Robot Vision

Image Formation & Camera Models

Dr. Chen Feng

cfeng@nyu.edu

ROB-UY 3203, Spring 2024

Overview

- Digital camera & image
  - Camera selection
- Color transformations (RGB, Grey, HSV)
- Pinhole camera model
  - 3D rotation representations (Euler angle, axis-angle, quaternion)
- Lens distortion

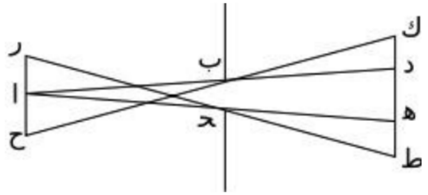
- Homogeneous coordinates
- Geometric transformations

References

- Forsyth & Ponce 2011:
  - Chapter 1; Section 3.1, 3.4, 3.5
- Szeliski 2022:
  - Section 2.1, 2.3, 2.3.2, 6.3.5
- Corke 2011:
  - Chapter 2; Section 10.2, 11.1
- Hartley & Zisserman 2003:
  - Section 2.2, 2.3, 2.4, 3.1, 3.4, 6.1
- Linear algebra:
  - Szeliski 2011: section A.1.1, A.1.2, A.1.3, A.2, A.2.1
  - Hartley & Zisserman 2003: A5.1, A5.2, A5.3

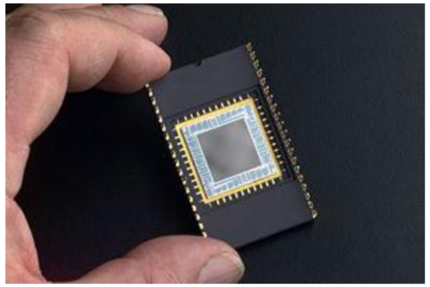
Historic Milestones of Camera

- **Pinhole model:** Mozi (470-390 BCE), Aristotle (384-322 BCE)
- **Principles of optics (including lenses):** Alhacen (965-1039 CE)
- **Camera obscura:** Leonardo da Vinci (1452-1519), Johann Zahn (1631-1707)
- **First photo:** Joseph Nicephore Niepce (1822)
- **Daguerréotypes** (1839)
- **Photographic film** (Eastman, 1889)
- **Cinema** (Lumière Brothers, 1895)
- **Color Photography** (Lumière Brothers, 1908)
- **Television** (Baird, Farnsworth, Zworykin, 1920s)
- **First consumer camera with CCD**: Sony Mavica (1981)
- **First fully digital camera:** Kodak DCS100 (1990)



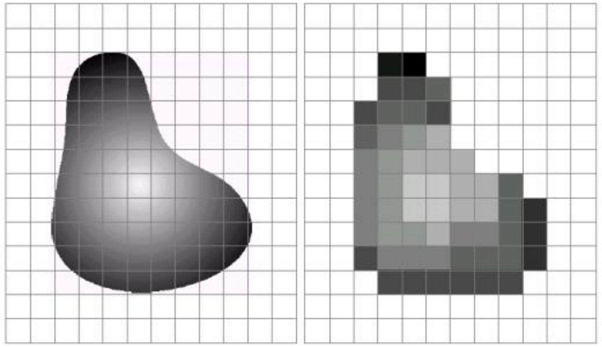
Alhacen's notes


Niepce, "La Table Servie," 1822


CCD chip

Slide by Rob Fergus

Digital Camera



a b

FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.

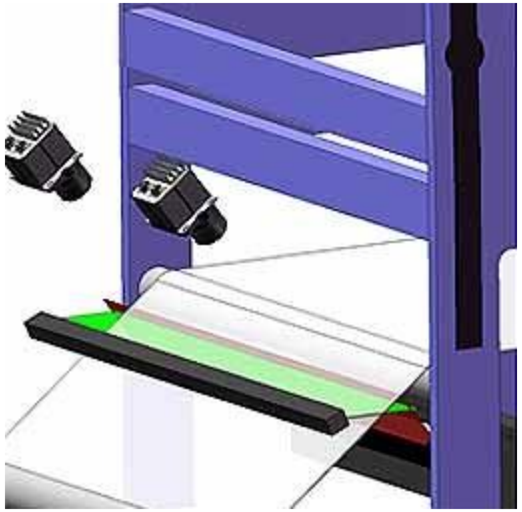
- A digital camera replaces film with a sensor array
  - Each cell in the array is light-sensitive diode that converts photons to electrons
  - Two common types
    - Charge Coupled Device (CCD)
    - Complementary metal oxide semiconductor (CMOS)

Slide by Steve Seitz

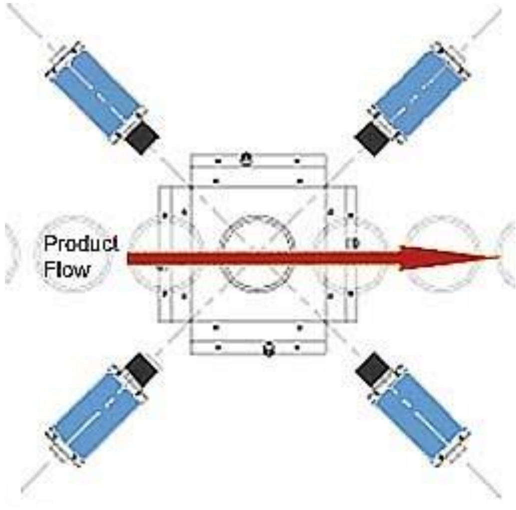
Choosing Machine Vision Cameras

- Area scan or Line scan?
- Color or Monochrome?
- CMOS or CCD?
- Global or Rolling Shutter?
- Frame Rate?
- Resolution?
- Connection Interface?
- Lens Focal length

Area Scan or Line Scan?



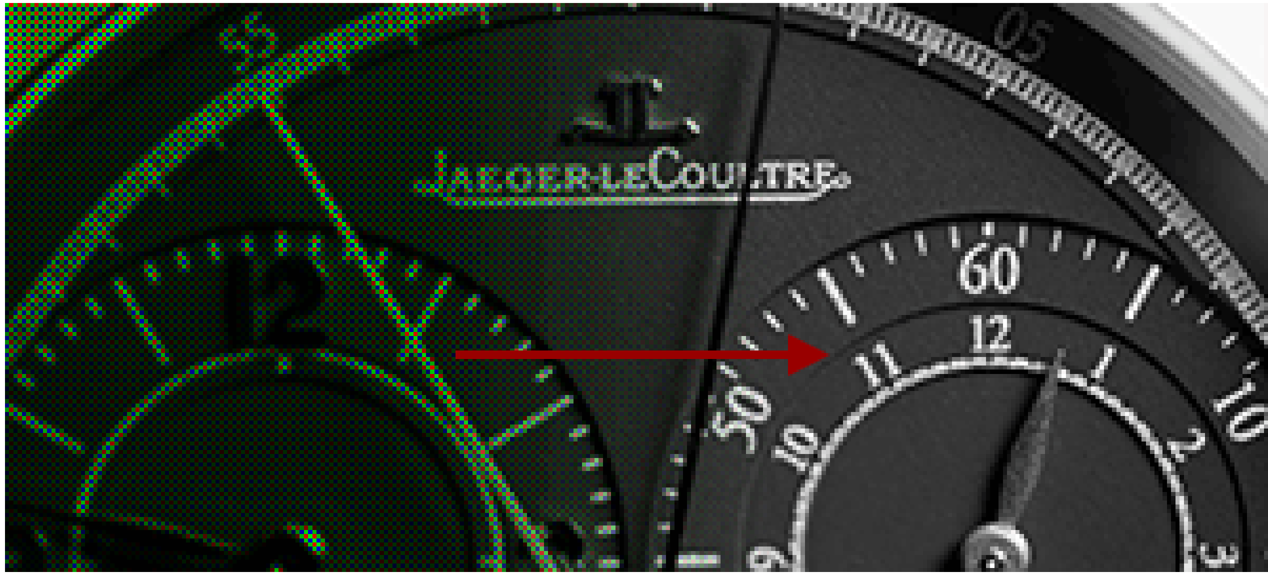
Line scan

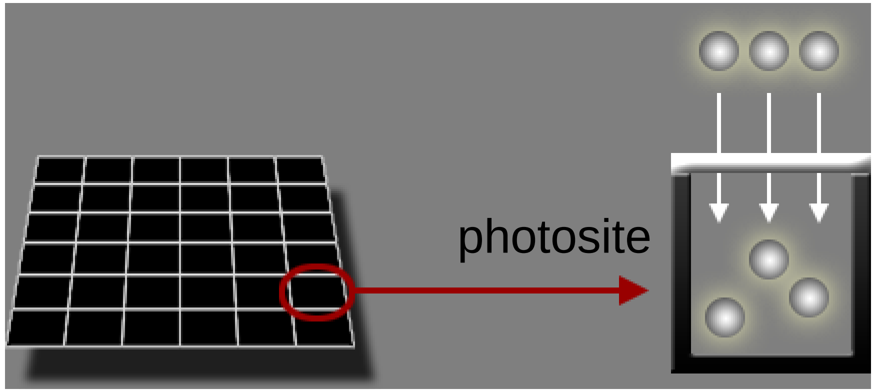


Area scan

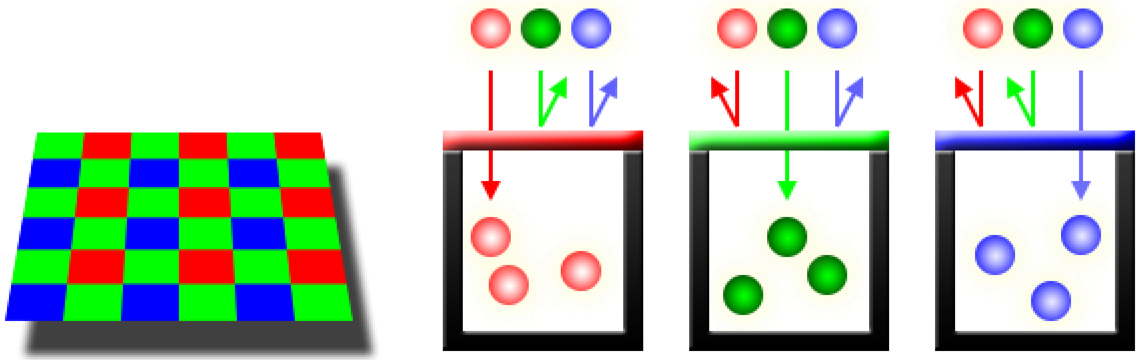
Images from https://newtonlabs.com/line_systems.htm

Color or Monochrome?





Monochrome sensor



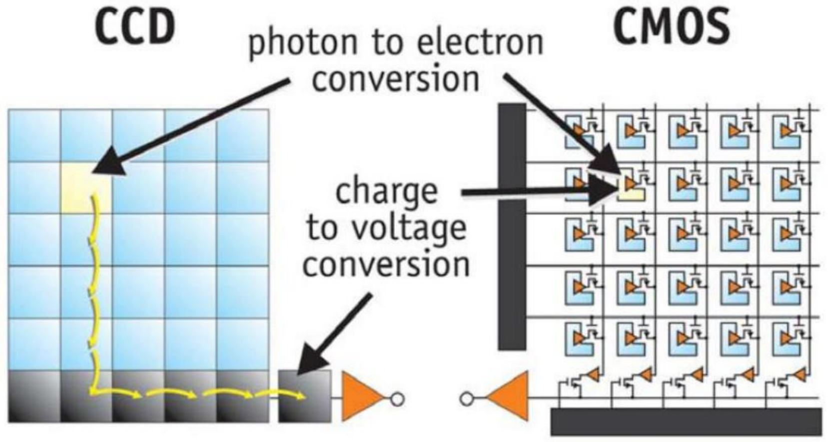
Color sensor (with color filter array)

Images from http://www.red.com/learn/red-101/color-monochrome-camera-sensors

CMOS or CCD?

