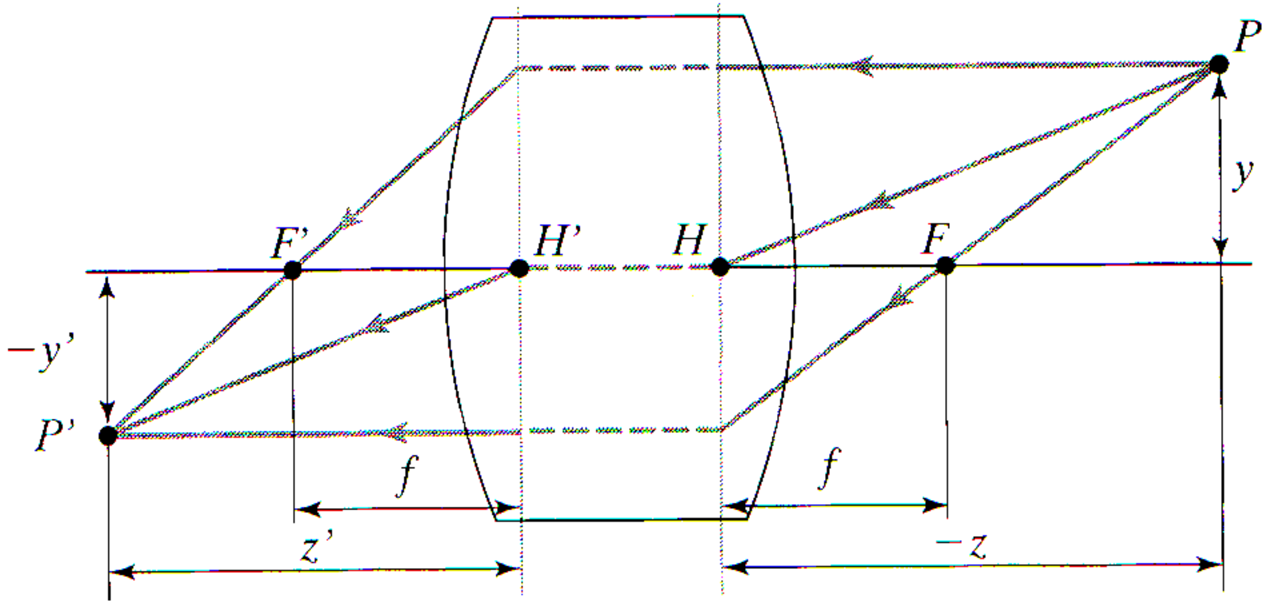


Focal Length /
Lens (Aperture) Diameter

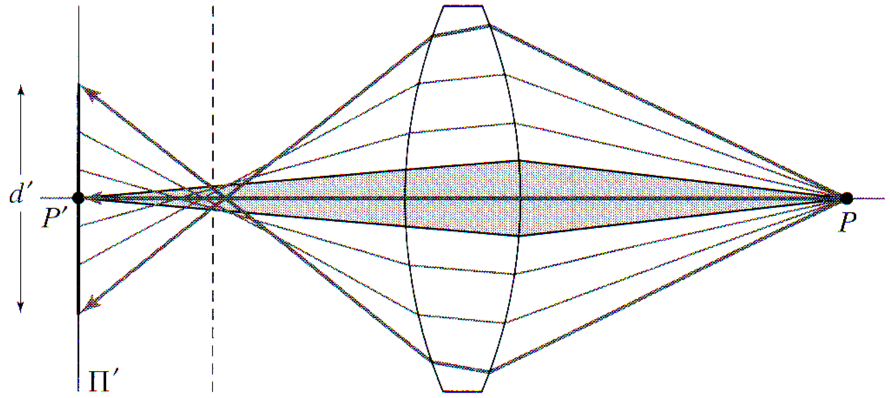
$$f\,/\,\# = N = \frac{f}{d}$$

For instance, f/16 means that the Focal Length is 16 times the Aperture Diameter (N=16).

