CS598 Fall 2024: 3D Vision
3D & Camera Basics

Shenlong Wang

Aug 27, 2024

ILLINOIS

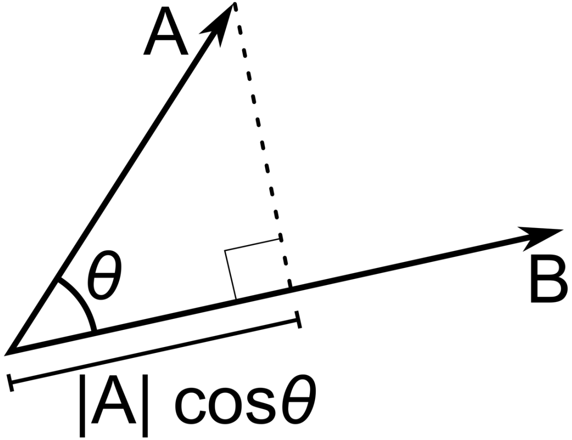
Some materials borrowed from Matthew O'Toole, Kris Kitani, Lana Lazebnik, Derek Hoeim, Sanja Fidler

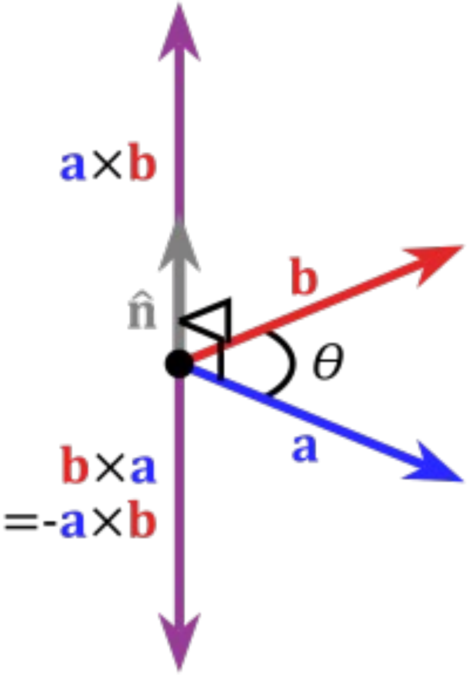
Today’s Agenda

- Coordinates & Axis
- Rigid Transforms & Rotations
- Camera Basics
- Perspective Geometry
- Homography

Prerequisite

- Vector
- Matrix
- Linear Transforms
- Dot Product
- Cross Product





Rigid Object

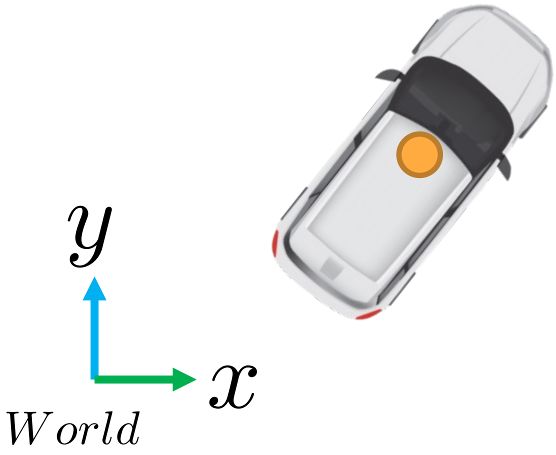
- **How would you quantitatively represent the state of the vehicle?**

Rigid Body Representation

- How would you represent the state of the vehicle?

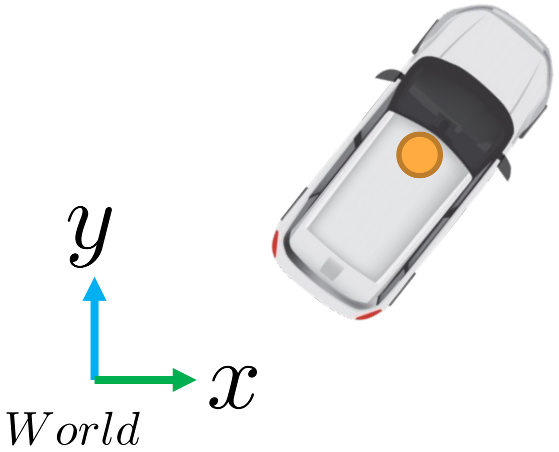
Rigid Body Representation

- How would you represent the state of the vehicle?