- CCD: charge coupled device
- CMOS: complementary metal oxide semiconductor
- Pros/Cons
  - Noise: CMOS > CCD
  - Light sensitivity: CMOS < CCD
  - Pixel quality: CMOS < CCD
  - Power consumption: CMOS < CCD
  - Speed/frame-rate: CMOS > CCD
  - Price: CMOS < CCD
- Current winner: CMOS

http://electronics.howstuffworks.com/digital-camera.htm

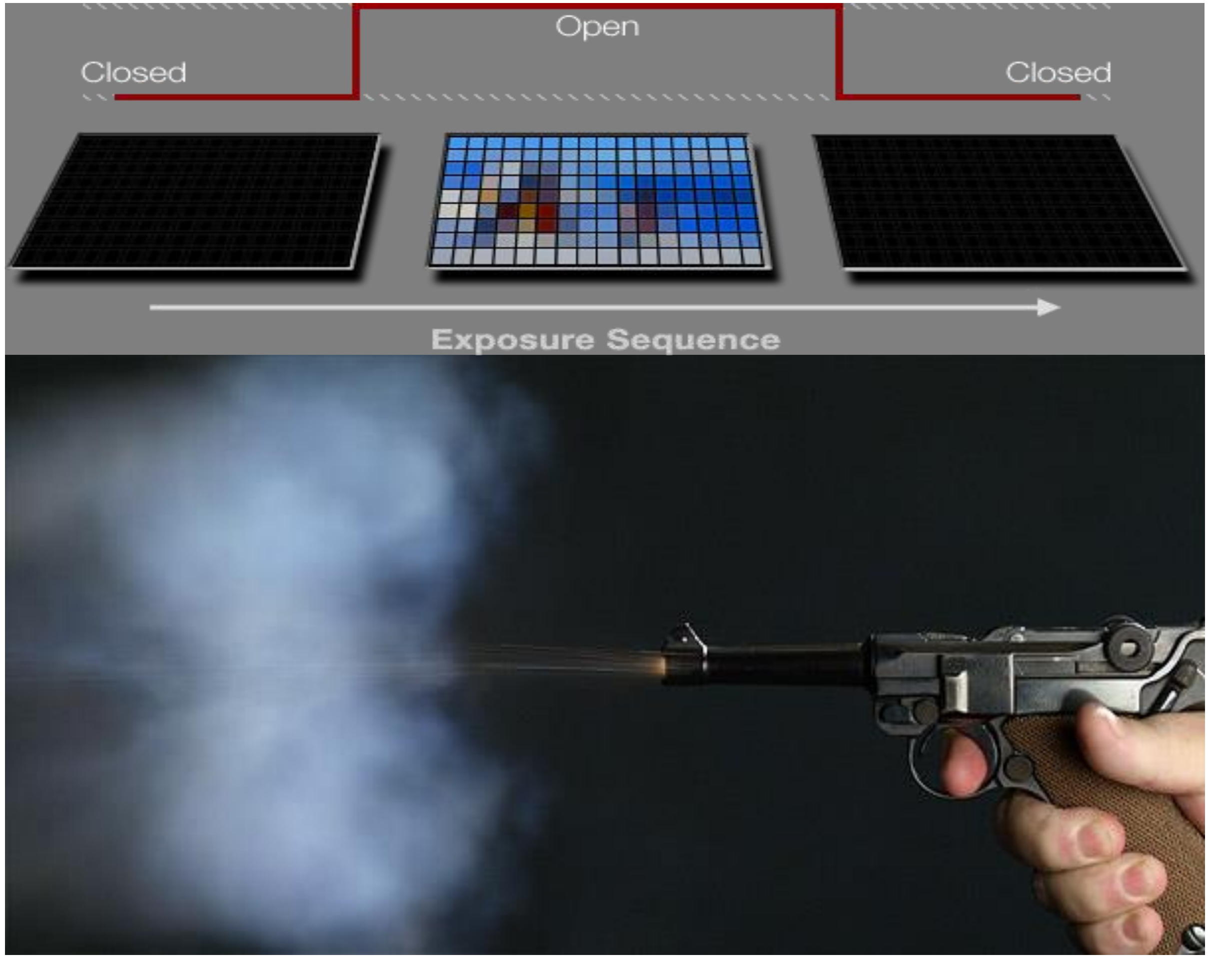


CCDs move photogenerated charge from pixel to pixel and convert it to voltage at an output node. CMOS imagers convert charge to voltage inside each pixel.

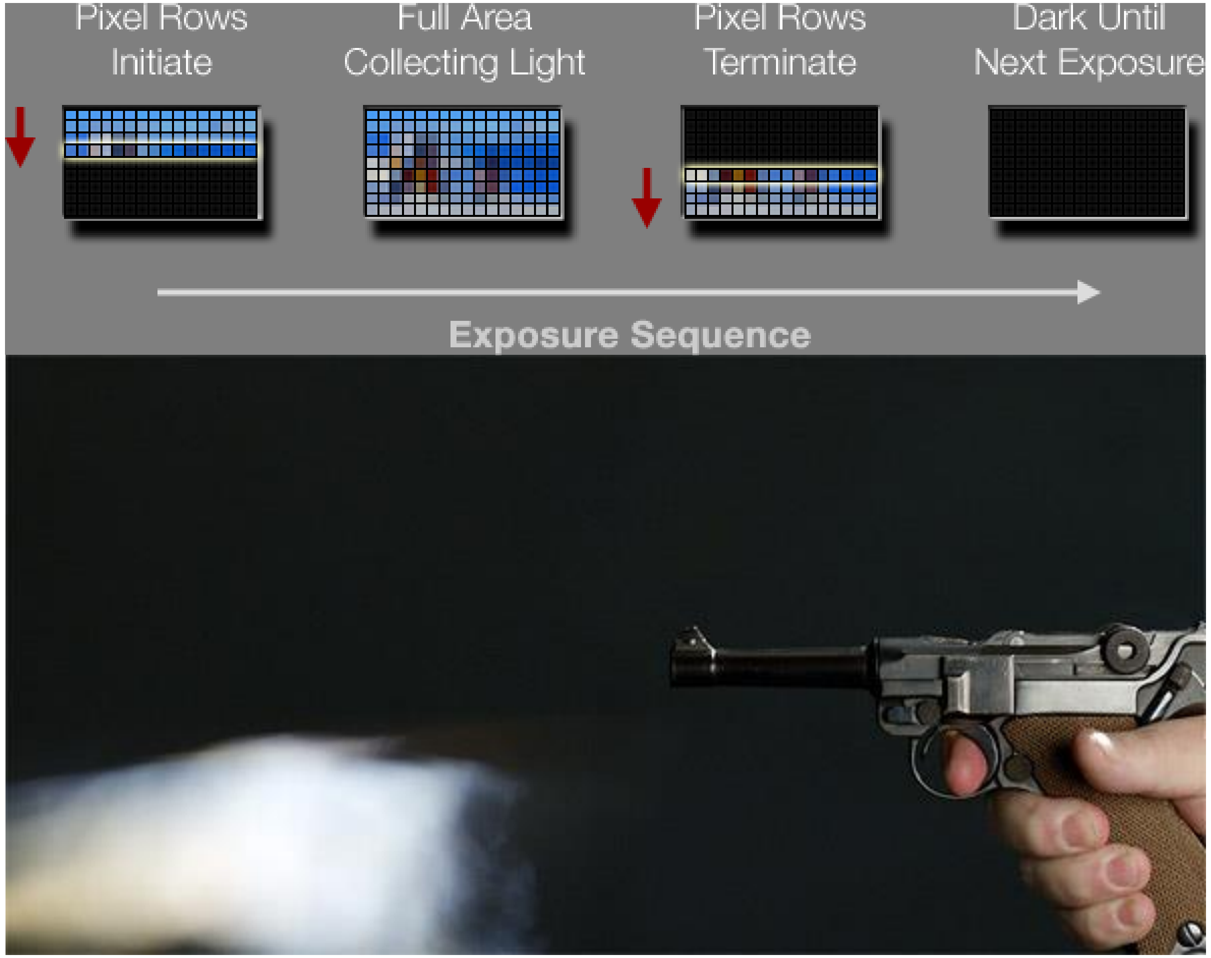
http://www.dalsa.com/shared/content/pdfs/CCD_vs_CMOS_Litwiller_2005.pdf

Image from Rob Fergus

Global or Rolling Shutter



Global Shutter



Rolling Shutter

Image from http://www.red.com/learn/red-101/global-rolling-shutter

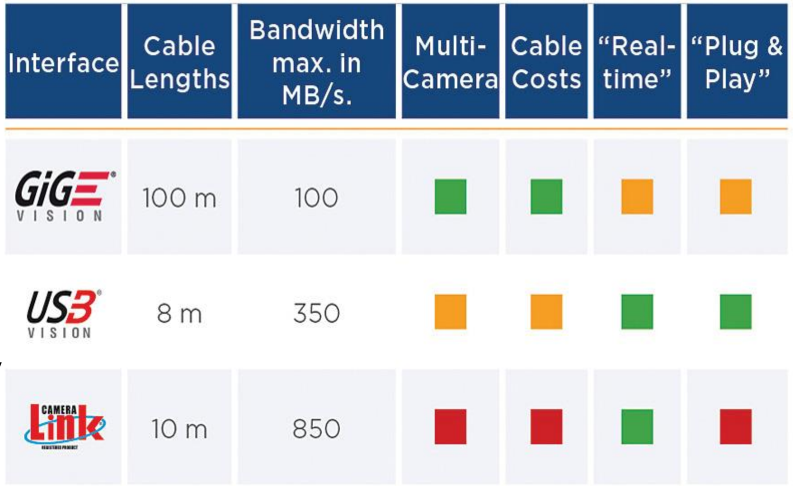
Effect of Rolling Shutter with Fast Motion

- Is the photographer moving towards left, or right?
  - Assuming rolling shutter moves from the top to bottom of the image

Image from https://www.bhphotovideo.com/explora/video/tips-and-solutions/rolling-shutter-versus-global-shutter

Other Factors

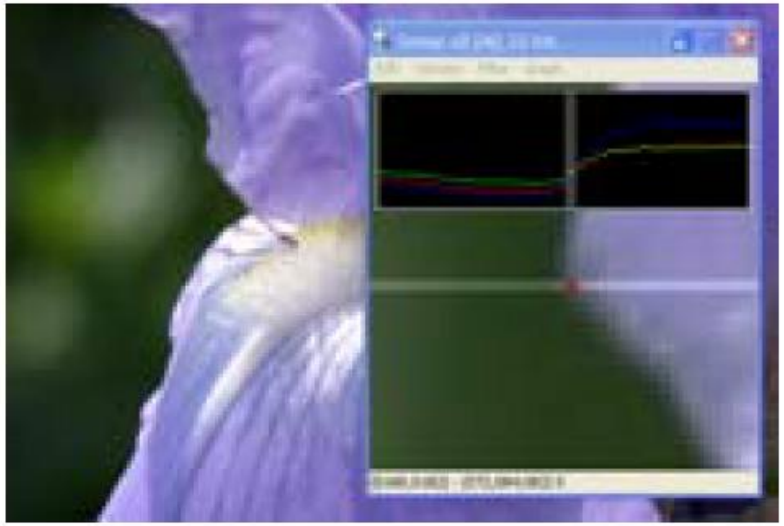
- Resolution
  - QVGA (320x240), VGA (640x480)
  - Choose based on your application!
- Frame rate
  - 15Hz, 30Hz, 60Hz
- Connection interface
  - USB3.0: 350MB/s, up to 8m cable (power + data), plug-and-play
  - GigE: 100MB/s, up to 100m cable, good for multiple cameras
- Focal length
  - Fixed: good for geometric vision (3D reconstruction/pose estimation)
  - Auto-focus



Interface	Cable Lengths	Bandwidth max. in MB/s.	Multi-Camera	Cable Costs	"Real-time"	"Plug & Play"
GiGE VISION	100 m	100				
USB3 VISION	8 m	350				
CAMERA Link	10 m	850				

Image: https://www.qualitymag.com/articles/93861-how-to-choose-a-machine-vision-camera

Digital Image



45	60	98	127	132	133	137	133
46	65	98	123	126	128	131	133
47	65	96	115	119	123	135	137
47	63	91	107	113	122	138	134
50	59	80	97	110	123	133	134
49	53	68	83	97	113	128	133
50	50	58	70	84	102	116	126
50	50	52	58	69	86	101	120