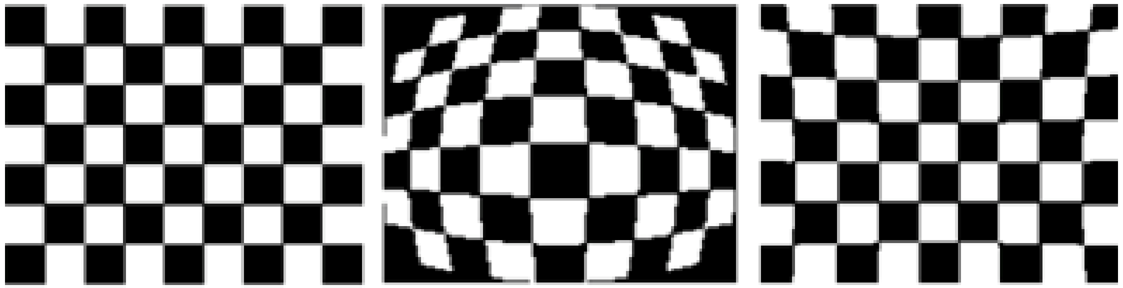
Thick Lenses



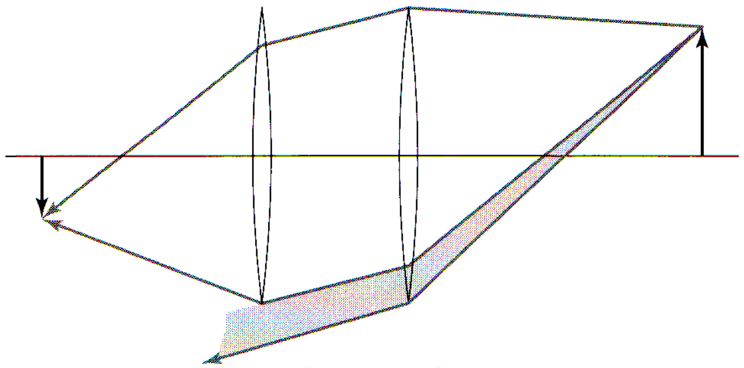
Thick Lens



Aberrations



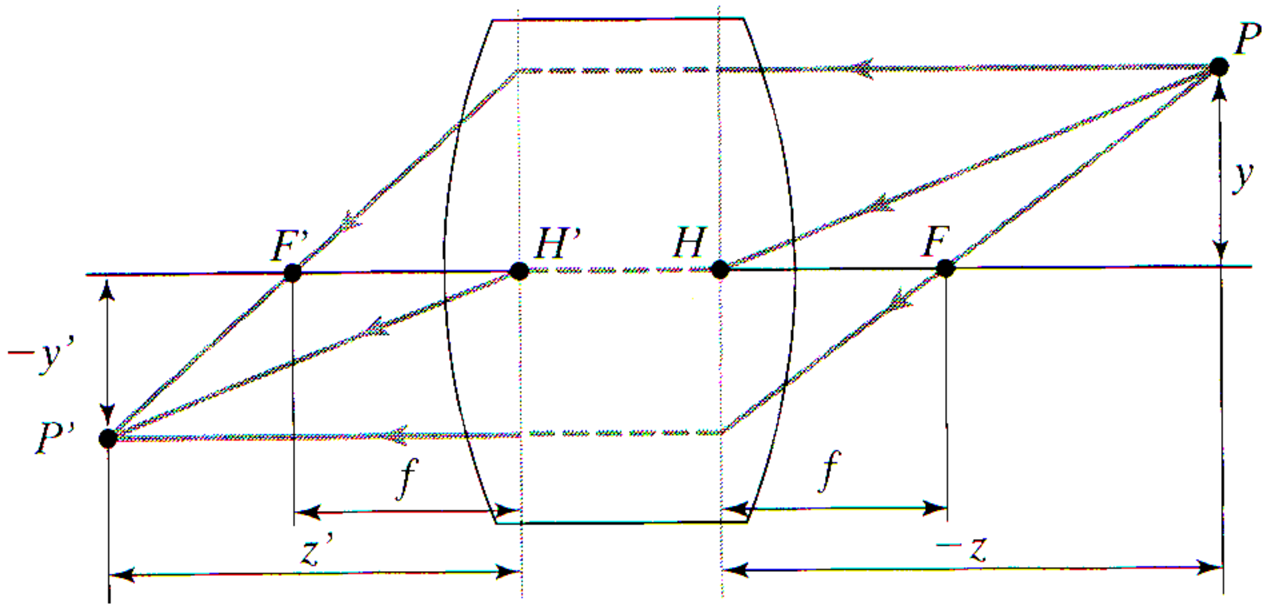
Pincushion Barrel



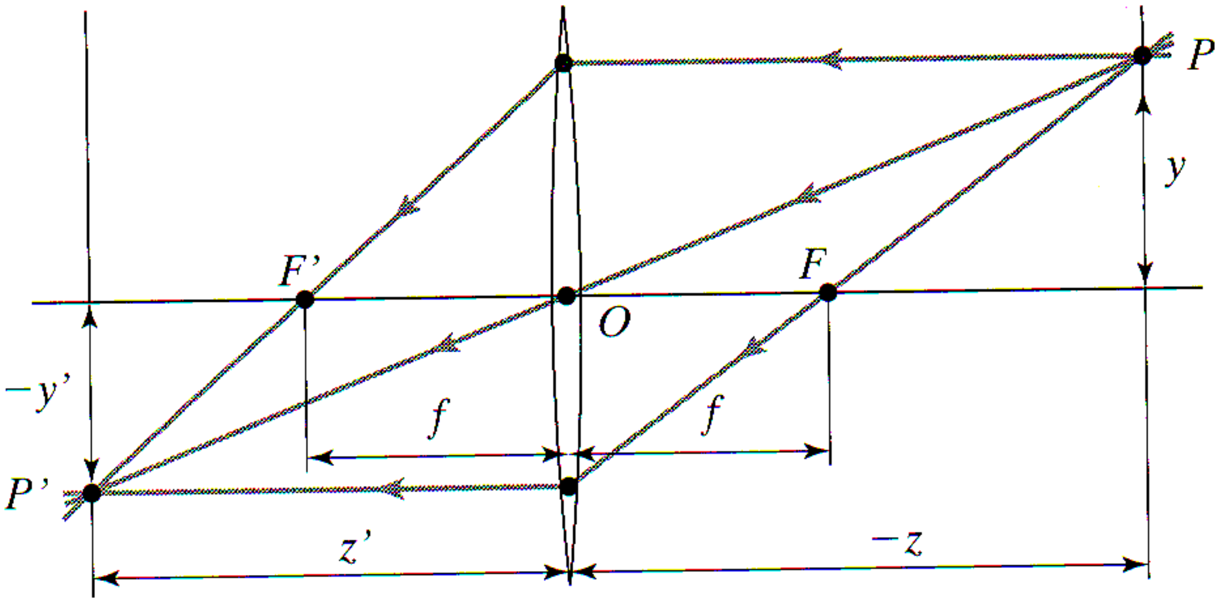
Vignetting

Thin-Lens Model

$$x = \frac{fX}{Z} \qquad y = \frac{fY}{Z}$$

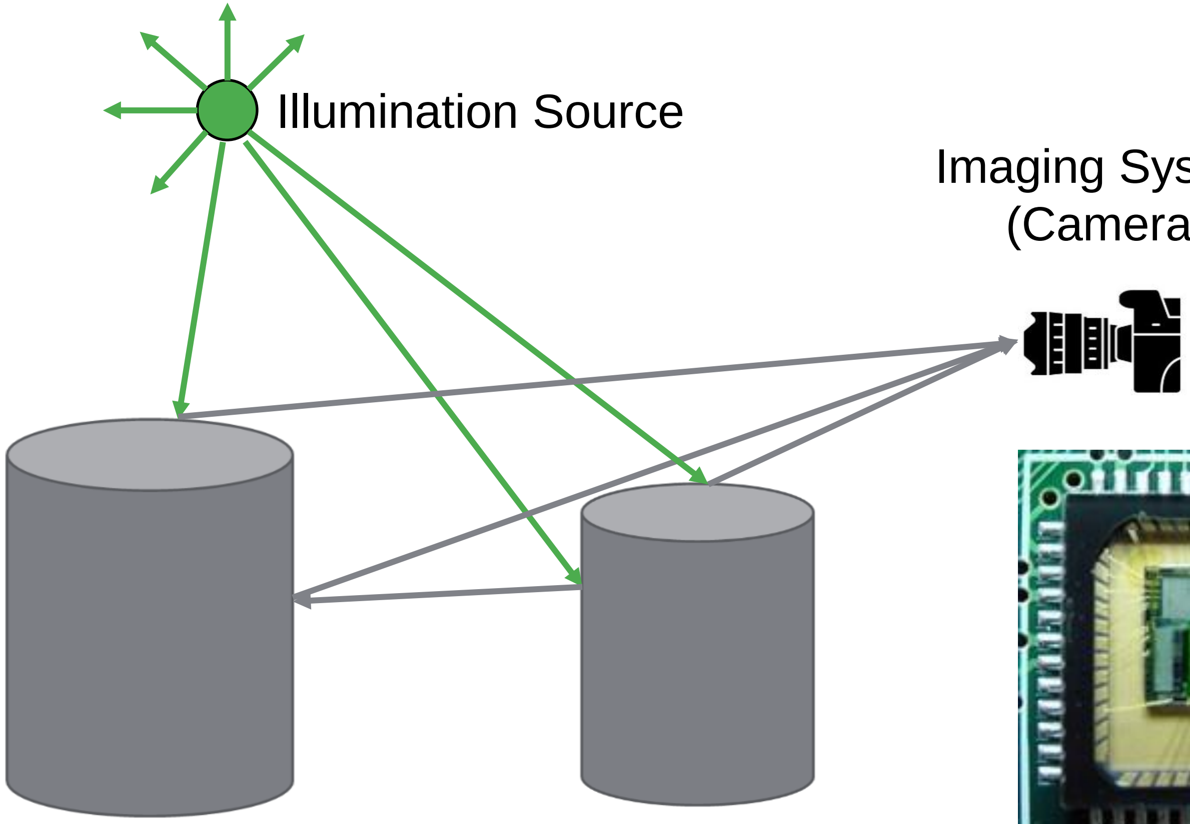


Thick Lens



Thin Lens

Formation of Digital Images



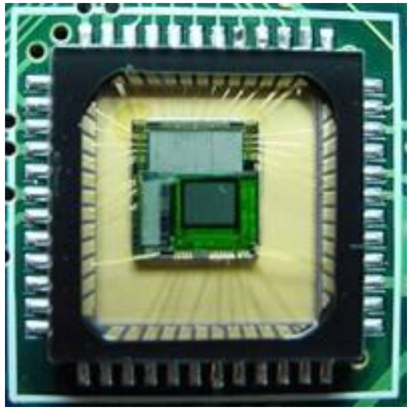
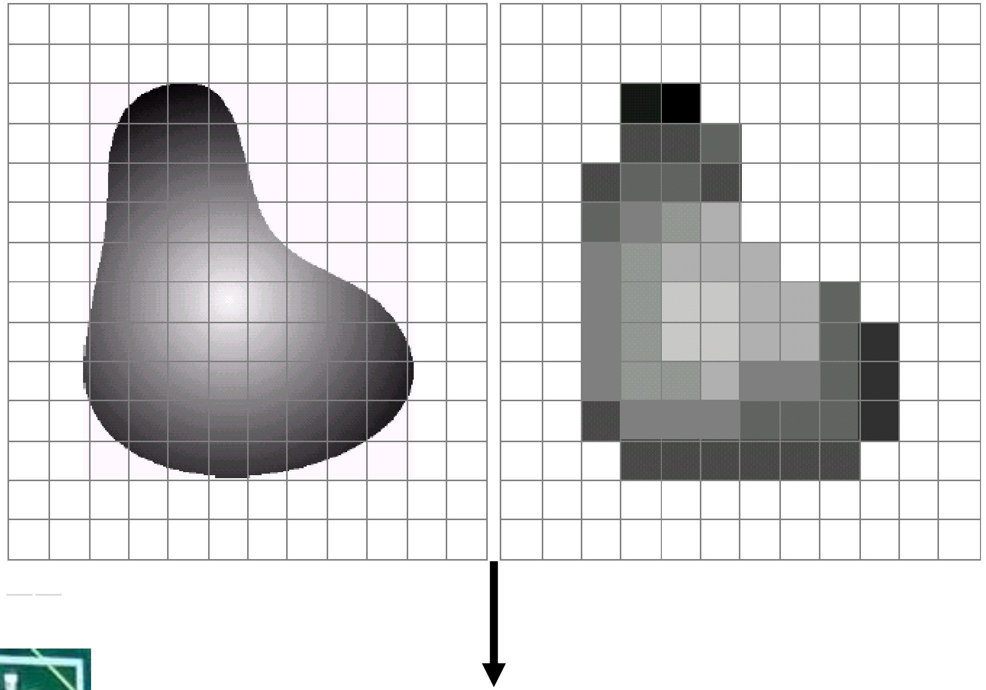
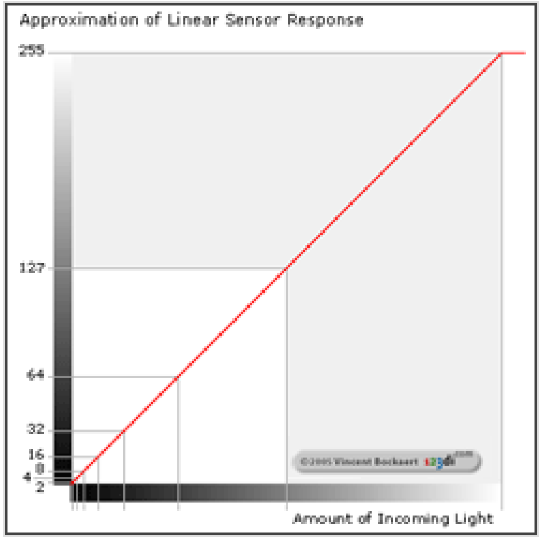


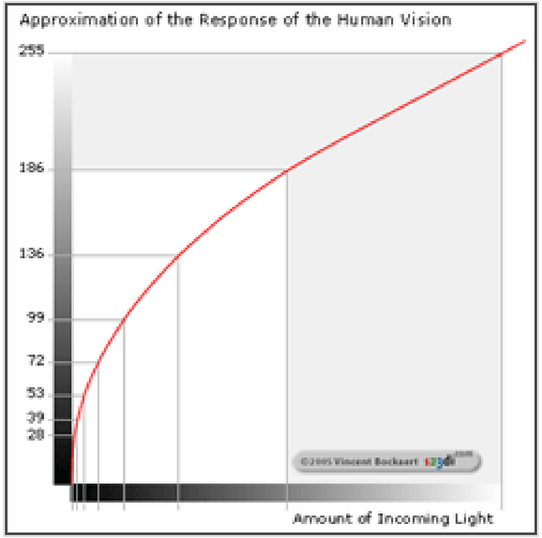
Image Sensor
(CCD/CMOS)

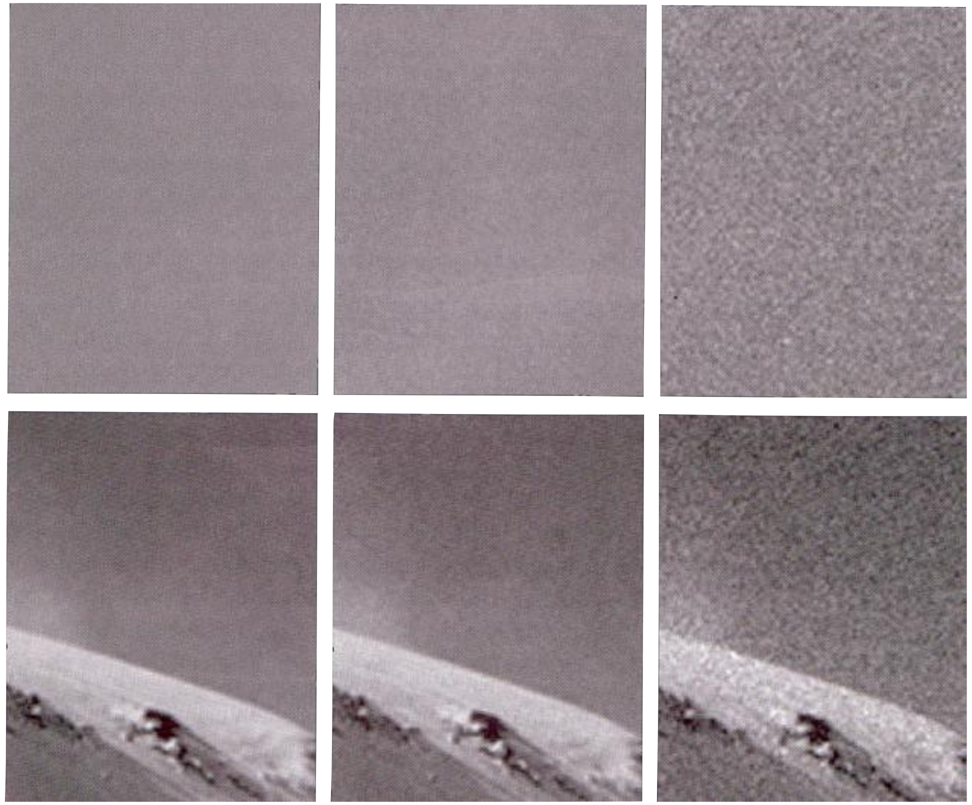


1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	1	1	1	1	1	1	1
1	1	1	0.2	0.2	0.3	1	1	1	1	1	1
1	1	0.2	0.3	0.3	0.2	1	1	1	1	1	1
1	1	0.3	0.4	0.5	0.6	1	1	1	1	1	1
1	1	0.4	0.5	0.6	0.6	0.6	1	1	1	1	1
1	1	0.4	0.5	0.8	0.8	0.6	0.6	0.3	1	1	1
1	1	0.4	0.5	0.8	0.8	0.6	0.6	0.3	0.1	1	1
1	1	0.4	0.5	0.5	0.6	0.4	0.4	0.3	0.1	1	1
1	1	0.2	0.4	0.4	0.4	0.3	0.3	0.3	0.1	1	1
1	1	1	0.2	0.2	0.2	0.2	0.2	0.2	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1

Sensor Response and Noise



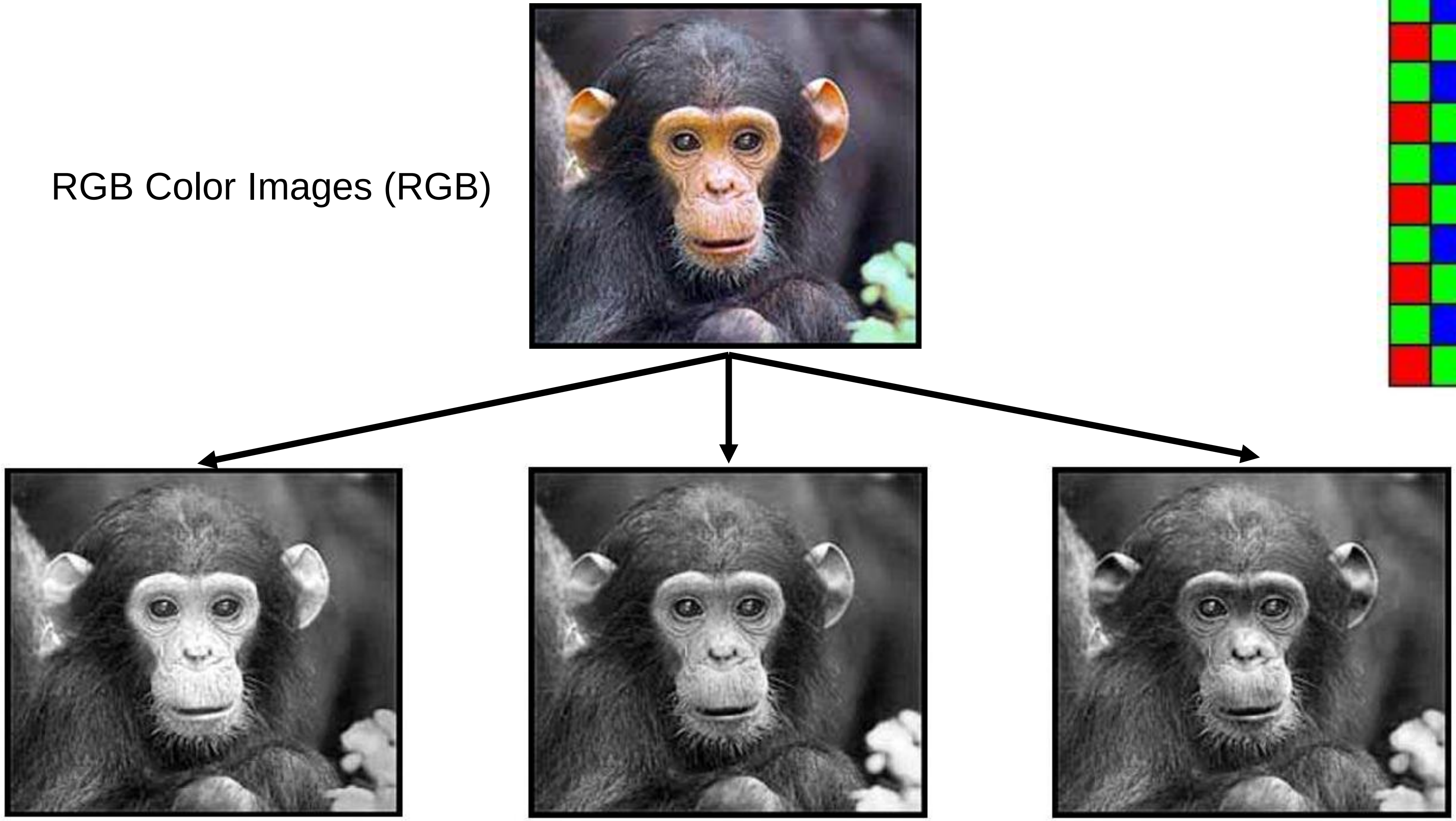




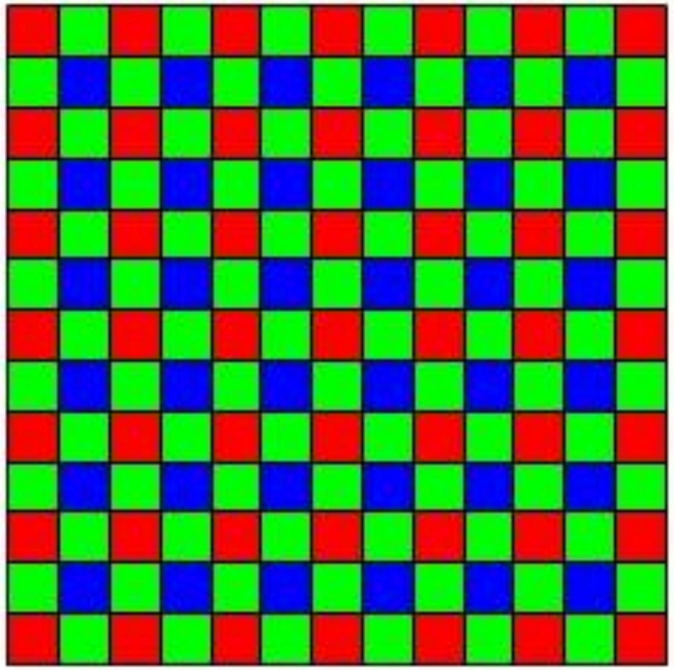
Varying Signal-2-Noise Ratio

Digital Color Images

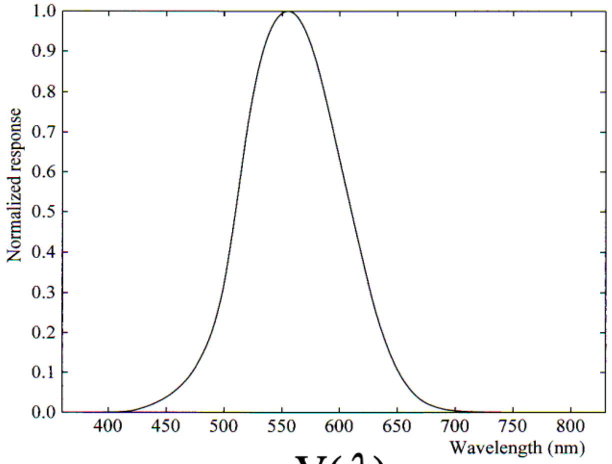
RGB Color Images (RGB)



R G B

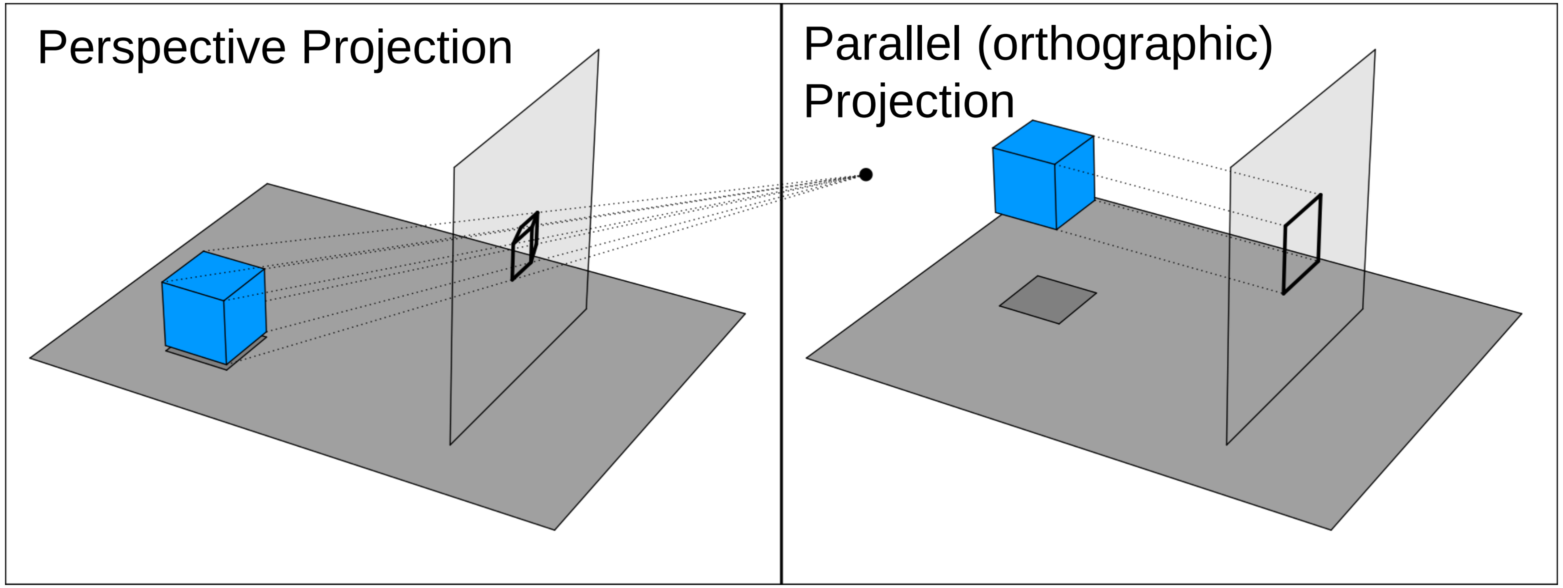


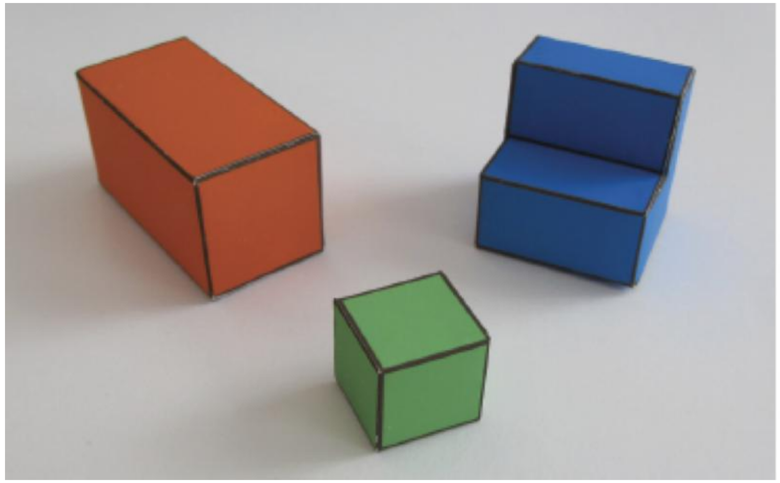
Bayer Filter

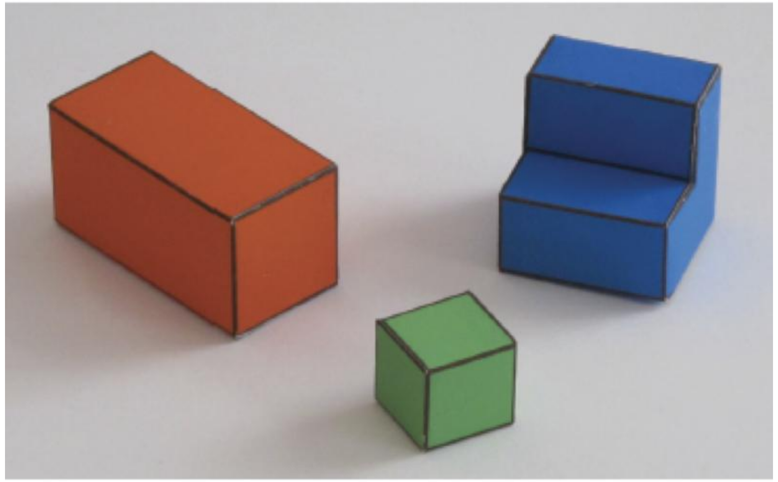


$V(\lambda)$

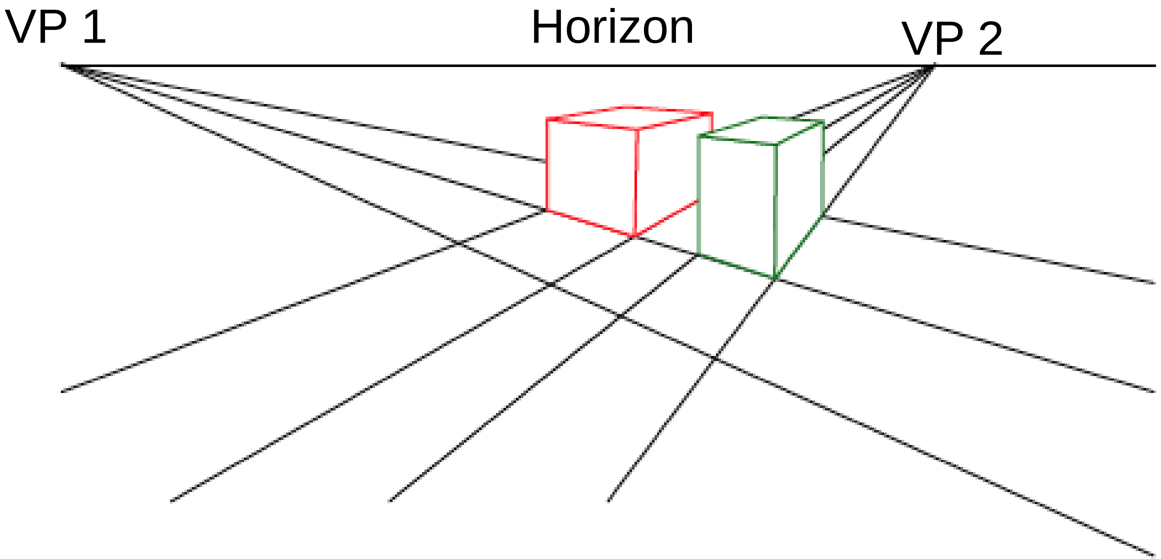
Projection


Perspective Projection
Parallel (orthographic) Projection



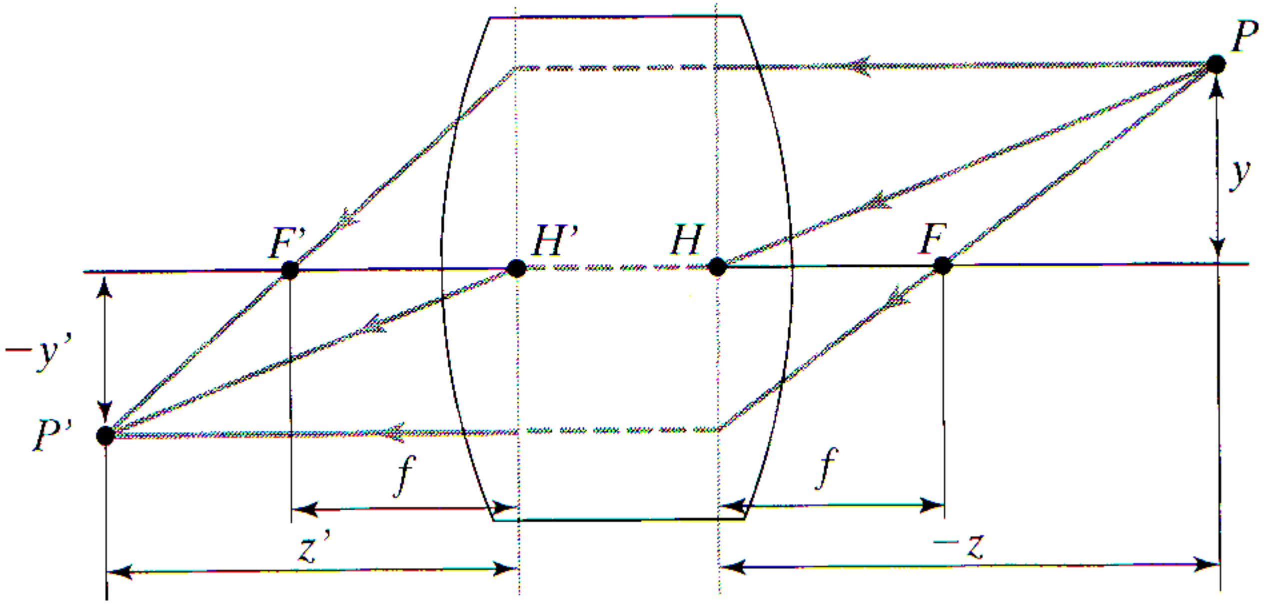


Vanishing Points

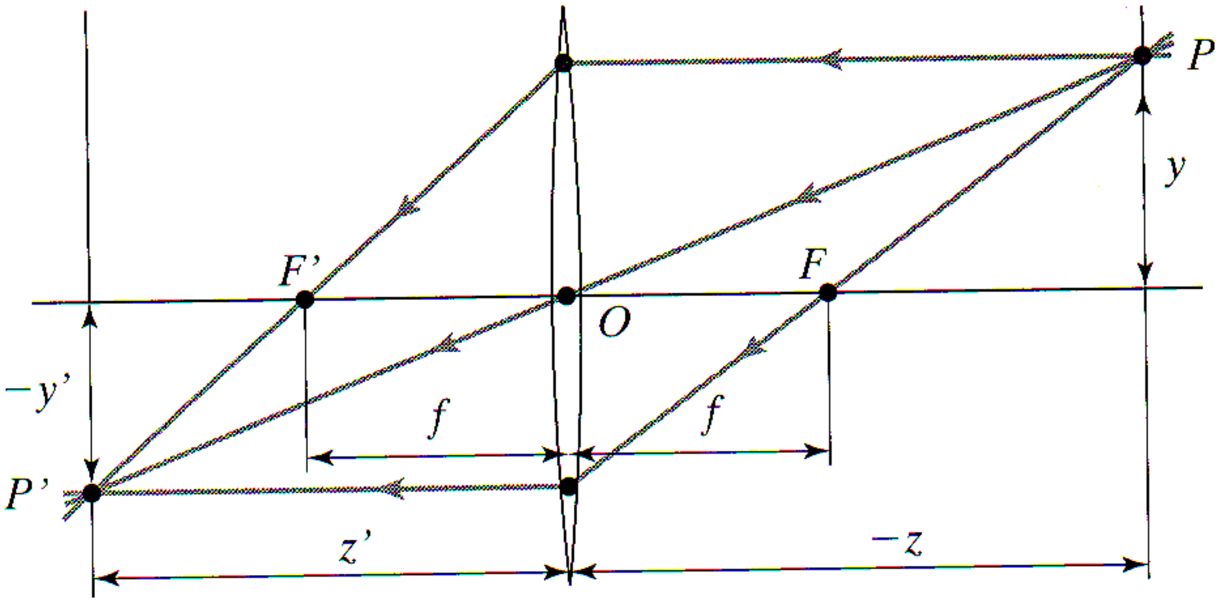


Recap: Thin-Lens Model

$$x = \frac{fX}{Z} \qquad y = \frac{fY}{Z}$$



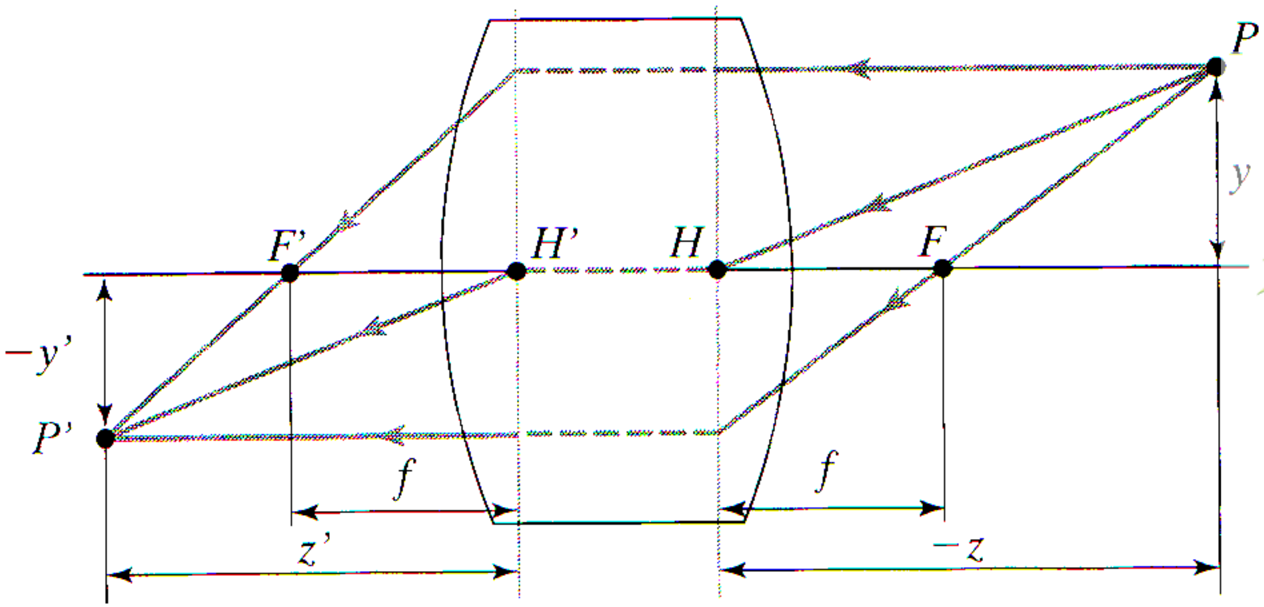
Thick Lens



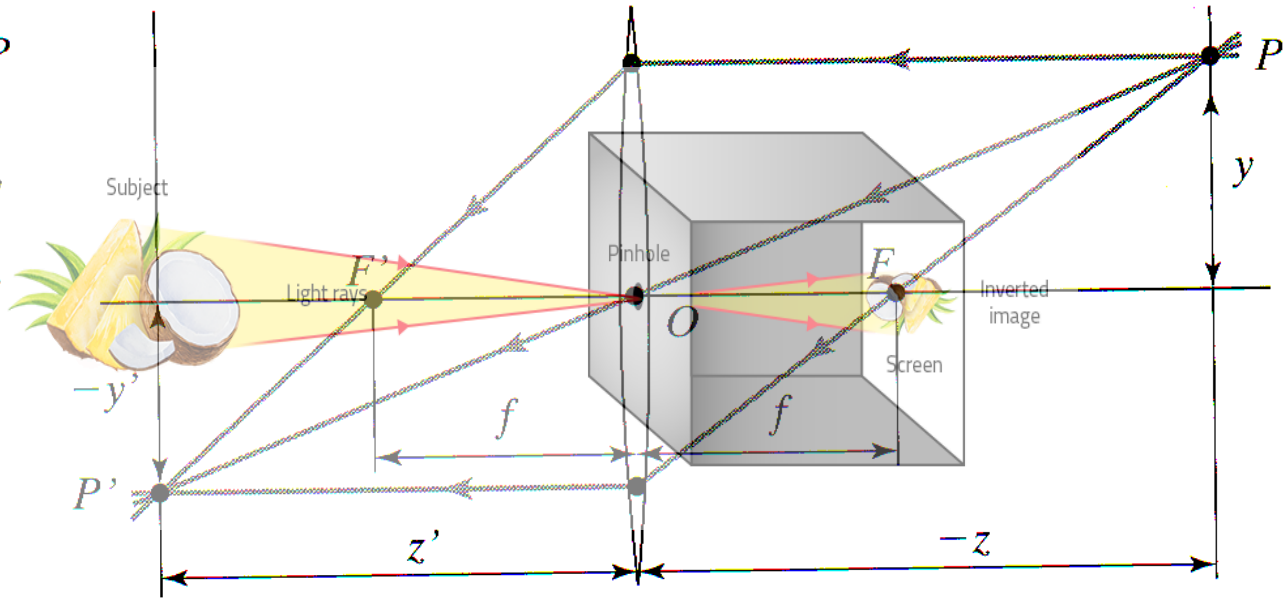
Thin Lens

Recap: Thin-Lens Model

$$x = \frac{fX}{Z} \qquad y = \frac{fY}{Z}$$

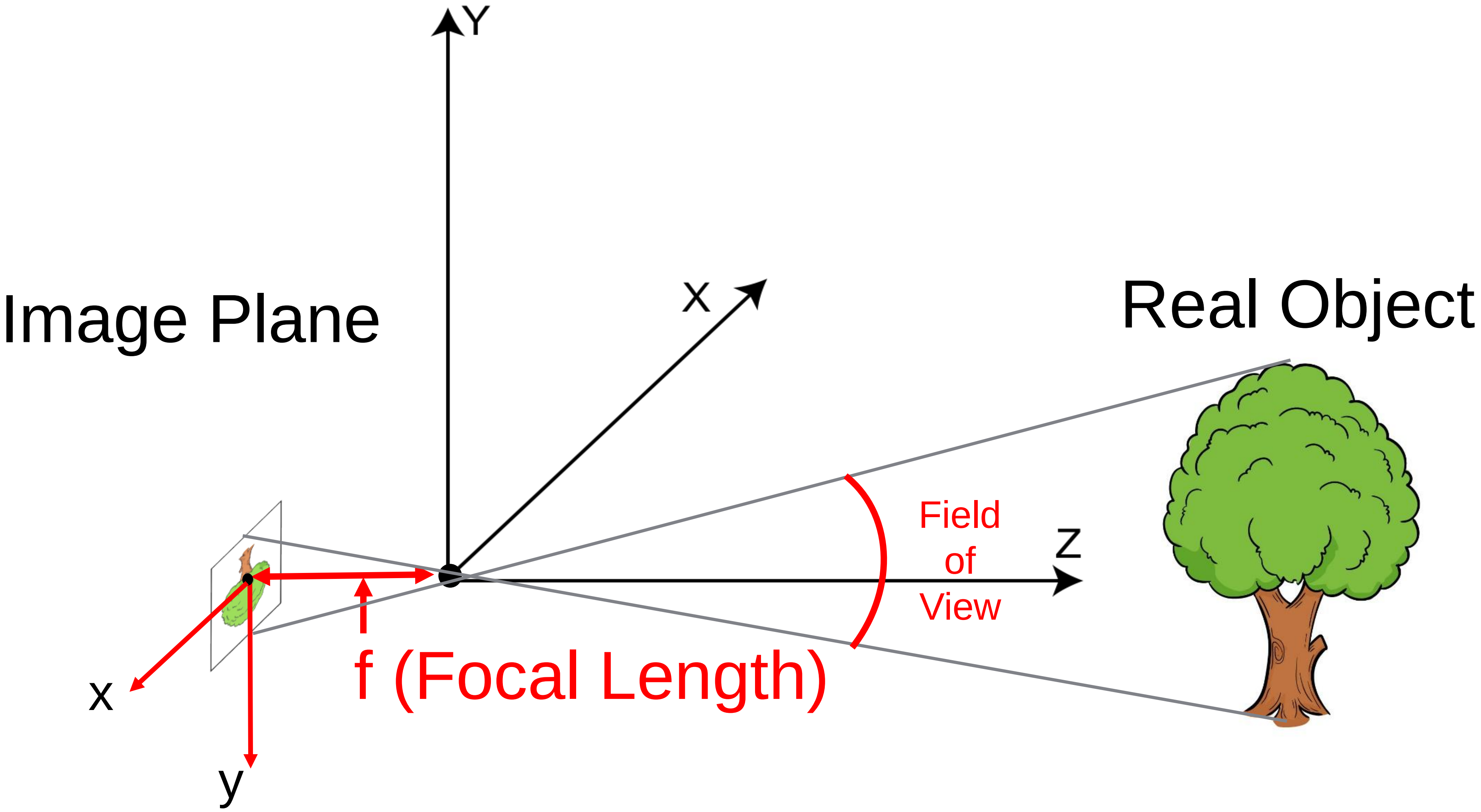


Thick Lens