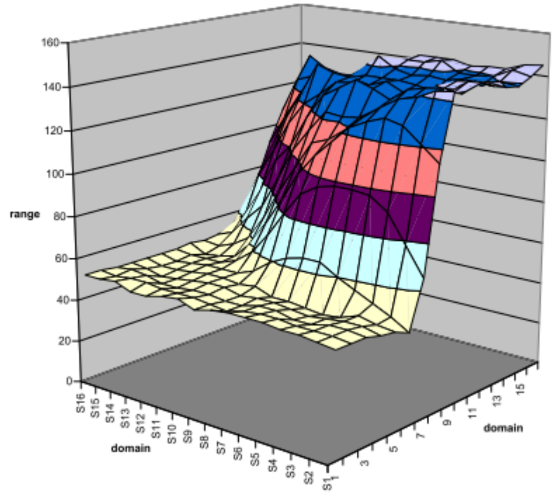


Image from Szeliski 2011

Digital Image

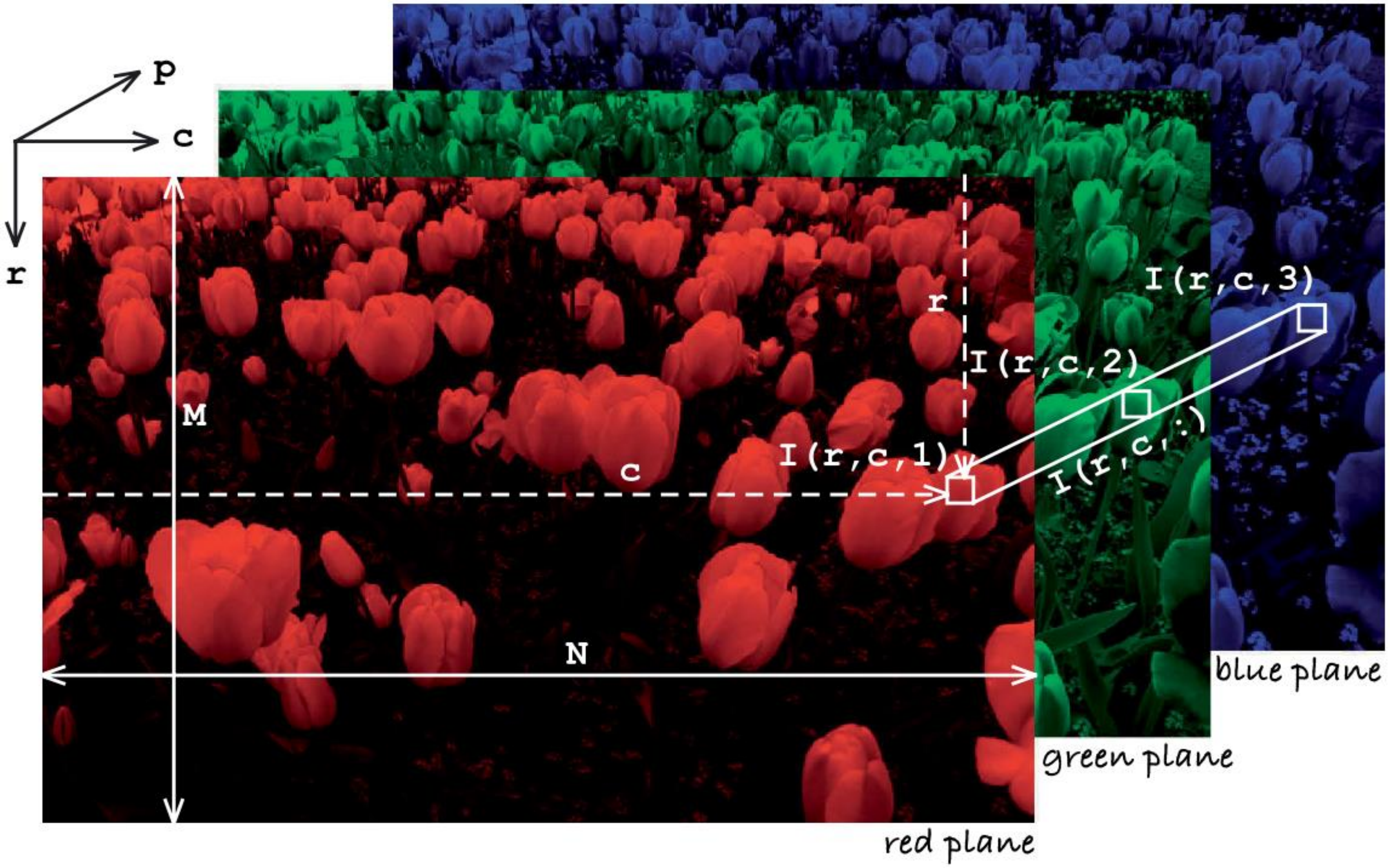


Image from Corke 2011

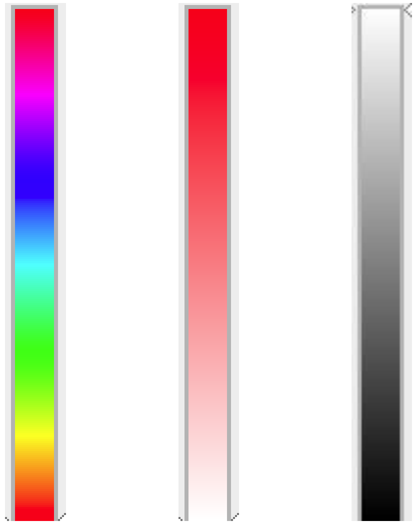
Color Space Transformations

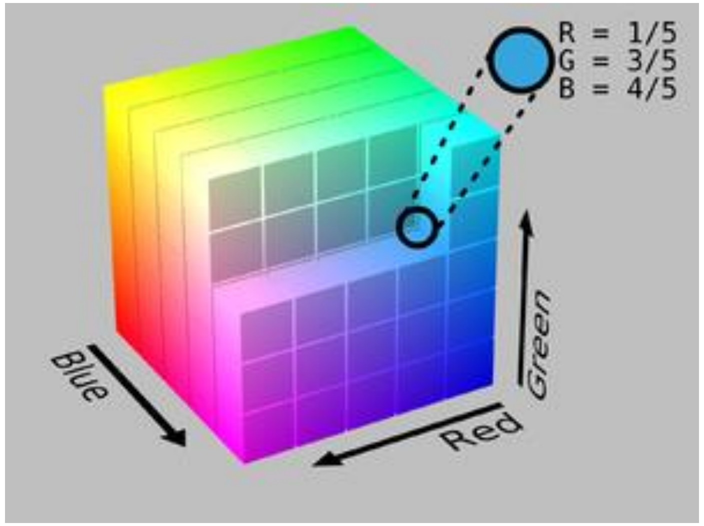
- RGB2Gray

$$\text{RGB[A] to Gray:} \quad Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

$$0.2126R + 0.7152G + 0.0722B$$

- RGB2HSV





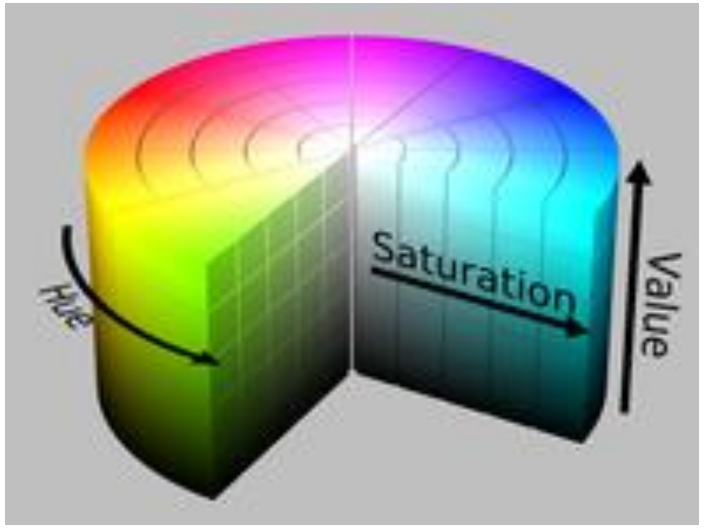
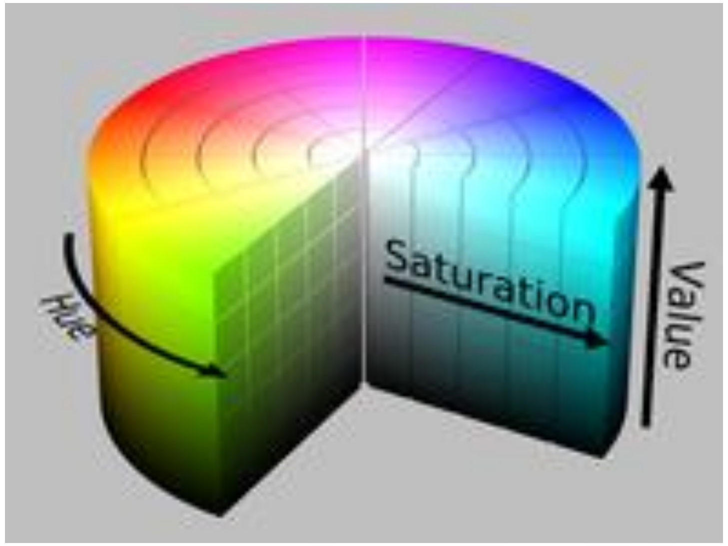


Image from Wiki & https://graphicdesign.stackexchange.com/a/16286

Color Space Transformations



- Higher values are shown as white pixels
- Lower values are shown as darker pixels

- Which image represents the Hue channel, b or c?