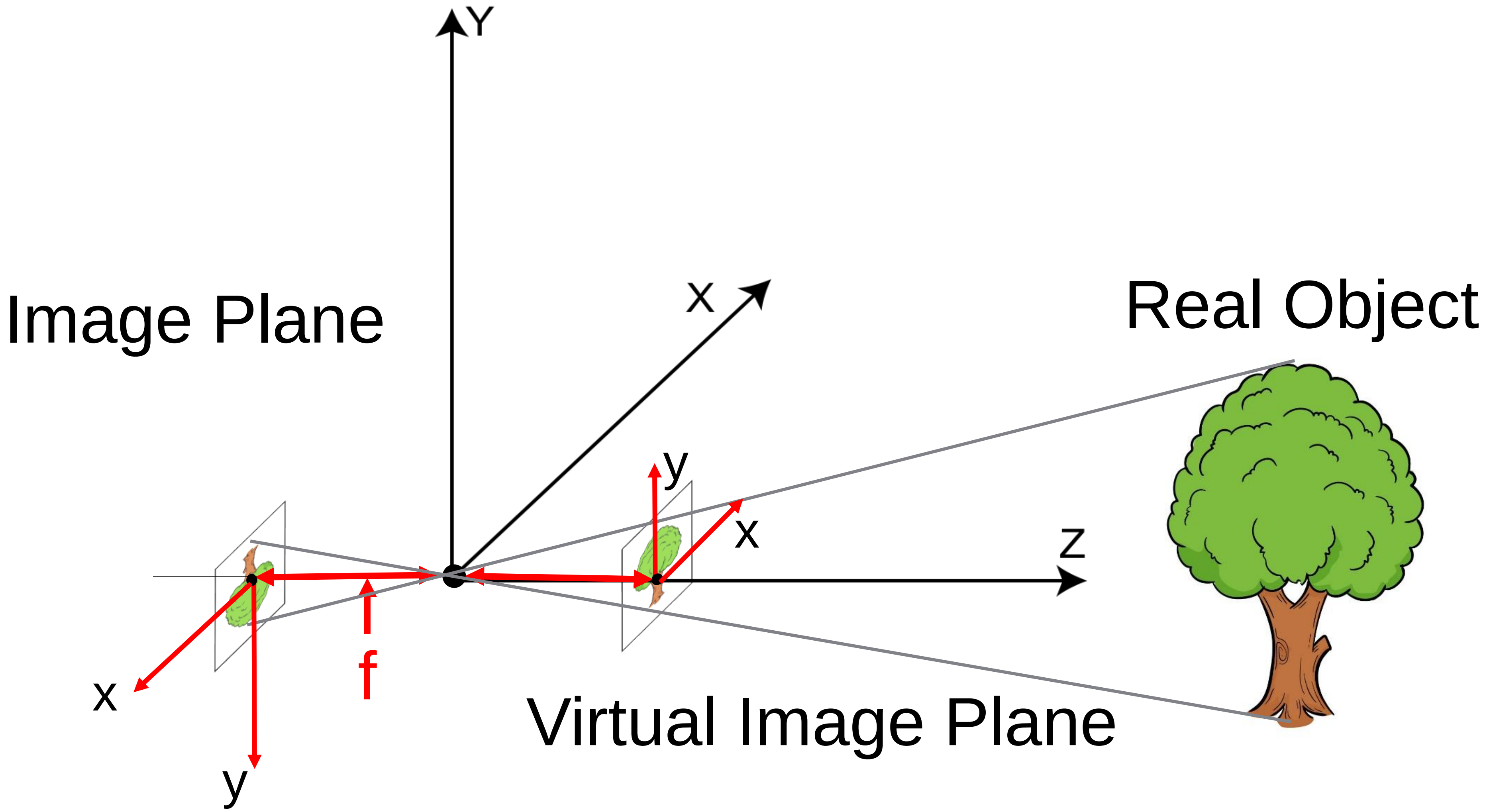


Thin Lens

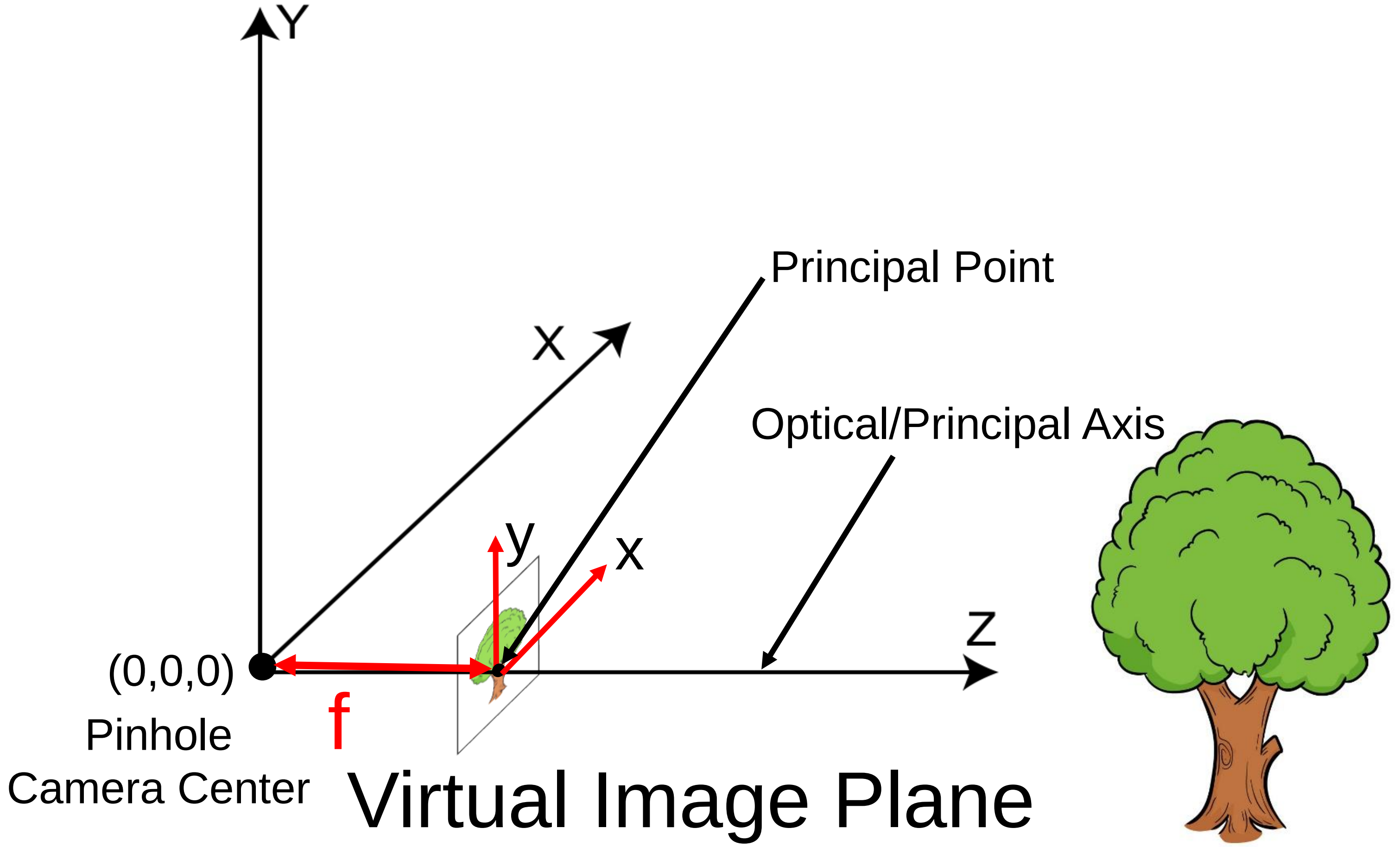
Perspective Projection



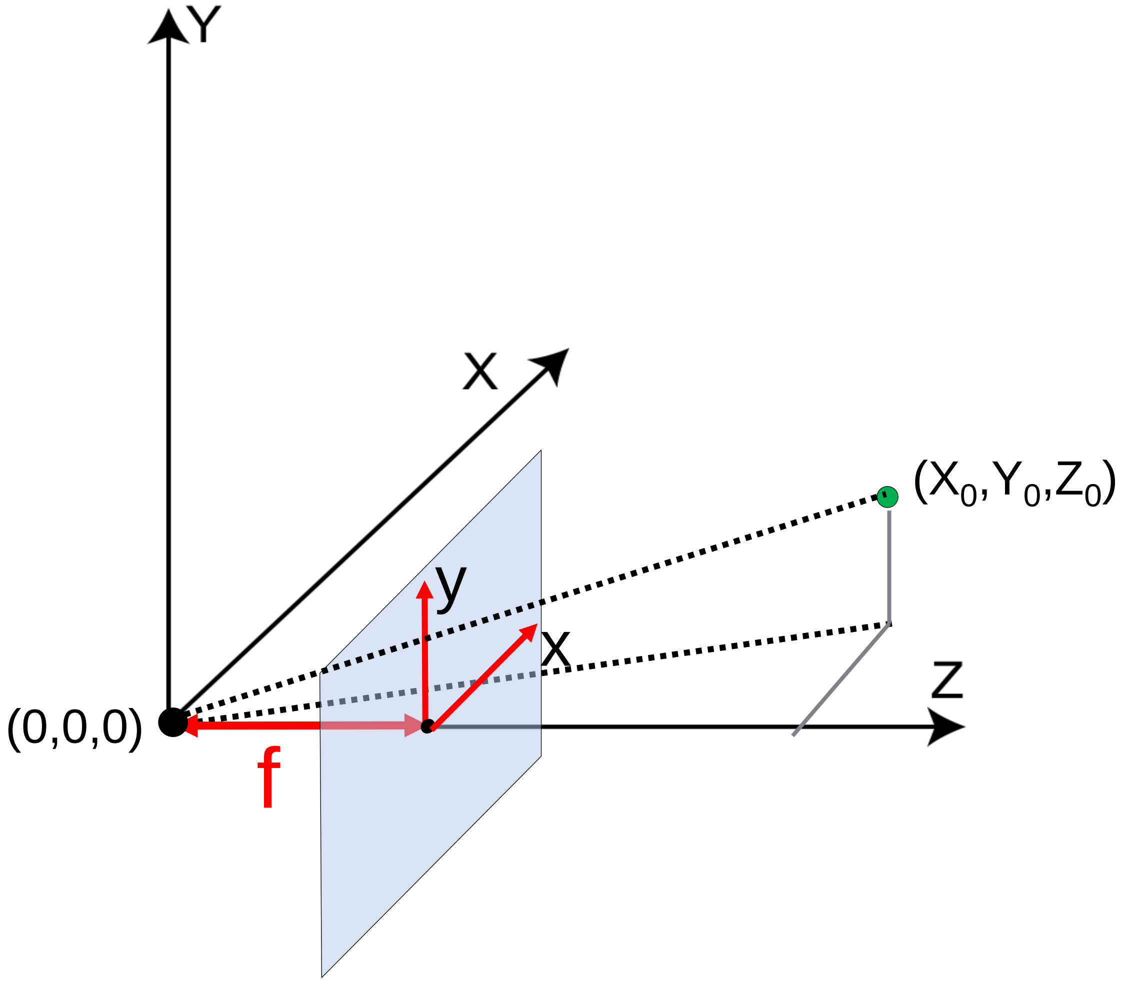
Perspective Projection

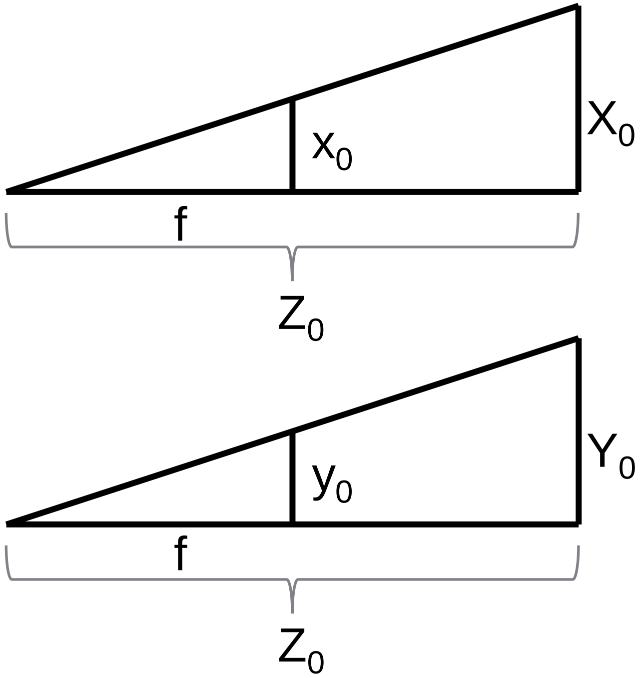


Perspective Projection



Perspective Projection



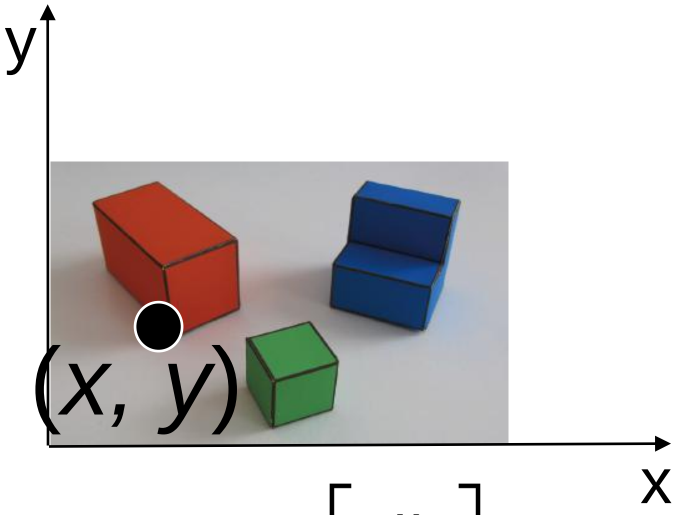


$$\rightarrow x_0 = fX_0/Z_0$$

$$\rightarrow y_0 = fY_0/Z_0$$

Homogeneous Coordinates

2D

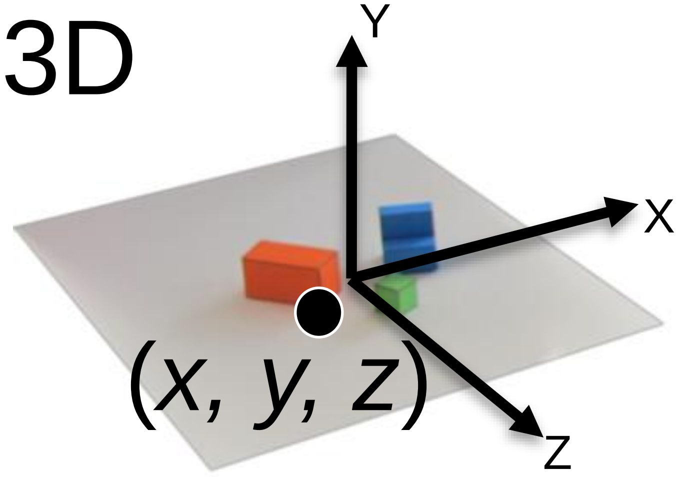


$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Heterogeneous Image Coordinates

Homogeneous Image Coordinates

3D



$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Homogeneous Coordinates

From Heterogeneous to Homogeneous:

$$(x, y) \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

From Homogeneous to Heterogeneous:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \rightarrow \left(\frac{x}{w}, \frac{y}{w}\right)$$

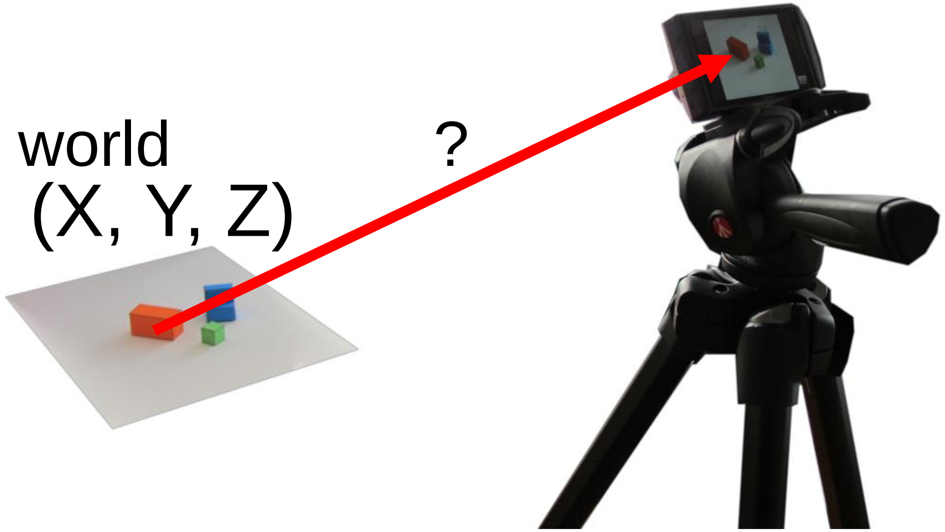
Mathematical Camera Model

2D image (x, y)

3D world

?

(X, Y, Z)



Pixel Coordinates

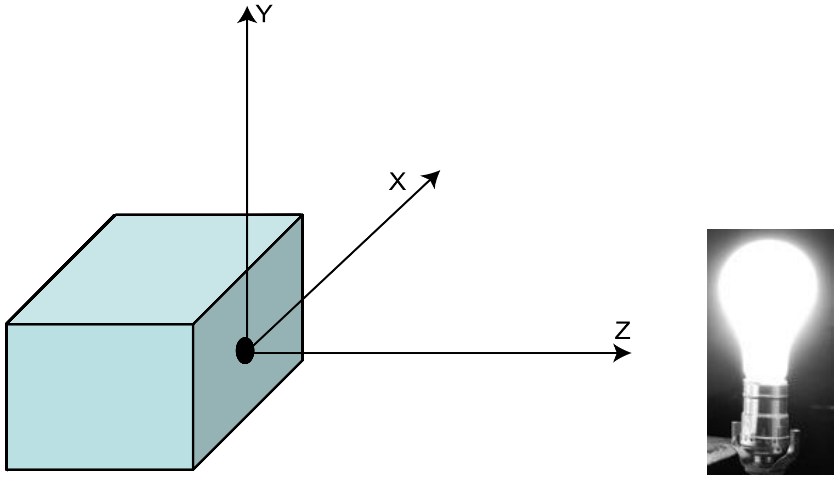
$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

World Coordinates

Perspective Projection

$$x = \frac{fX}{Z} \qquad y = \frac{fY}{Z}$$

$$x = \frac{fX}{Z} \qquad y = \frac{fY}{Z}$$



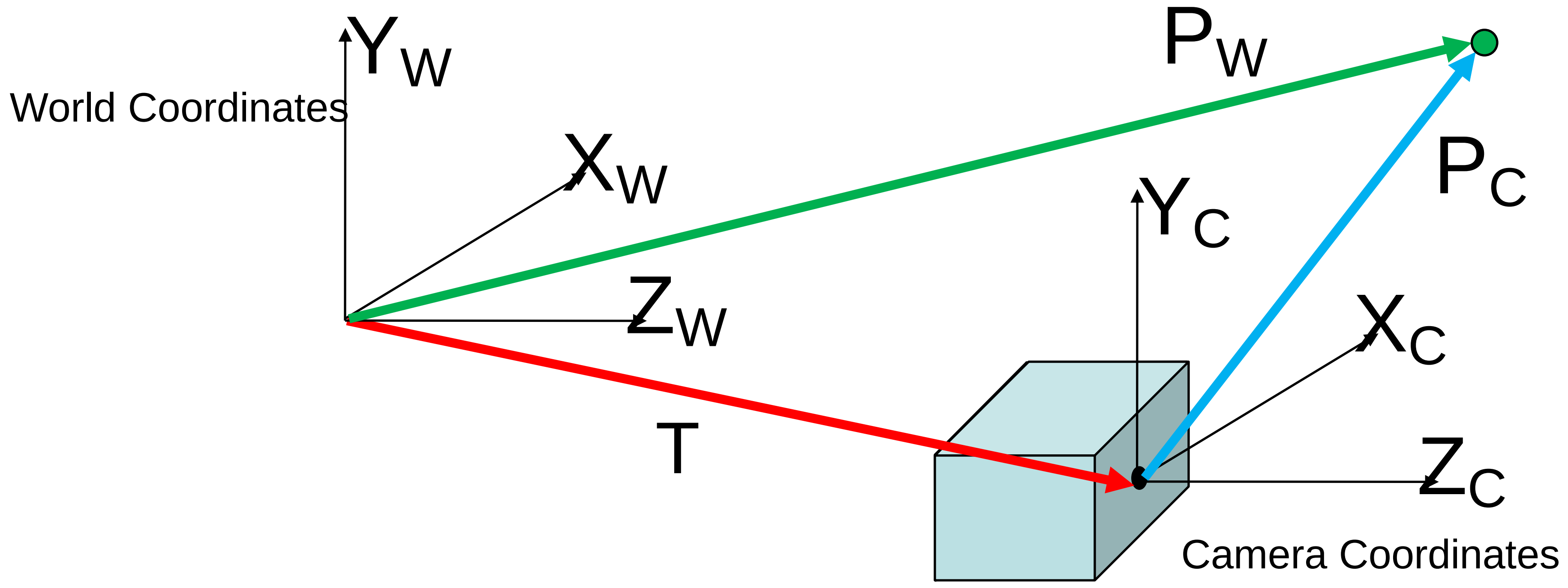
Let’s assume the Origin of the World Coordinate System at the Aperture of the Pinhole.