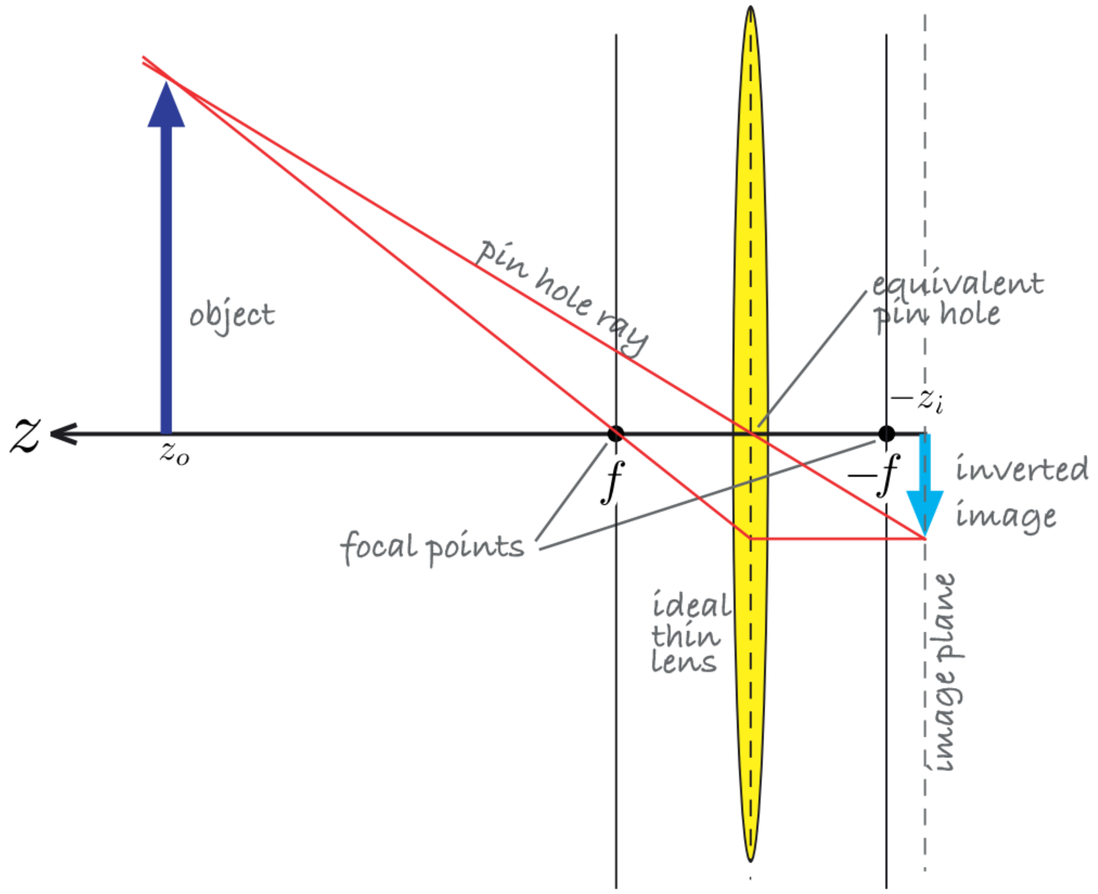
Hue Saturation

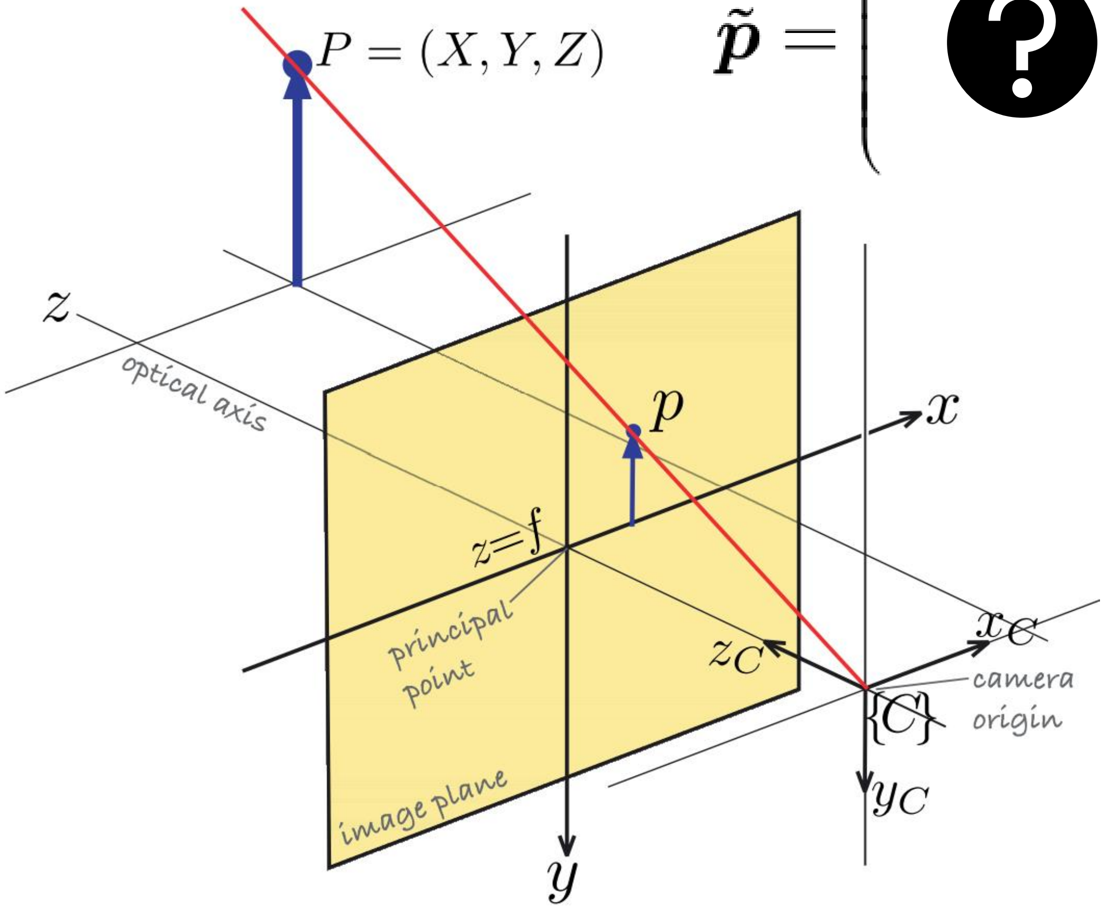
Image from Corke 2011

Pinhole Camera

Image from http://thedelightsofseeing.blogspot.com/2010/10/pinhole-photography-and-camera-obscura.html

Evolution of Pinhole Camera Model

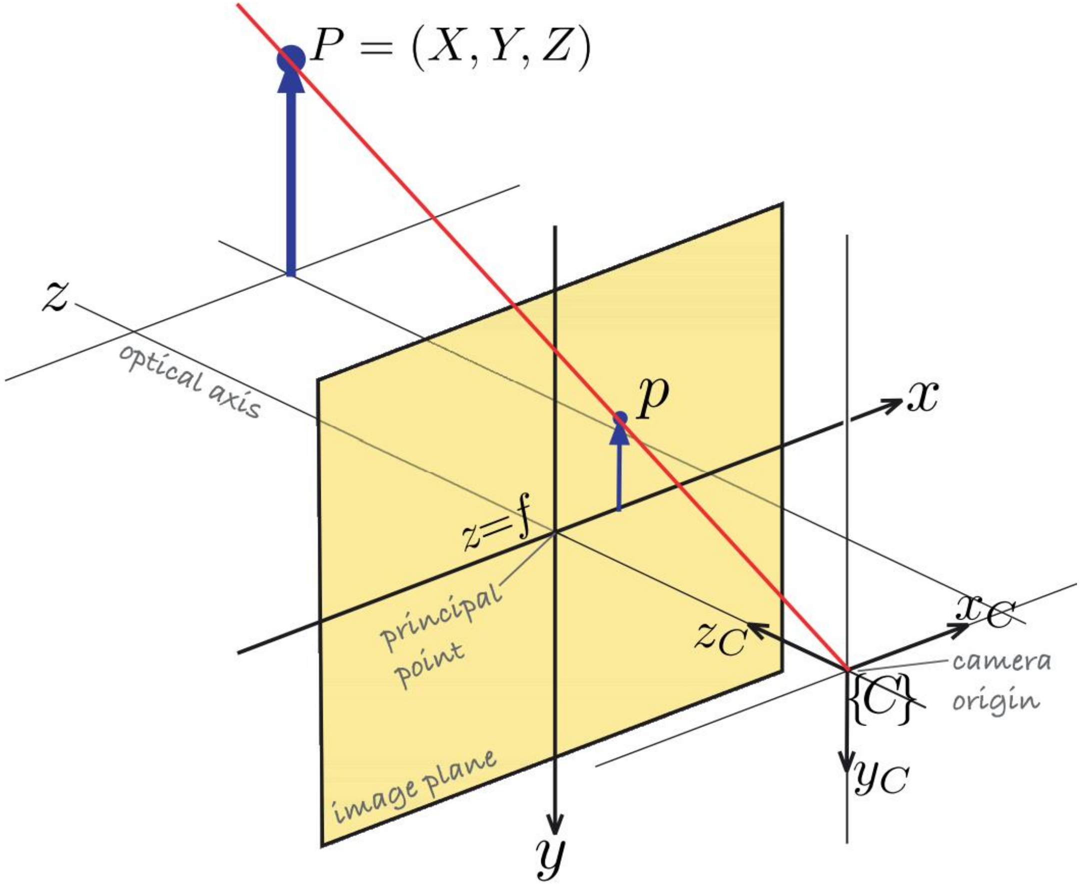




$$\tilde{\boldsymbol{p}} = \Big(\; ? \; \Big) \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

Image from Corke 2011

Basic Equation of Pinhole Camera Model



Camera Calibration Matrix $\boldsymbol{K}$

Image Point → $$\boldsymbol{p} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$ ← Object Point

$$= \boldsymbol{K}P$$

Image from Corke 2011

Add Digitization/Discretization/Pixelization

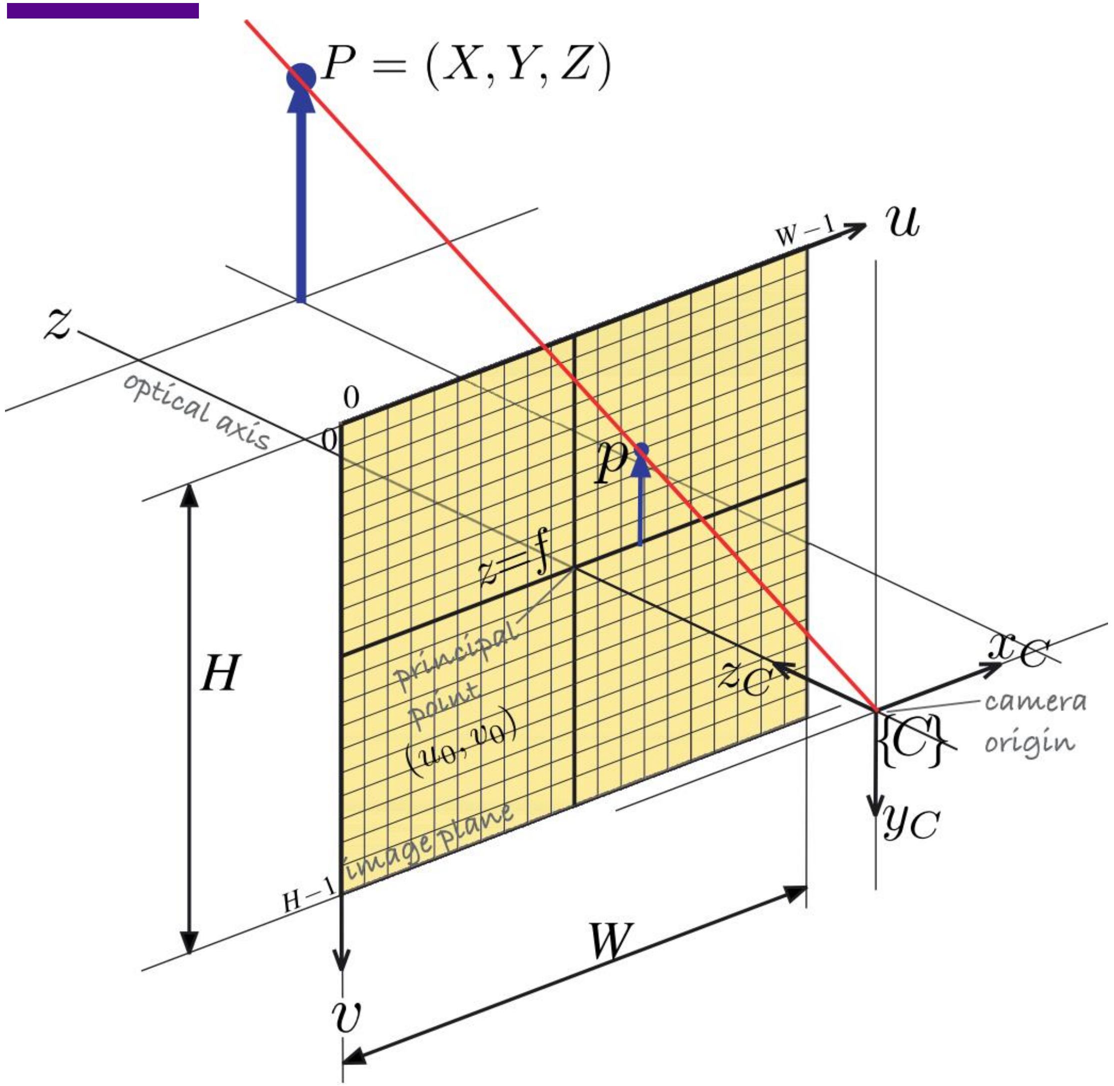


Image from Corke 2011

ρ_w and ρ_h are the width and height of each pixel

$$\boldsymbol{p} = \underbrace{\begin{pmatrix} 1/\rho_w & 0 & u_0 \\ 0 & 1/\rho_h & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{K}} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

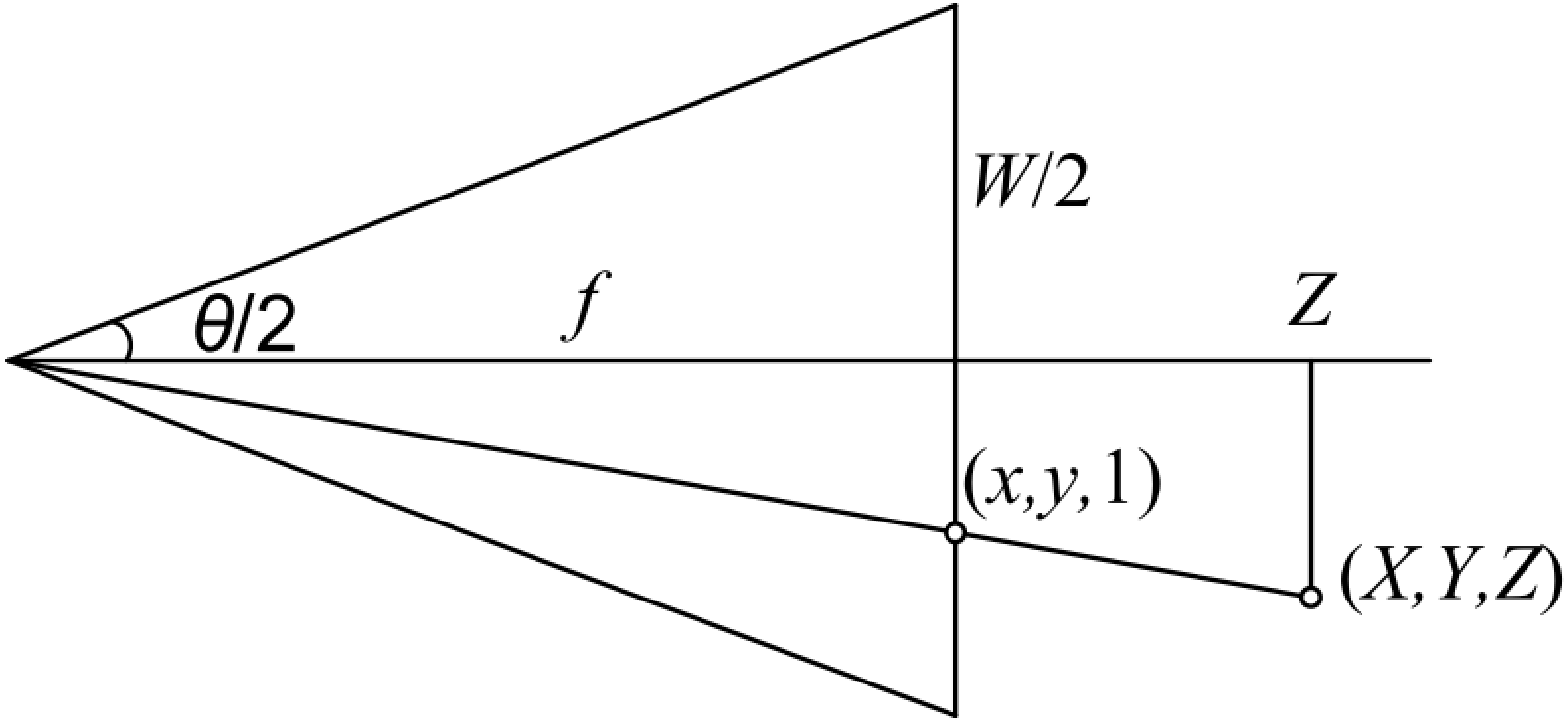
$$\boldsymbol{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Various Modeling Complexities of K