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} \longrightarrow (f\,X/Z,\ f\,Y/Z)$$

When changing to Pixels, there will be an arbitrary Scaling in horizontal and vertical Directions.

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} aX \\ bY \\ Z \end{bmatrix} \longrightarrow (a\,X/Z,\ b\,Y/Z)$$

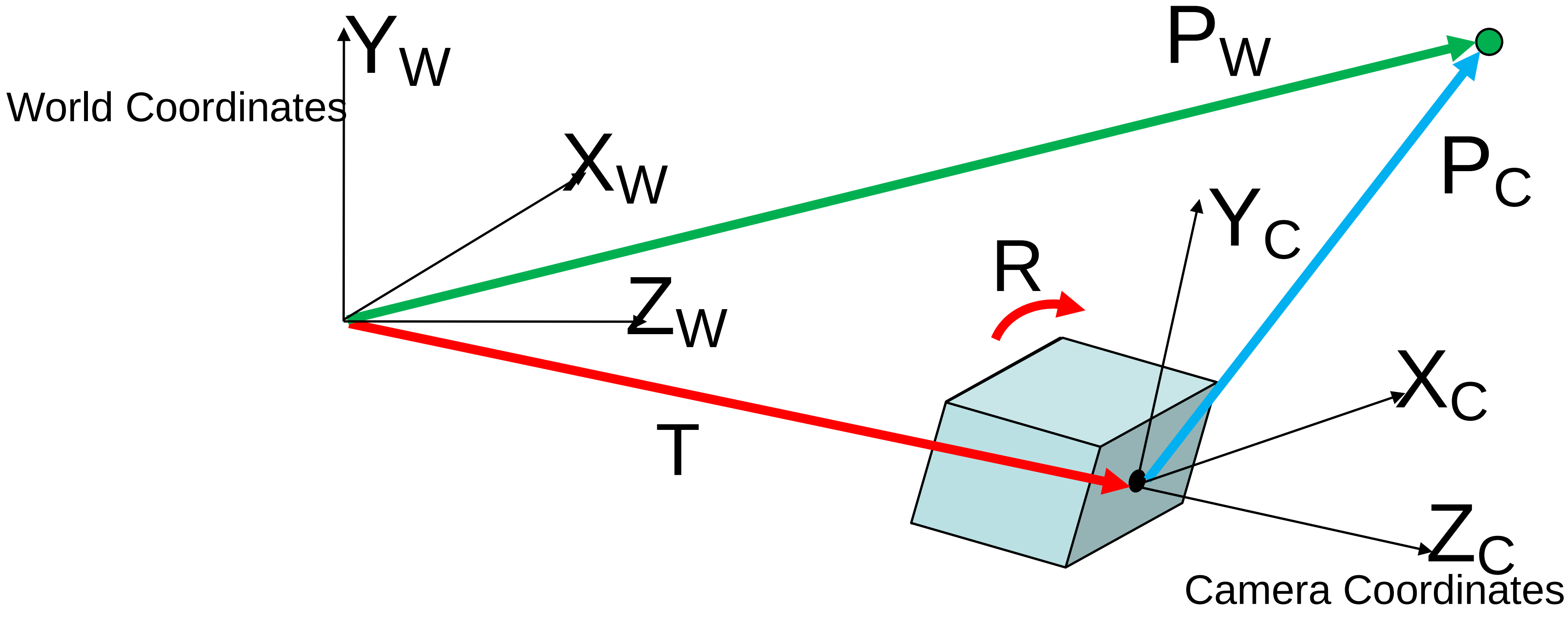
Camera Translation



In Heterogeneous Coordinates:

$$P_C = P_W - T$$

Camera Rotation



In Heterogeneous Coordinates:

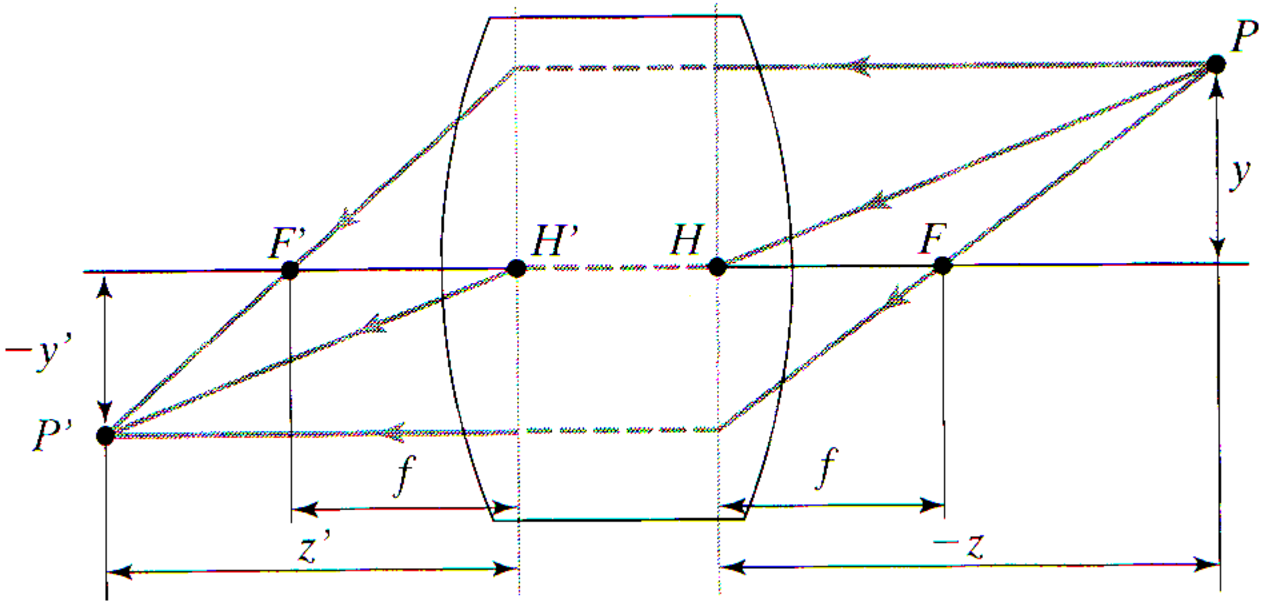
$$P_C = R(P_W - T)$$

Simplified Mathematical Camera Model

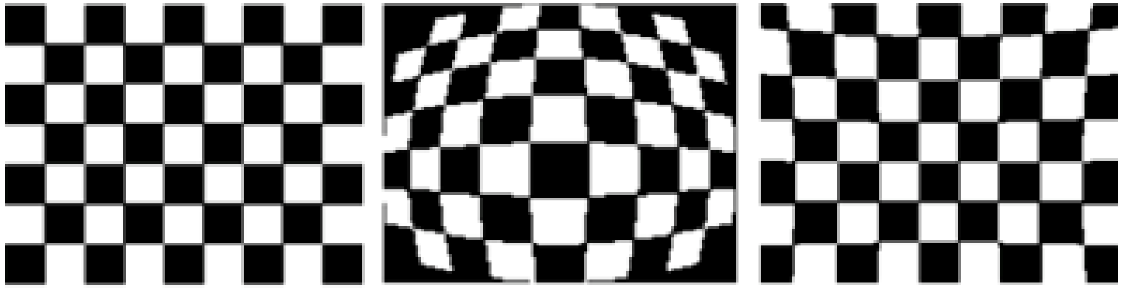
$$
\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \underbrace{\overset{[3x3]}{\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix}}}_{\text{Intrinsic Parameters (K)}} \cdot \underbrace{\overset{[3x3]}{\begin{bmatrix} \mathbf{R} \end{bmatrix}} \cdot \overset{[3x4]}{\begin{bmatrix} \mathbf{I} & \mathbf{-T} \end{bmatrix}}}_{\text{Extrinsic Parameters}} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}
$$

Note, that in Practice there are more intrinsic Parameters: e.g. Principle Point and Skew Angle describing Misalignment of Sensor on Optical Axis.

Radial Distortion

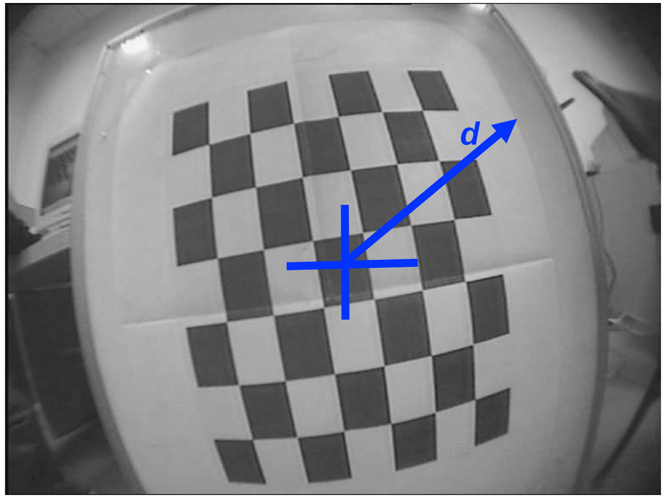


Thick Lens



Pincushion

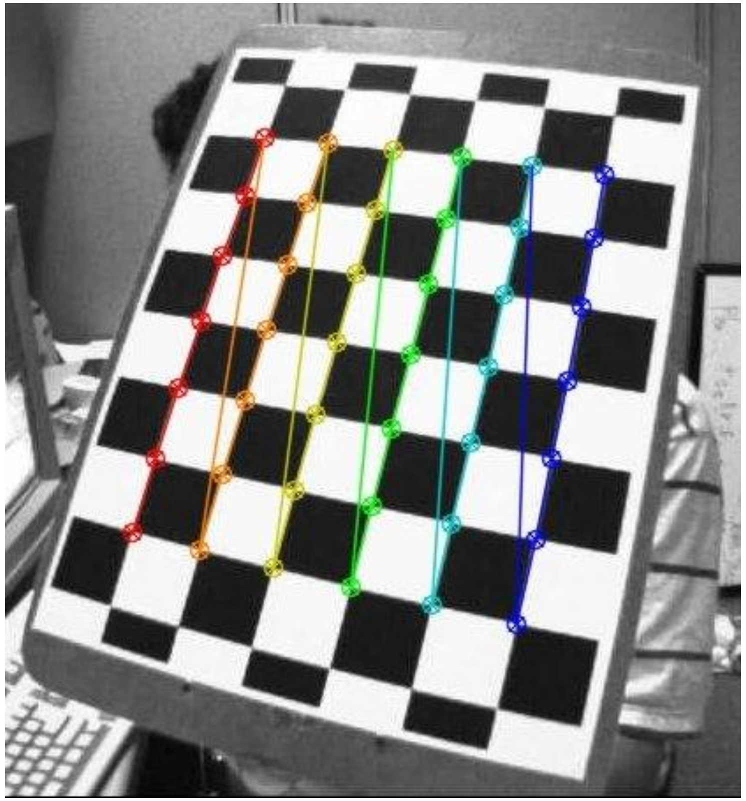
Barrel



Radial Distortion

If Lens Distortion is considered, our Mathematical Model becomes non-linear (usually a Polynomial Function)!

Camera Calibration



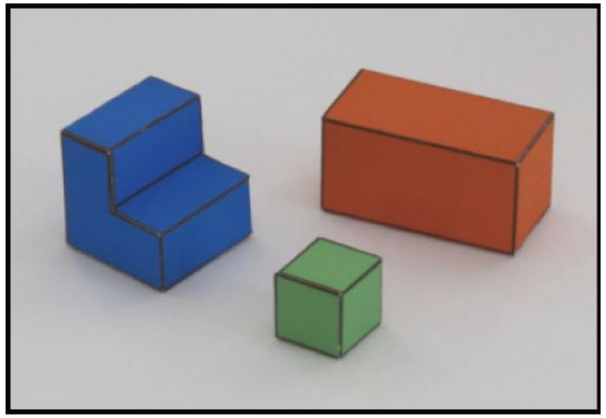
Images of known Calibration Pattern

cv.calibrateCamera() returns the Camera Matrix, distortion Coefficients, Rotation and Translation Vectors etc.