Skew of un-orthogonal u- and v- axis

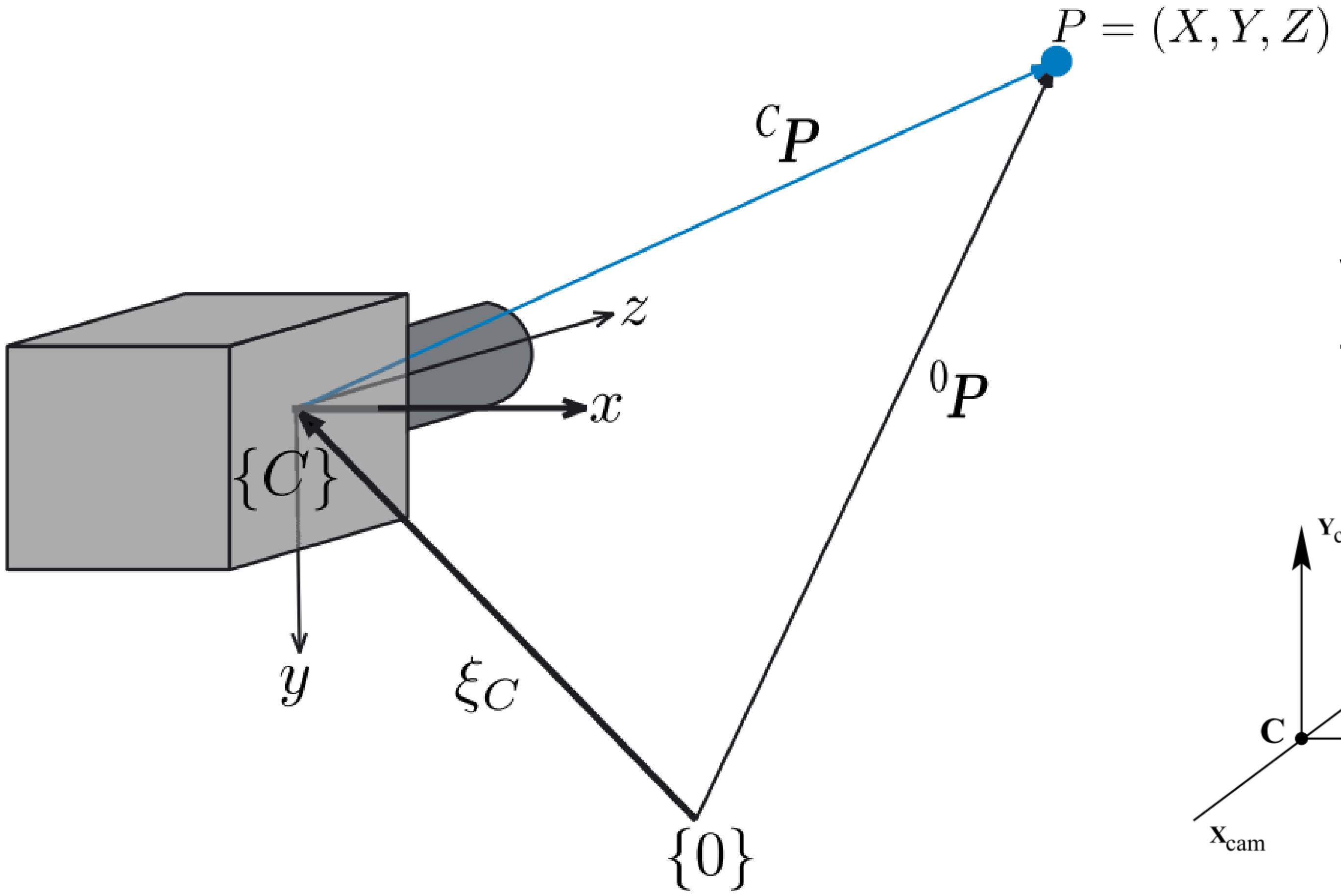
$$\boldsymbol{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{K} = \begin{bmatrix} f & s & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

The Unit of Focal Length: Depends on Image Coordinate's Unit



$$\tan\frac{\theta}{2} = \frac{W}{2f} \quad \text{or} \quad f = \frac{W}{2}\left[\tan\frac{\theta}{2}\right]^{-1}$$

Camera Coordinate Frames: Moving Camera Around



We need to deal with 3D translations and rotations!

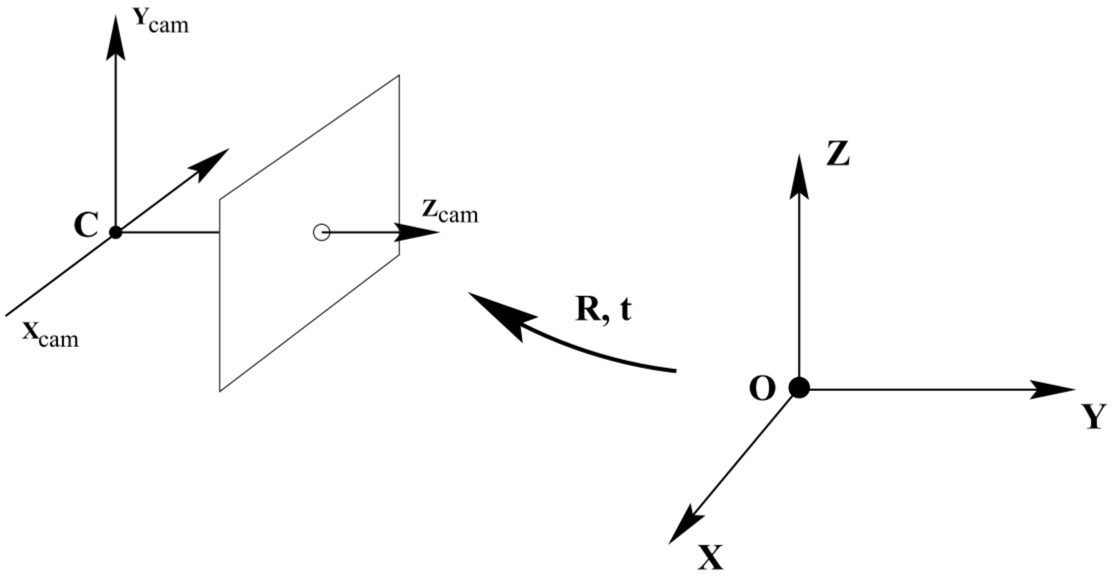
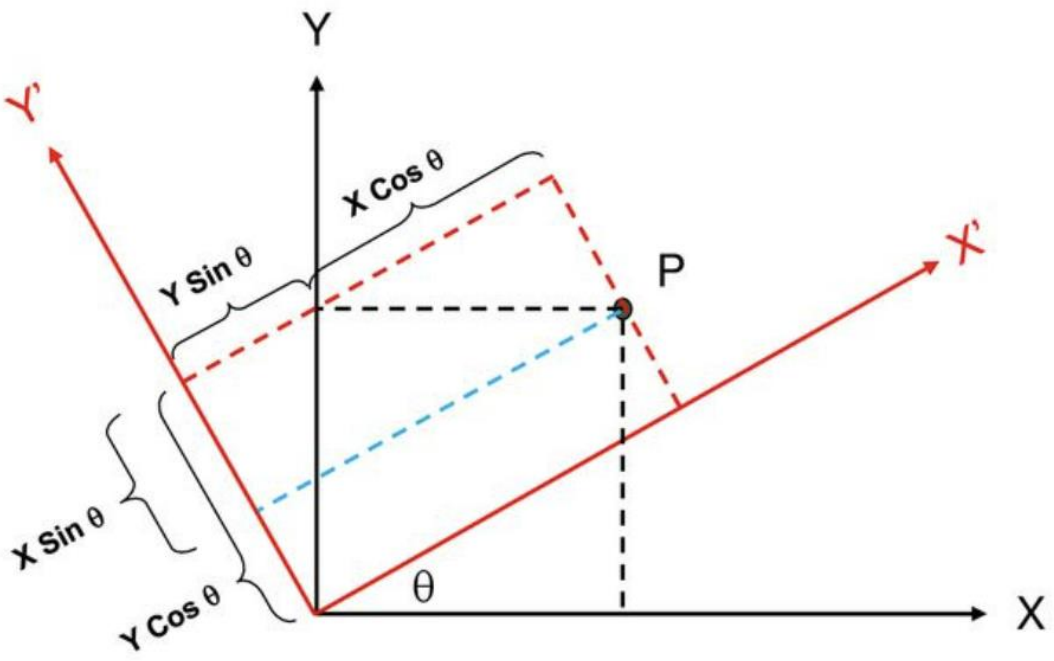


Image from Corke 2011

2D Rotation



X' = X Cos θ + Y Sin θ

Y' = Y Cos θ – X Sin θ

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Rotation Matrix

$${}^{B}\boldsymbol{R}_{A}$$

3D Rotation Representation: Rotation Matrix

- An orthonormal matrix
  - A real square matrix
  - whose columns & rows are orthonormal vectors
  - $\det(R) = 1$
  - $R^T = R^{-1}$
- Basic rotation around X/Y/Z axis →

$$
\begin{cases}
R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \\
R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \\
R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{cases}
$$

3D Rotation Representation: Euler Angles

- 3 successive rotations

- Eulerian: with repetition in axis
  - XYX, XZX, YXY, YZY, ZXZ, ZYZ
- Cardanian: without repetition
  - XYZ, XZY, YZX, YXZ, ZXY, ZYX

$$\boldsymbol{R} = \boldsymbol{R}_z(\phi)\boldsymbol{R}_y(\theta)\boldsymbol{R}_z(\psi)$$

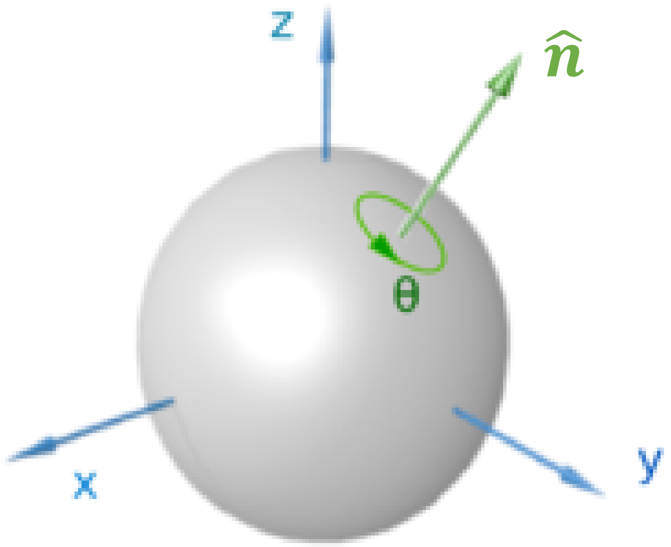
- 12 types
  - And you should always check handedness, intrinsic or extrinsic rotation

3D Rotation Representation: Axis-Angle Vector

- A 3D vector $\theta\hat{\boldsymbol{n}} \in \mathbf{R}^3$
  - aka, rotation vector

- Conversion to rotation matrix by Rodrigues' rotation formula:

$$\boldsymbol{R}(\hat{\boldsymbol{n}}, \theta) = \boldsymbol{I} + \sin\theta[\hat{\boldsymbol{n}}]_\times + (1 - \cos\theta)[\hat{\boldsymbol{n}}]_\times^2$$



$$[\hat{\boldsymbol{n}}]_\times = \begin{bmatrix} 0 & -\hat{n}_z & \hat{n}_y \\ \hat{n}_z & 0 & -\hat{n}_x \\ -\hat{n}_y & \hat{n}_x & 0 \end{bmatrix}$$

3D Rotation Representation: Unit Quaternion

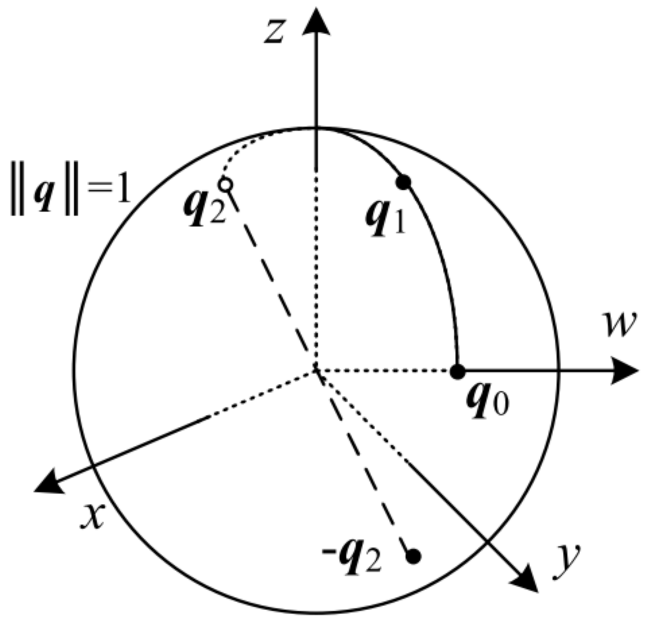
- A 4D unit vector

$$\boldsymbol{q} = (\boldsymbol{v}, w) = (\sin\frac{\theta}{2}\hat{\boldsymbol{n}}, \cos\frac{\theta}{2})$$

$$\boldsymbol{q} = (x, y, z, w)$$

- Conversion to rotation matrix:

$$\boldsymbol{R}(\boldsymbol{q}) = \begin{bmatrix} 1-2(y^2+z^2) & 2(xy-zw) & 2(xz+yw) \\ 2(xy+zw) & 1-2(x^2+z^2) & 2(yz-xw) \\ 2(xz-yw) & 2(yz+xw) & 1-2(x^2+y^2) \end{bmatrix}$$



Summary of 3D Rotation Representation

- Rotation matrix (9 parameters)
- Euler (3 parameters)
  - Minimal
  - Many ambiguities
  - Gimbal Lock
- Axis-angle (3 parameters)
  - Minimal & continuous
  - 1 rotation ~ 2 axis-angle vectors
- Unit quaternion (4 parameters)
  - Almost minimal & continuous
  - 1 rotation ~ 2 unit quaternions
  - Easy inverse/multiplication
  - Interpolation using *slerp*

procedure *slerp*$(\boldsymbol{q}_0, \boldsymbol{q}_1, \alpha)$:

1. $\boldsymbol{q}_r = \boldsymbol{q}_1/\boldsymbol{q}_0 = (\boldsymbol{v}_r, w_r)$
2. if $w_r < 0$ then $\boldsymbol{q}_r \leftarrow -\boldsymbol{q}_r$
3. $\theta_r = 2\tan^{-1}(\|\boldsymbol{v}_r\|/w_r)$
4. $\hat{\boldsymbol{n}}_r = \mathcal{N}(\boldsymbol{v}_r) = \boldsymbol{v}_r/\|\boldsymbol{v}_r\|$
5. $\theta_\alpha = \alpha\,\theta_r$
6. $\boldsymbol{q}_\alpha = (\sin\frac{\theta_\alpha}{2}\hat{\boldsymbol{n}}_r, \cos\frac{\theta_\alpha}{2})$
7. **return** $\boldsymbol{q}_2 = \boldsymbol{q}_\alpha \boldsymbol{q}_0$

Recall the Basic Equation of Pinhole Camera Model

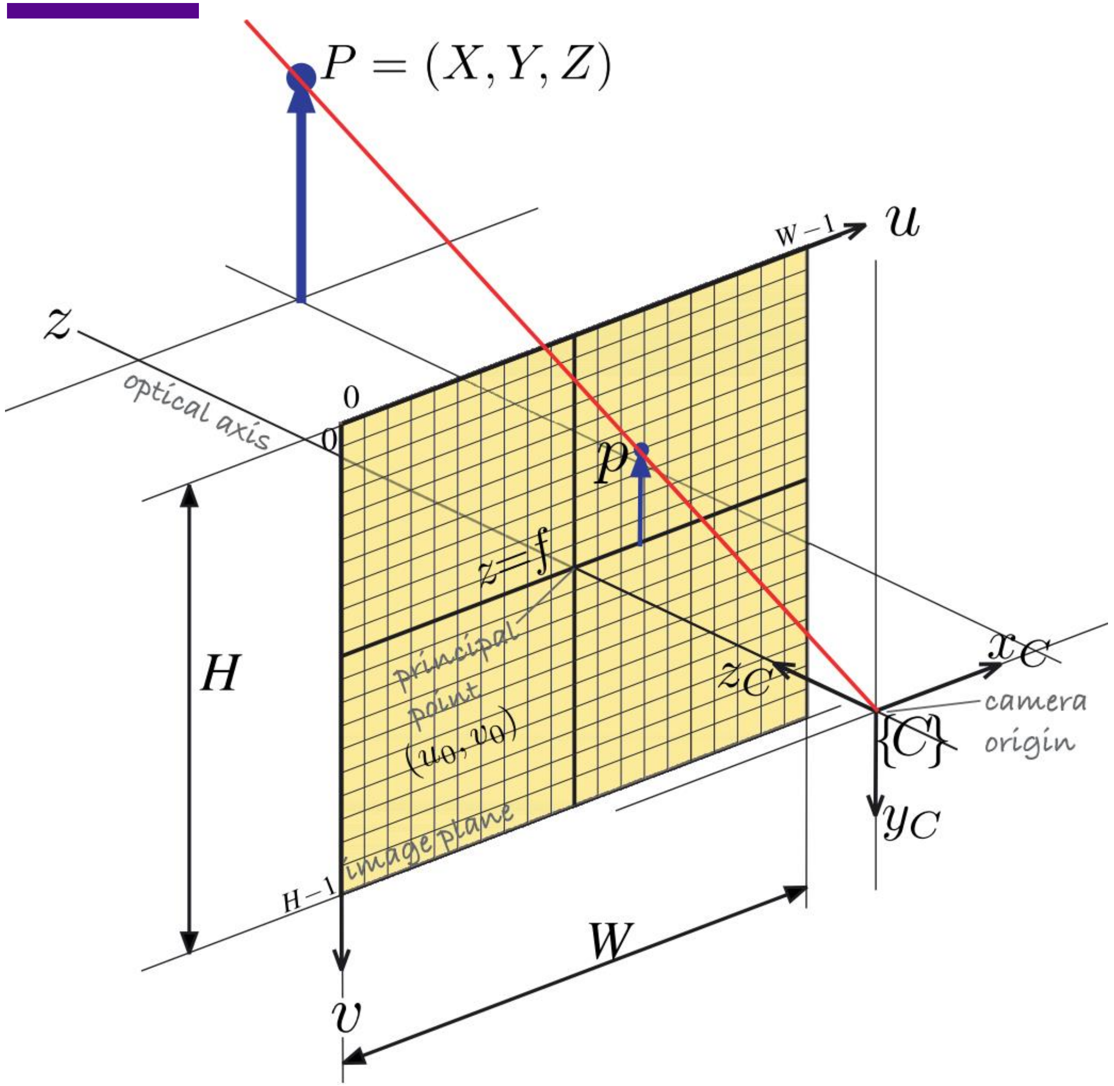


Image from Corke 2011

ρ_w and ρ_h are the width and height of each pixel

$$\boldsymbol{p} = \underbrace{\begin{pmatrix} 1/\rho_w & 0 & u_0 \\ 0 & 1/\rho_h & v_0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{K}}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

$$\boldsymbol{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Updated Equation of Pinhole Camera Model

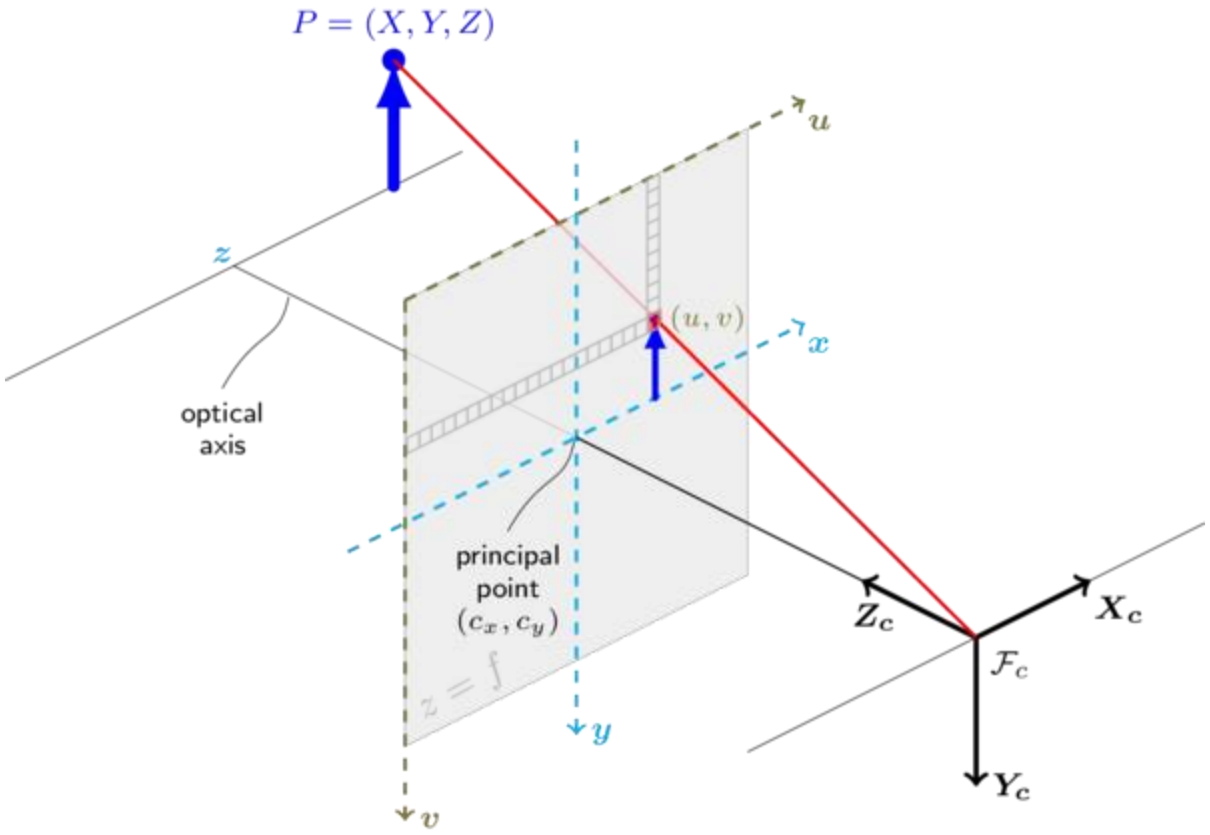
- Putting intrinsics & extrinsics together:

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \overbrace{\underbrace{\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinsics } \boldsymbol{K}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}}_{\text{extrinsics } \boldsymbol{R},\boldsymbol{t}}}^{\text{Camera Matrix } \boldsymbol{P}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