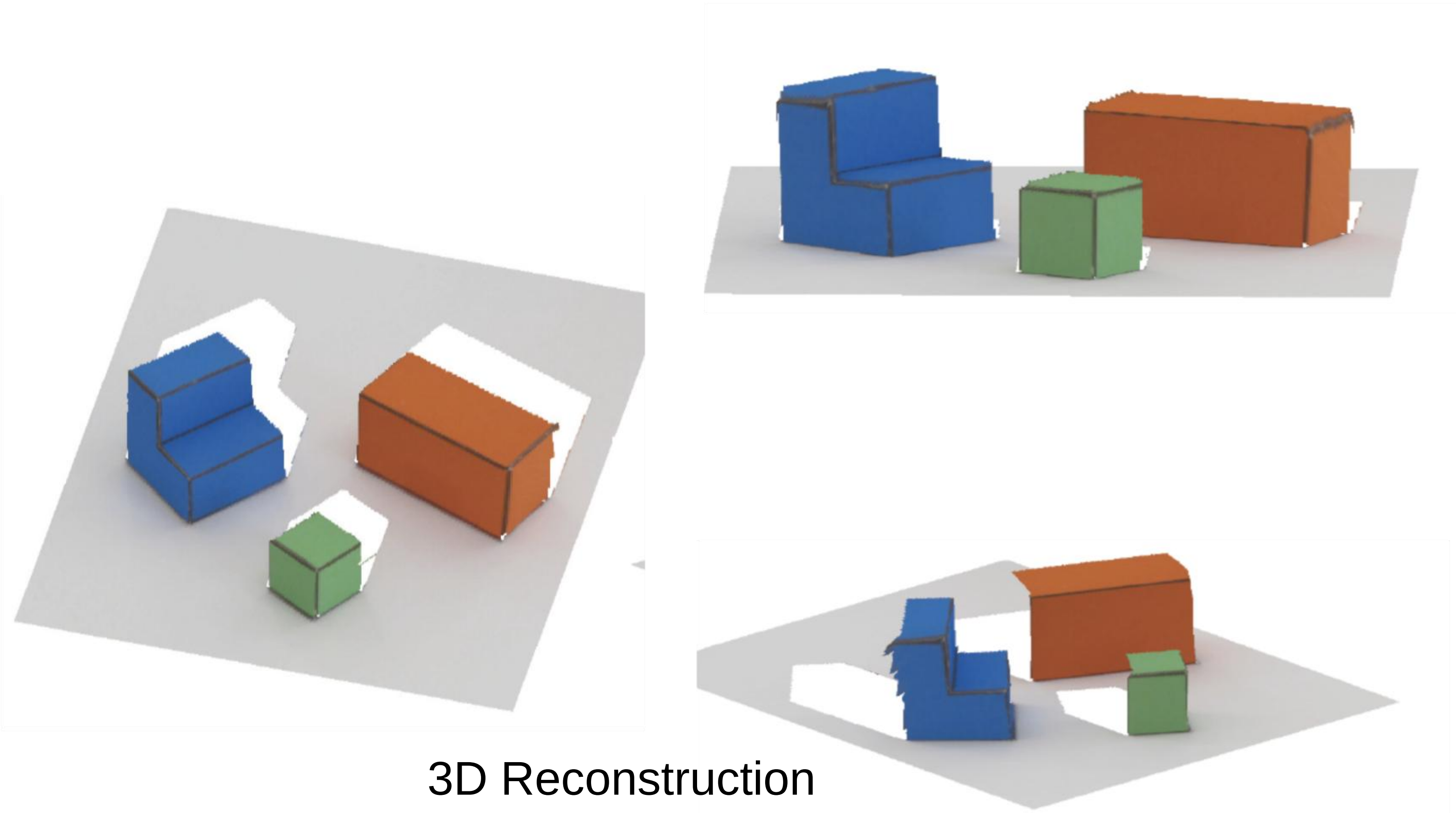
Optimization minimizes Re-Projection Error of Checker Corners

In OpenCV: https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html

Why do we need all of this?

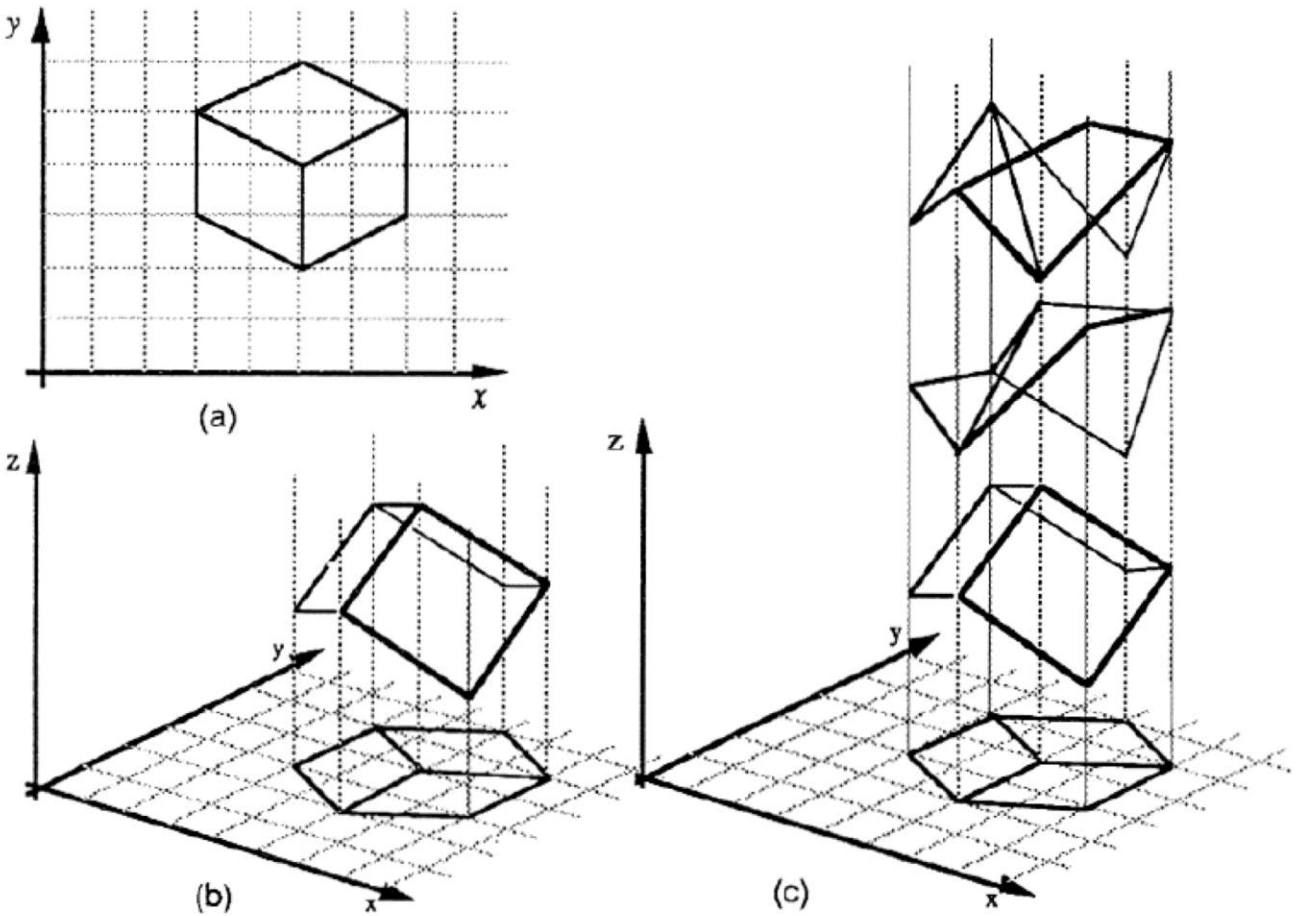


2D Images



3D Reconstruction

3D Reconstruction is not simple!



3D Reconstruction from a single Image is an under-constrained Problem.

On a lower Level: Detect Features (e.g., Edges)

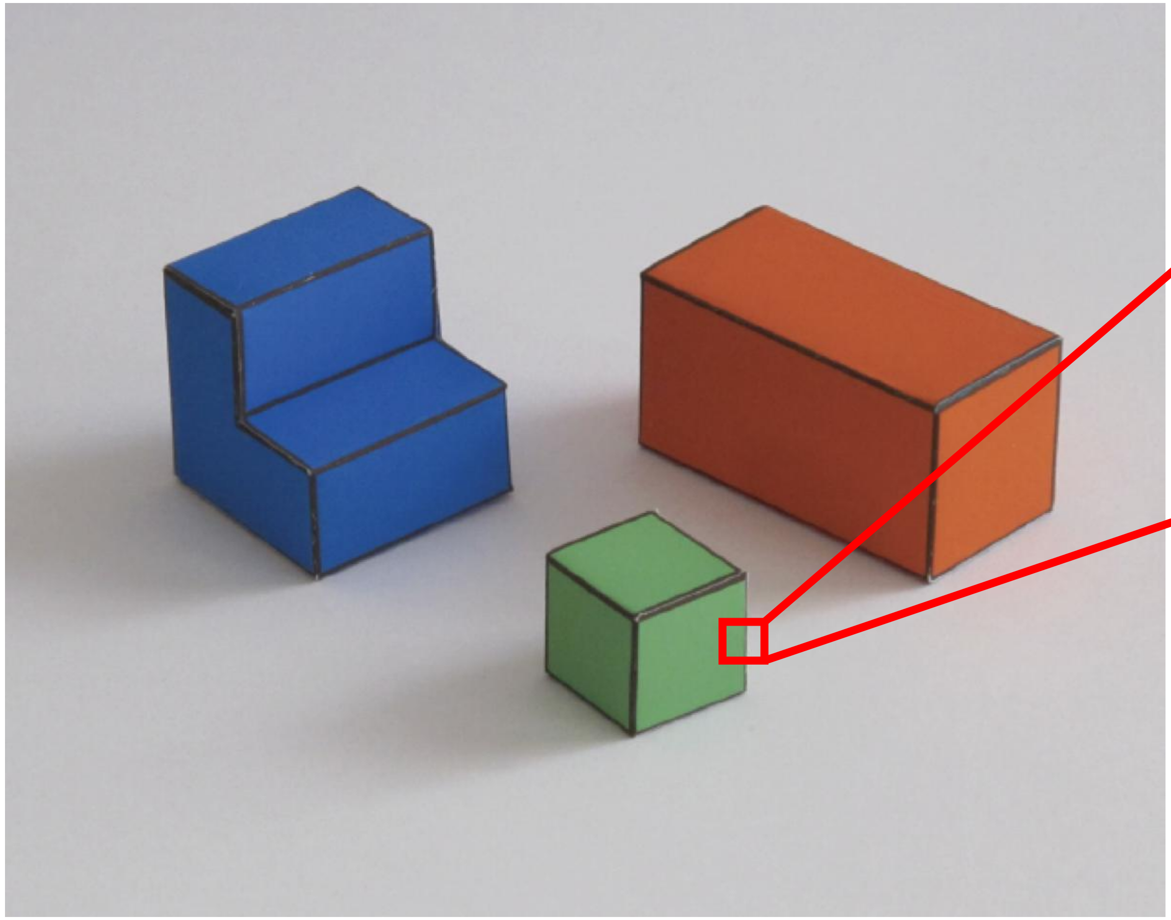
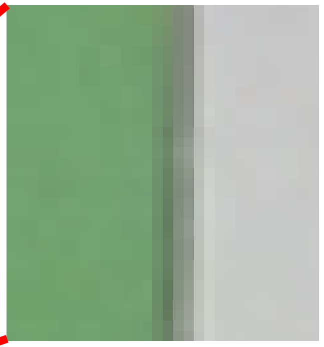


Image Patch



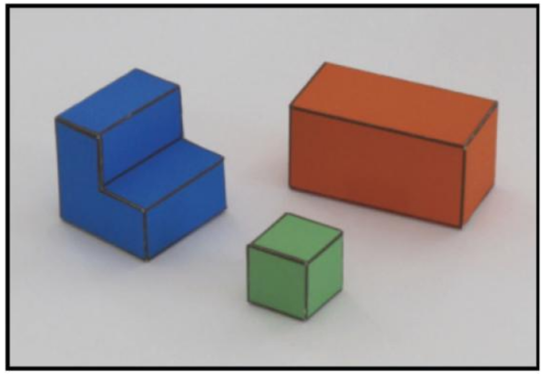
… 125, 126, 50, 10, 223, 223, …
…,124, 126, 50, 10, 223, 224, …
…,125, 127, 51, 9, 223, 224, …
…

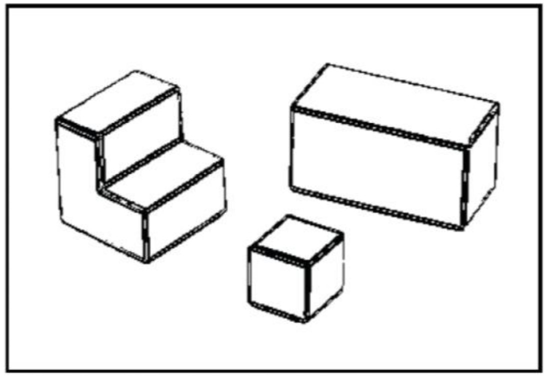
What Measure can tell us that there is an Edge here?

Image gradient:

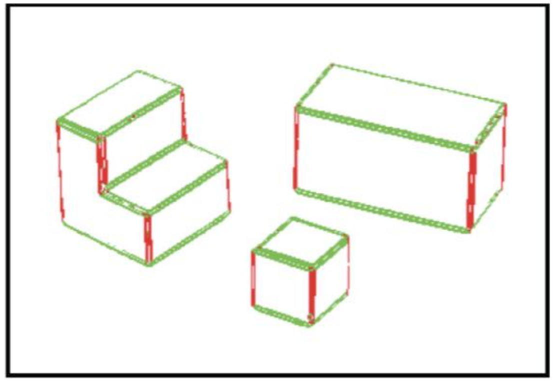
$$\nabla \mathrm{I} = \left(\frac{\partial \mathrm{I}}{\partial x}, \frac{\partial \mathrm{I}}{\partial y} \right)$$

On a lower Level: Detect Features (e.g., Edges)





Using E(x,y)



Using θ(x,y)

Image gradient:

$$\nabla \mathbf{I} = \left(\frac{\partial \mathbf{I}}{\partial x}, \frac{\partial \mathbf{I}}{\partial y} \right)$$

Approximation image derivative:

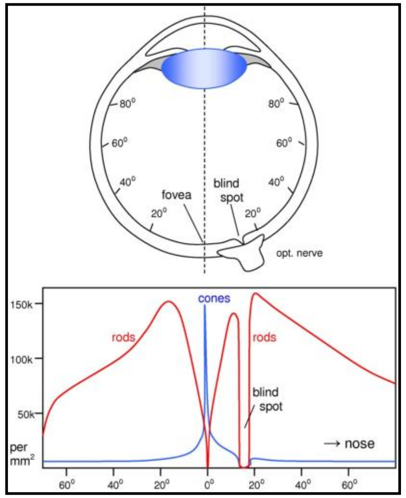
$$\frac{\partial \mathbf{I}}{\partial x} \simeq \mathbf{I}(x, y) - \mathbf{I}(x-1, y)$$

Edge strength $E(x,y) = |\nabla \mathbf{I}(x,y)|$

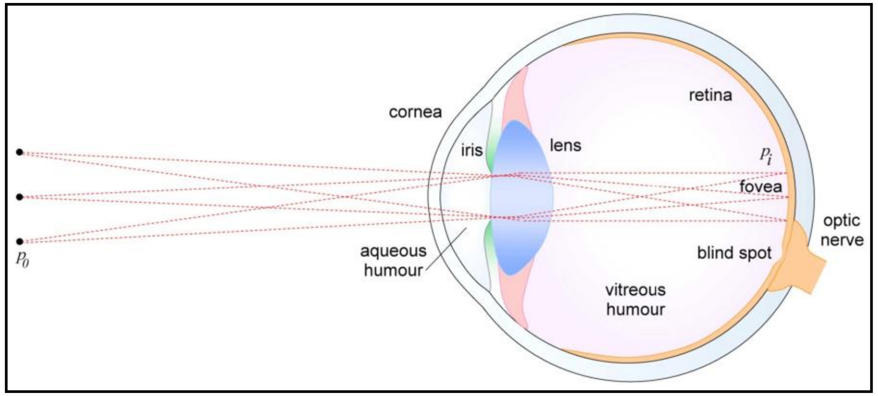
Edge orientation: $\theta(x,y) = \angle \nabla \mathbf{I} = \arctan \frac{\partial \mathbf{I}/\partial y}{\partial \mathbf{I}/\partial x}$

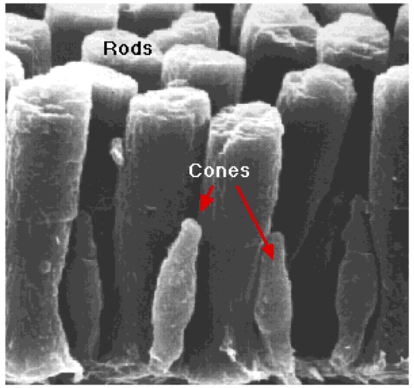
Edge normal: $\mathbf{n} = \frac{\nabla \mathbf{I}}{|\nabla \mathbf{I}|}$

How Humans do it?



Distribution of cones and rods in the eye

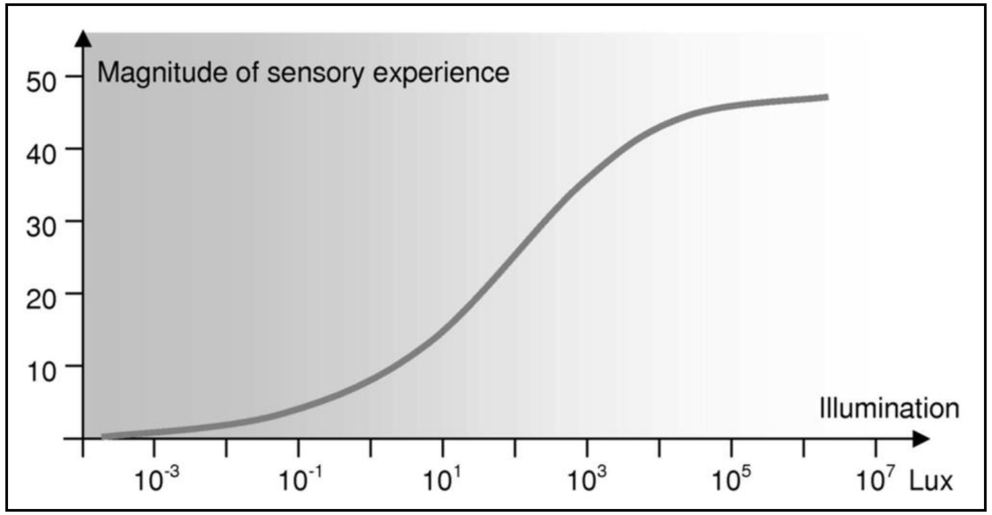




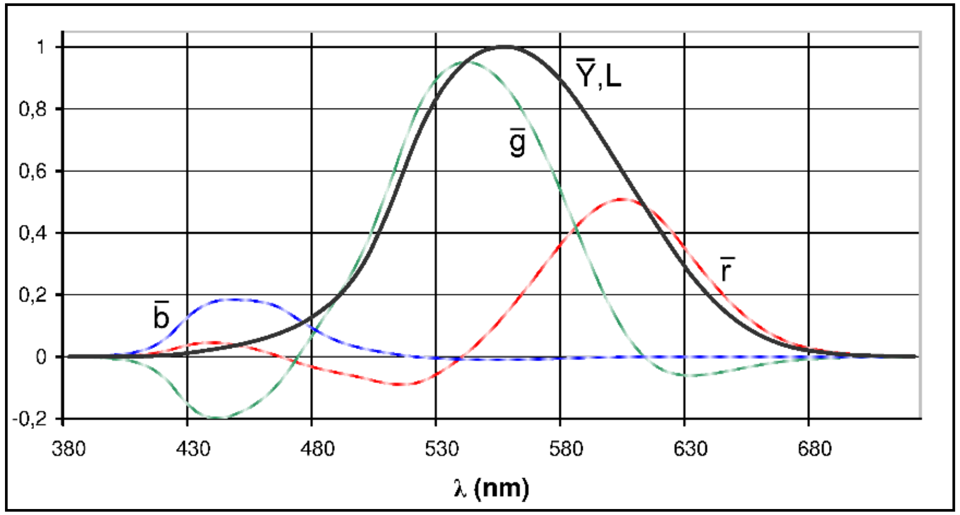
The Human Eye is a biological Camera (Lens, Aperture=Iris, Sensor=Retina) and is used for Measurement=Sensing.

The Perception of the Visual Cortex is really what we want to mimic with Computer Vision!

Human Visual Perception (Sensory Basics)



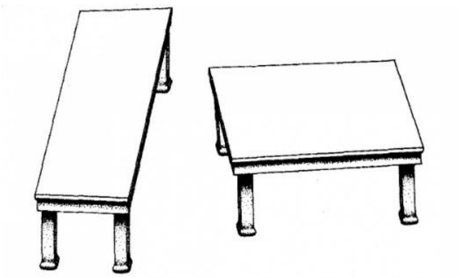
Magnitude of sensory experience vs. brightness



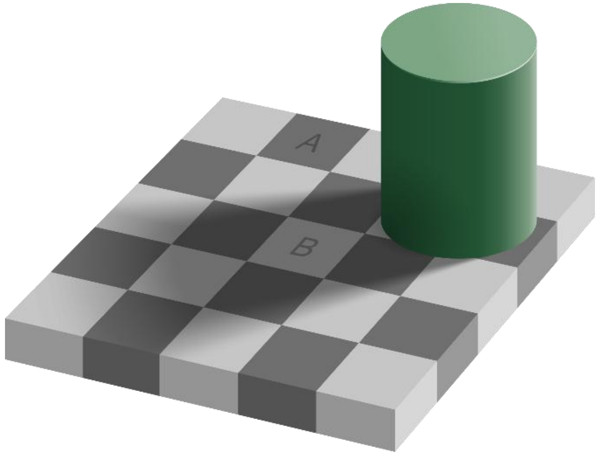
CIE 1931 RGB color matching functions in real ratio (dashed), and resulting luminance (Y) curve (black)

Why is our Brain so efficient?

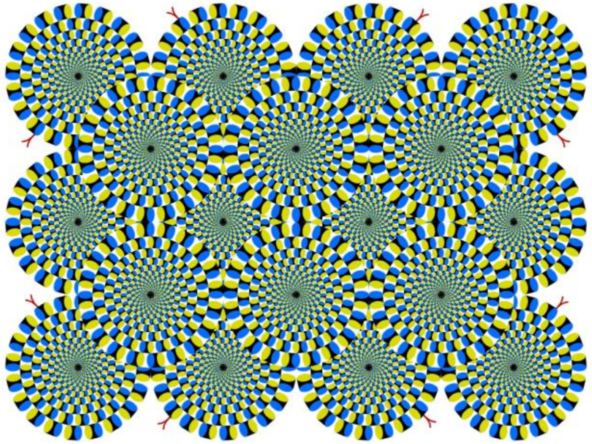
Likely because we are making ***prior*** assumptions about the world regarding depth, color, shadows, motion, etc.



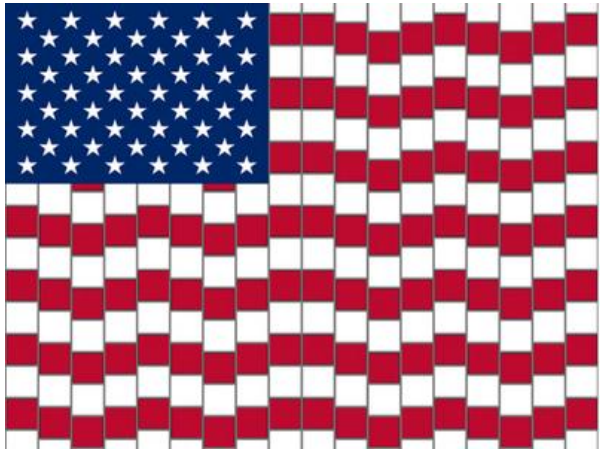
Shepherd Tabletop Illusion



Anderson Shadow Illusion



Spinning Wheels Illusion



Crooked Lines Illusion

Course Overview

CW	Topic	Date	Place	Lab
41	Introduction and Course Overview	07.10.2025	Zoom	Lab 1
42	Capturing Digital Images	14.10.2025	Zoom	Lab 2
43	Digital Image Processing	21.10.2025	Zoom	Assignment 1
44	Machine Learning	28.10.2025	Zoom	
45	Feature Extraction	04.11.2025	Zoom	Open Lab 1
46	Segmentation	11.11.2025	Zoom	Assignment 2
47	Optical Flow	18.11.2025	Zoom	Open Lab 2
48	Object Detection	25.11.2025	Zoom	Assignment 3
49	Multi-View Geometry	02.12.2025	Zoom	Open Lab 3
50	3D Vision	09.12.2025	Zoom	Assignment 4
3	Trends in Computer Vision	13.01.2026	Zoom	
4	Q&A	20.01.2026	Zoom	Open Lab 4
5	Exam	27.01.2026	HS1 (Linz), S1/S3 (Vienna), S5 (Bregenz)	
9	Retry Exam	24.02.2026	tba	

Next Week: Digital Image Processing

Convolution

$$y[m,n] = x * h = \sum_{k,l=-N}^{N} x[m+k,n+l]h[-k,-l]$$

0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	0	1	1	1	0	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0

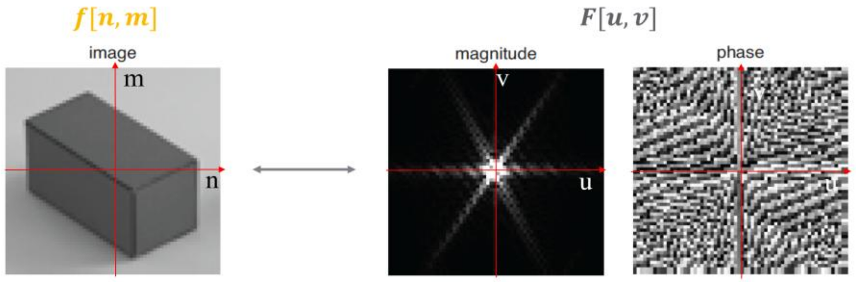
l			
1	-1	-2	-1
0	0	0	0
-1	1	2	1
k→	-1	0	1

In this case, N = 1.

Convolution is a shift-invariant linear Operation.

JOHANNES KEPLER UNIVERSITY LINZ

Spatial vs. Frequency Domain



Fourier Transformation /
Inverse Fourier Transformation

JOHANNES KEPLER UNIVERSITY LINZ

Spatial vs. Gradient Domain

Laplacian!

$$\mathrm{div}(\nabla I) = \frac{\partial I_x}{\partial x} + \frac{\partial I_y}{\partial y} = I_{xx} + I_{yy} = \nabla^2 I$$

$$\mathrm{curl}(\nabla I) = \frac{\partial I_y}{\partial x} - \frac{\partial I_x}{\partial y} = I_{yx} - I_{xy}$$

Gradient Domain

Image gradient:

$$\nabla I = \left(\underset{I_x}{\frac{\partial I}{\partial x}}, \underset{I_y}{\frac{\partial I}{\partial y}}\right)$$

I_x I_y We can consider this as Coefficients of a Vectorfield.

Derivative / Integral

Processing the Vectorfield is called Gradient Domain Processing.

Spatial Domain

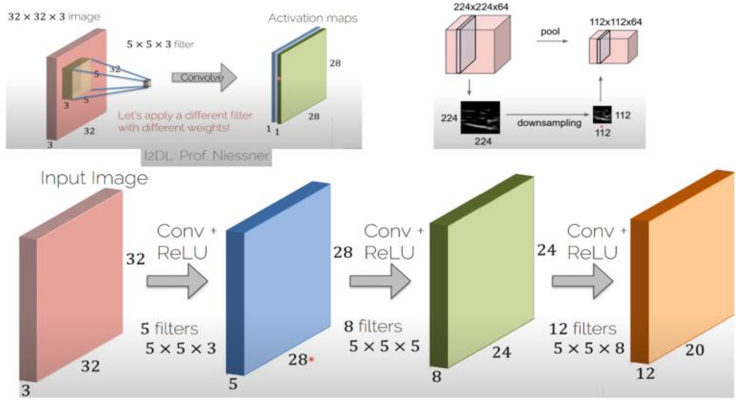
$$I = \int \nabla I$$

Integrating this Vectorfield results in the original Image as long as the Vectorfield has zero Curl. **In Practice: that does not work as the Vectorfields is no longer conservative anymore after processing it (it has non-zero Curl).**

JOHANNES KEPLER UNIVERSITY LINZ

Instead: solve 2D Poisson Equation: $\nabla^2 I = \mathrm{div}(G)$

Convolutional Neural Networks (CNNs)



JOHANNES KEPLER UNIVERSITY LINZ

Thank You