In OpenCV: $\boldsymbol{K}\left[\ \boldsymbol{R} \mid \boldsymbol{t}\ \right]$

Image from Corke 2011

The Pinhole Camera Model Calculation Process in OpenCV



$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

$$x' = x/z$$

$$y' = y/z$$

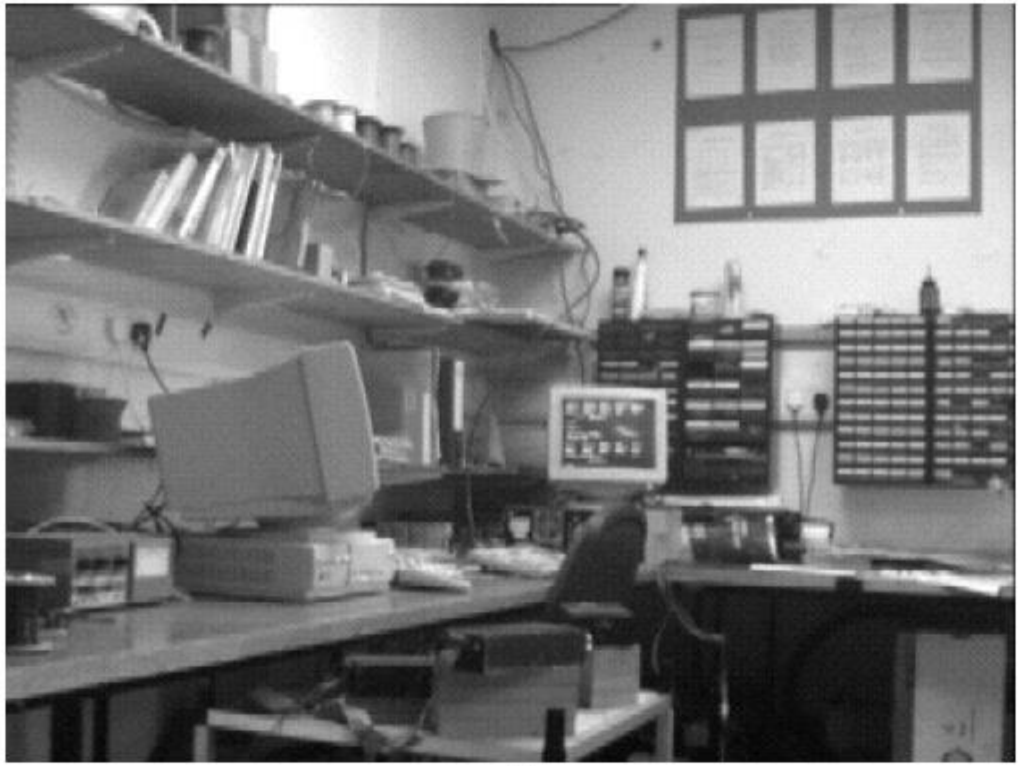
$$u = f_x * x' + c_x$$

$$v = f_y * y' + c_y$$

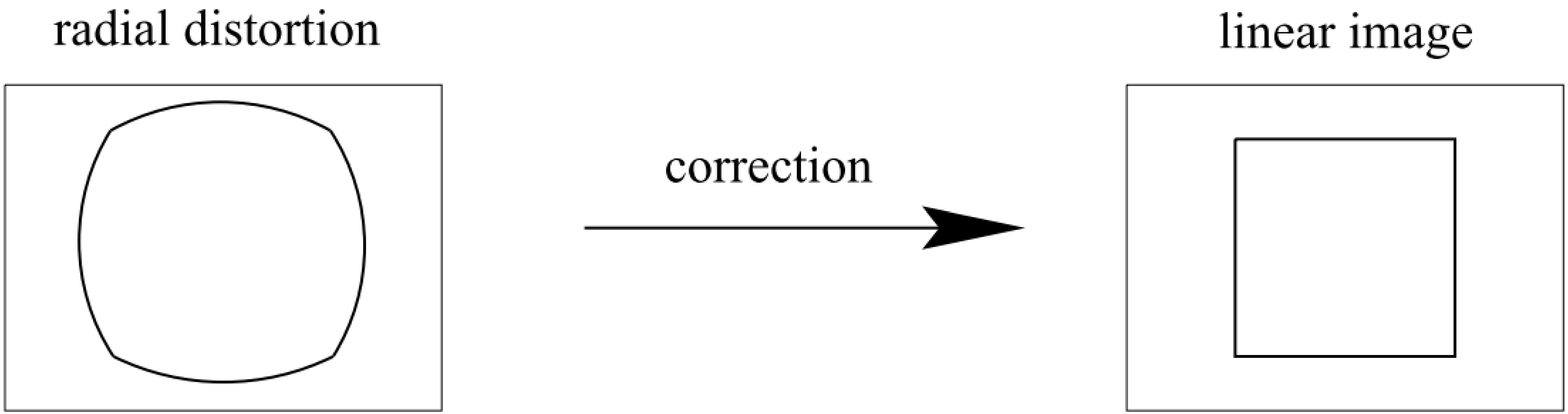
https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

Lens Distortion

- Radial distortion



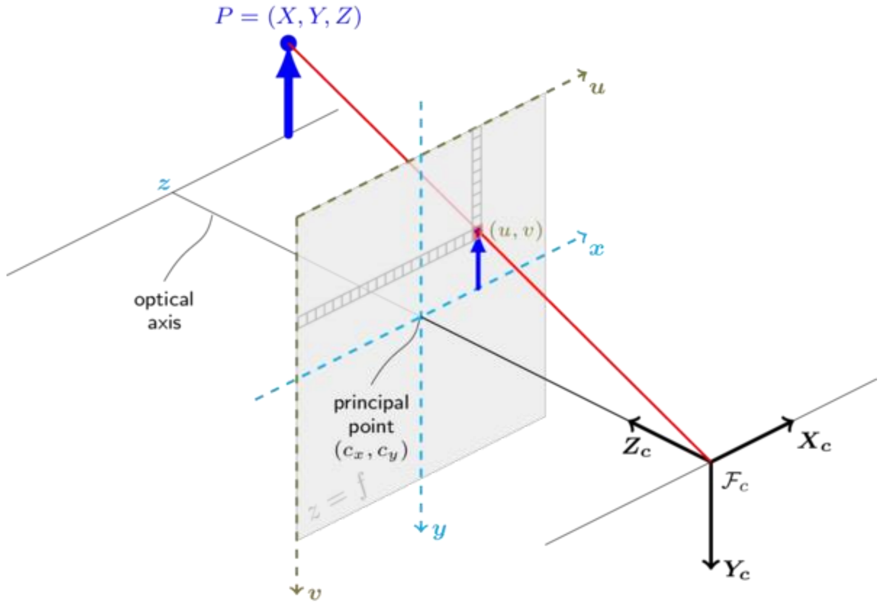
Lens Distortion

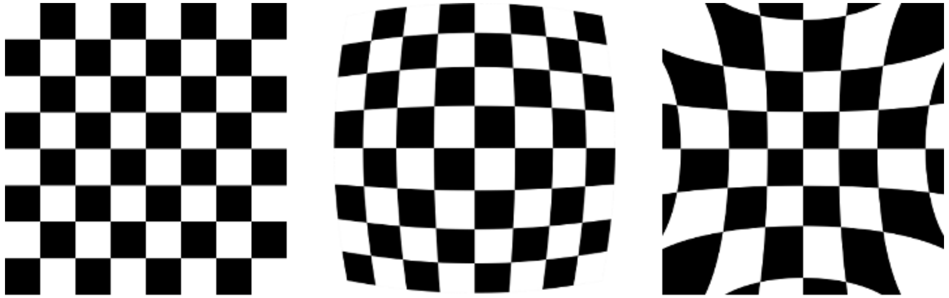


$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$
$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

https://hal-enpc.archives-ouvertes.fr/hal-01556898/document
https://jo.dreggn.org/home/2016_optics.pdf

Everything Together in OpenCV (Full Model)





No distortion

Positive radial distortion
(Barrel distortion)

Negative radial distortion
(Pincushion distortion)

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

$$x' = x/z$$

$$y' = y/z$$

$$x'' = x' \frac{1+k_1 r^2 + k_2 r^4 + k_3 r^6}{1+k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 x' y' + p_2(r^2 + 2x'^2)$$

$$y'' = y' \frac{1+k_1 r^2 + k_2 r^4 + k_3 r^6}{1+k_4 r^2 + k_5 r^4 + k_6 r^6} + p_1(r^2 + 2y'^2) + 2p_2 x' y'$$

$$\text{where} \quad r^2 = x'^2 + y'^2$$

$$u = f_x * x'' + c_x$$

$$v = f_y * y'' + c_y$$

https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

Recall the Updated Equation of Pinhole Camera Model

- Why do we put an extra 1 on both sides of the equation?

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

$$\boxed{\begin{array}{l} x' = x/z \\ y' = y/z \end{array}}$$

$$u = f_x * x' + c_x$$

$$v = f_y * y' + c_y$$

Image from Corke 2011

Euclidean Geometric Primitives

- Point

$$\boldsymbol{x} = (x, y) \in \mathcal{R}^2 \qquad \boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix} \qquad \boldsymbol{x} = (x, y, z) \in \mathcal{R}^3$$

- Points at infinity?

Homogeneous Representation of Euclidean Points

- Inhomogeneous point

$$\boldsymbol{x} = (x, y) \in \mathcal{R}^2 \qquad \boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix} \qquad \boldsymbol{x} = (x, y, z) \in \mathcal{R}^3$$

- Homogeneous point

2D projective space $\mathcal{P}^2 = \mathcal{R}^3 - (0, 0, 0)$

$$\tilde{\boldsymbol{x}} = (\tilde{x}, \tilde{y}, \tilde{w}) \in \mathcal{P}^2 \qquad \tilde{\boldsymbol{x}} = (\tilde{x}, \tilde{y}, \tilde{w}) = \tilde{w}(x, y, 1) = \tilde{w}\bar{\boldsymbol{x}},$$

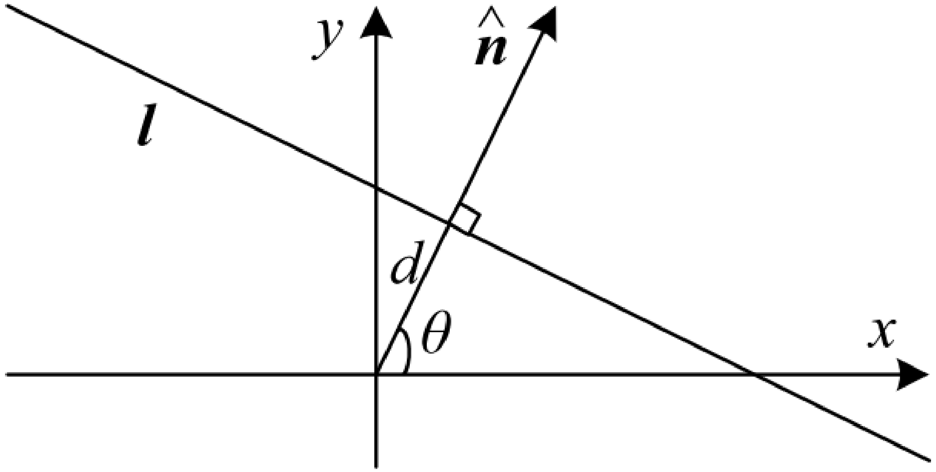
  - Points at infinity (or ideal points): $\{(\tilde{x}, \tilde{y}, \tilde{w}) | \tilde{w} = 0\}$

Advantages of Homogeneous Representation

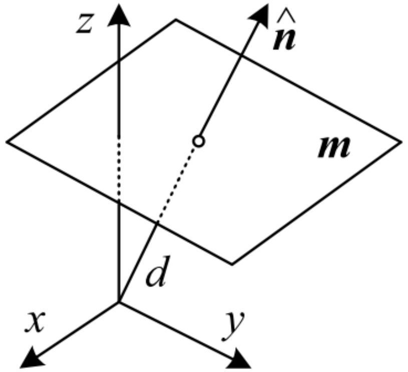
- Homogeneous 2D line

$$\bar{\boldsymbol{x}} \cdot \tilde{\boldsymbol{l}} = ax + by + c = 0$$

$$\tilde{\boldsymbol{l}} = (a, b, c)$$



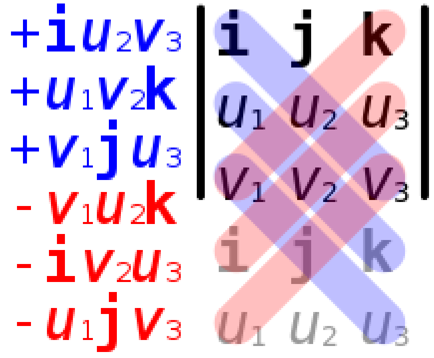
- Homogeneous 3D plane



Advantages of Homogeneous Representation

- Joints and intersections
  - Intersection point of two lines

$$\tilde{x} = \tilde{l}_1 \times \tilde{l}_2$$



  - Line jointing two points

$$\tilde{l} = \tilde{x}_1 \times \tilde{x}_2$$

- Duality between 2D lines and 2D points

Advantages of Homogeneous Representation

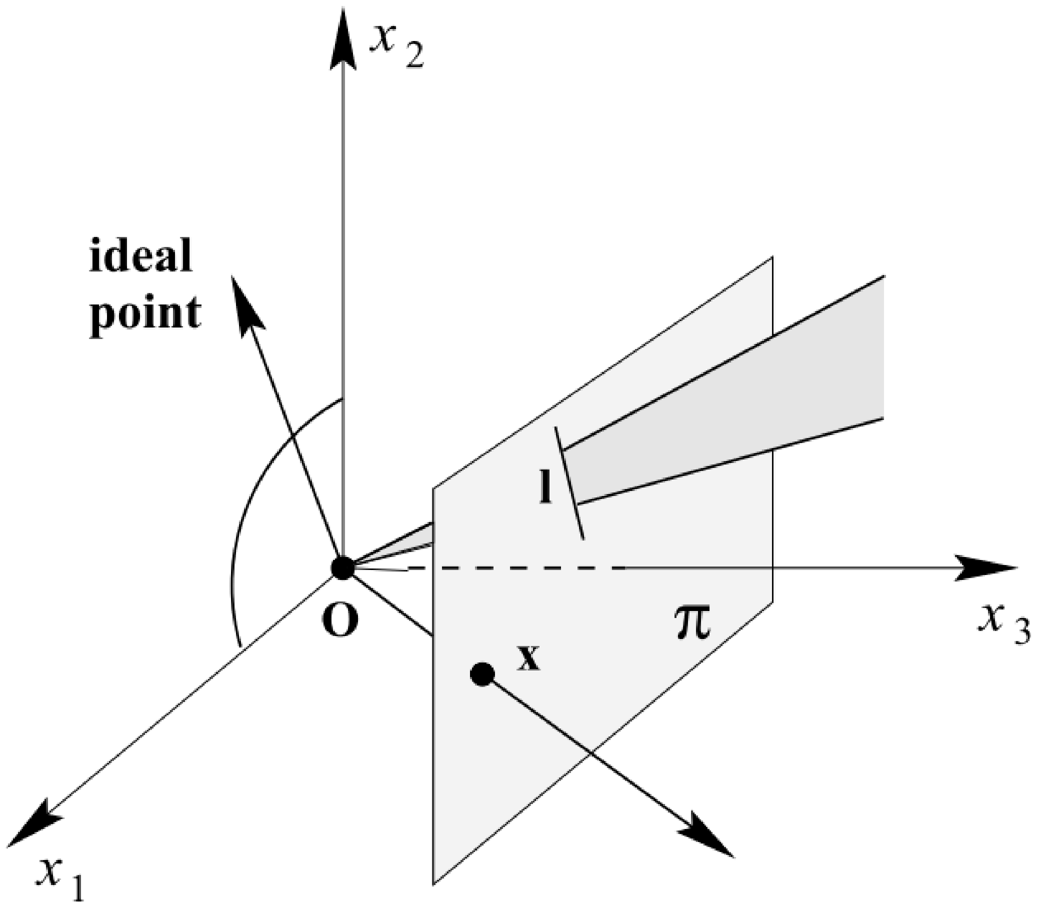
- Intersection of parallel lines $ax+by+c=0$ and $ax+by+c'=0$?
  - Point at infinity $(b, -a, 0)^\mathsf{T}$
- Line at infinity (or ideal line)?

$$\mathbf{l}_\infty = (0, 0, 1)^\mathsf{T}$$

  - The set of all ideal points
  - The set of directions of lines

Projective Space

- Projective geometry
  - The study of the geometry of projective space
  - A point in 2D as a ray in 3D
  - No need to distinguish between points/lines at infinity and ordinary points
  - One of the most important theoretic foundation of geometric computer vision



Recall the Updated Equation of Pinhole Camera Model

- In essence, this is a 3D-to-2D transformation/mapping

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- It is a composition of multiple kinds of transformations
  - 2D-to-2D (affine)
  - 3D-to-3D (euclidean)
  - 3D-to-2D (perspective)

Image from Corke 2011

2D-to-2D Transformations

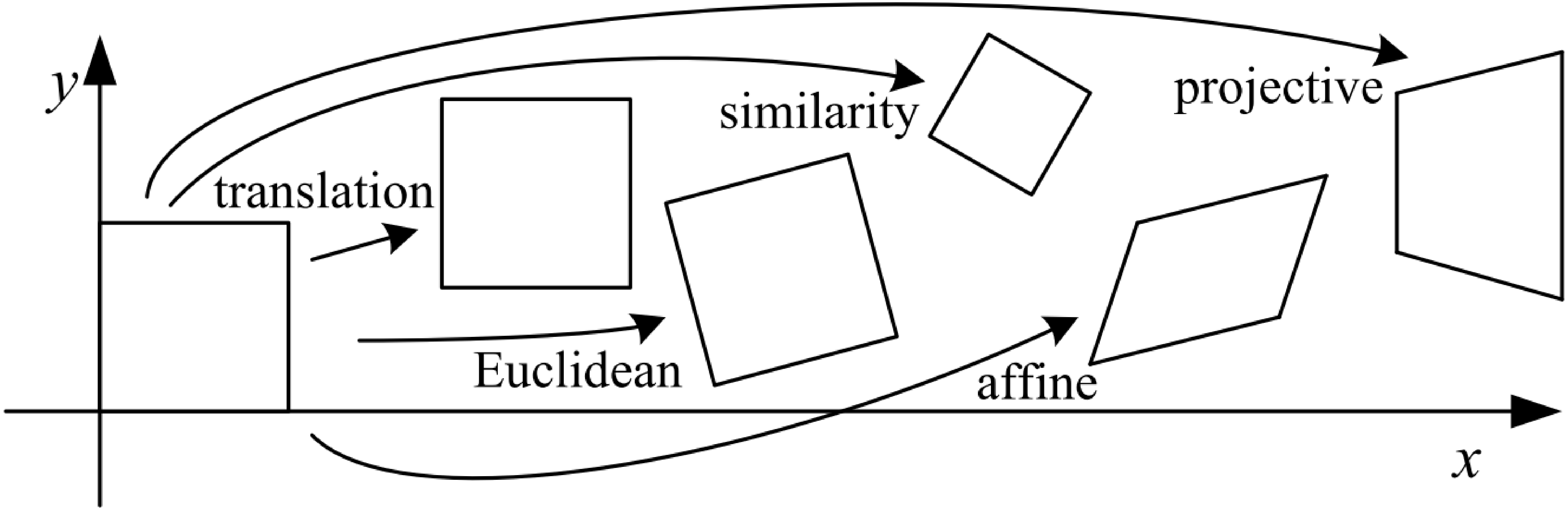


Image from: Szeliski 2011

Hierarchy of 2D Transformations

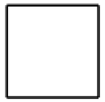
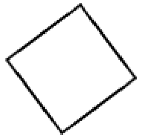
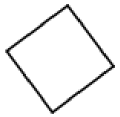
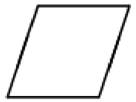
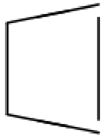
Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	3	lengths	
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{2\times 3}$	4	angles	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times 3}$	6	parallelism	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times 3}$	8	straight lines	

Image from: Szeliski 2011

Hierarchy of 3D Transformations

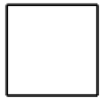
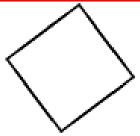
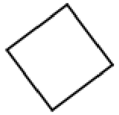
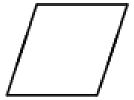
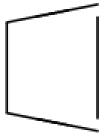
Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{3\times 4}$	3	orientation	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{3\times 4}$	6	lengths	
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{3\times 4}$	7	angles	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{3\times 4}$	12	parallelism	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{4\times 4}$	15	straight lines	

Image from: Szeliski 2011

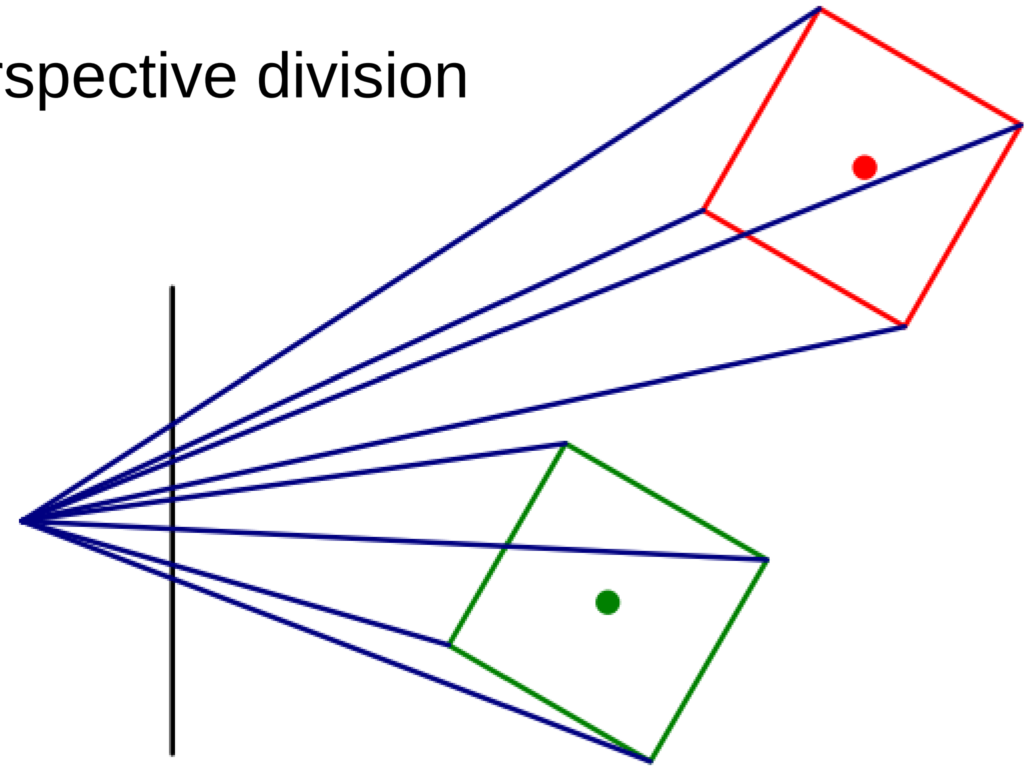
3D to 2D Projection

- **Perspective projection**

Perspective division

$$\bar{\boldsymbol{x}} = \mathcal{P}_z(\boldsymbol{p}) = \begin{bmatrix} x/z \\ y/z \\ 1 \end{bmatrix}$$

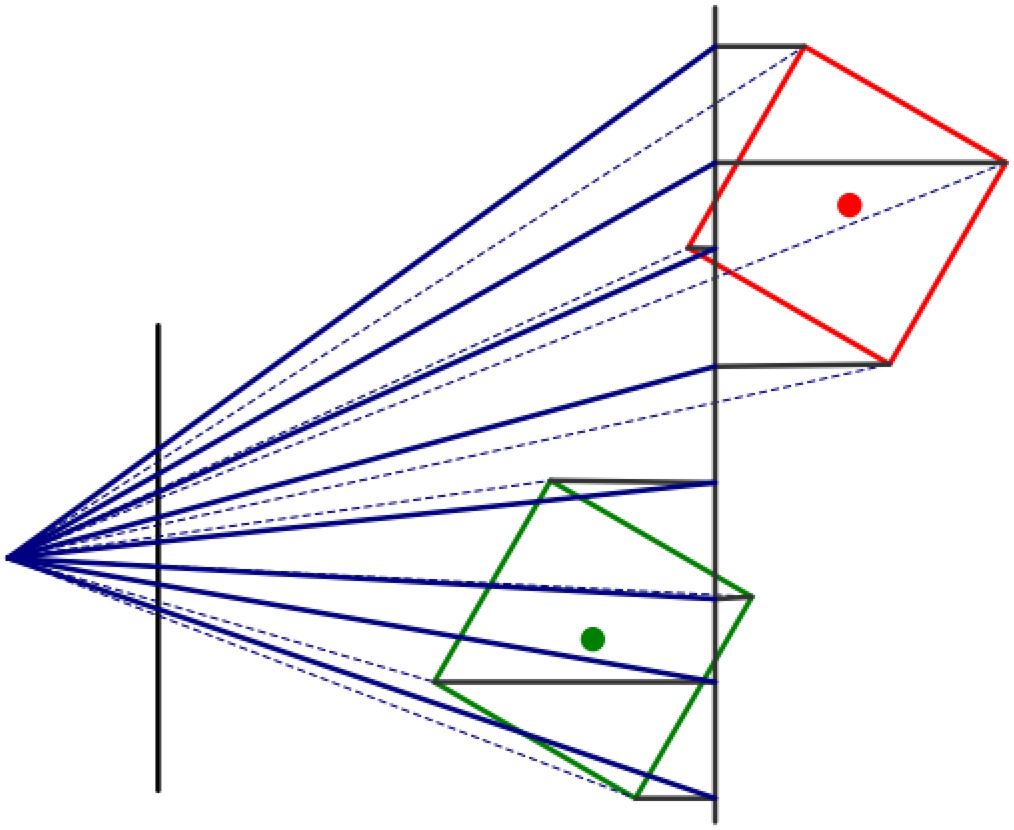
$$\tilde{\boldsymbol{x}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\boldsymbol{p}}$$



3D to 2D Projection

- Orthographic projection
- Approximation of telephoto

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tilde{\boldsymbol{p}}$$



References for Next Week

- Forsyth & Ponce 2011:
  - Section 1.2, 1.3, 12.1
- Szeliski 2011:
  - Section 6.3.1
- Corke 2011:
  - Section 11.2, 11.1
- Hartley & Zisserman 2003:
  - Section 2.3, 4.1, 4.4, 7.1, 7.2, 7.4
- Linear algebra:
  - Szeliski 2011: section A.1.1, A.1.2, A.1.3, A.2, A.2.1
  - Hartley & Zisserman 2003: A5.1, A5.2, A5.3