Rigid Body Representation

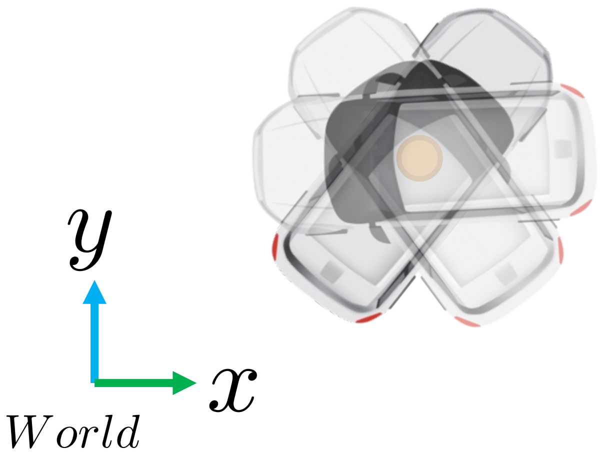
- How would you represent the state of the vehicle?



$$\mathbf{x} = (t_x, t_y)$$

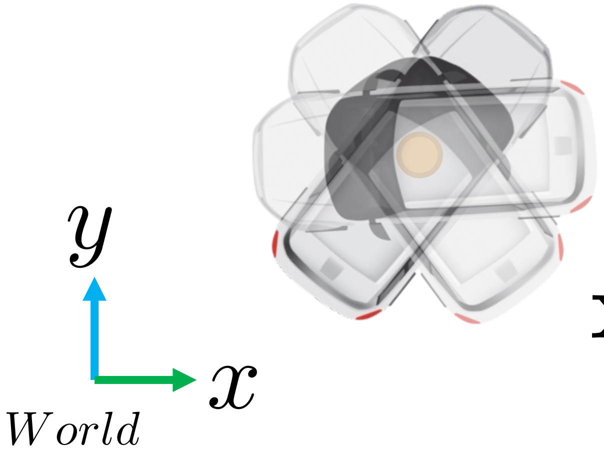
Rigid Body Representation

- How would you represent the state of the vehicle?



Rigid Body Representation

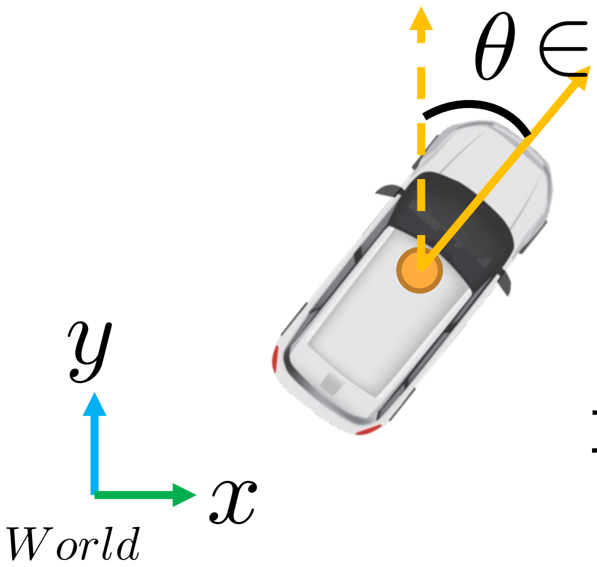
- How would you represent the state of the vehicle?



$$\mathbf{x} = (t_x, t_y, \theta)$$

Rigid Body Representation

- How would you represent the state of the vehicle?
    - State of a static rigid body = (Position, Orientation)

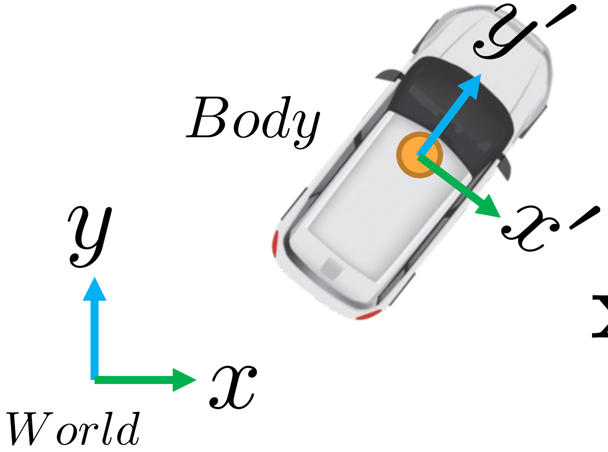


$$\theta \in [0, 2\pi)$$

$$\mathbf{x} = (t_x, t_y, \theta)$$

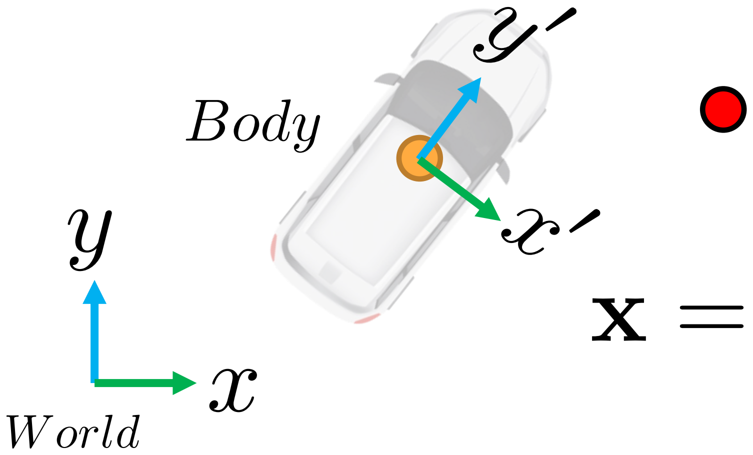
Body Frame

- Parameters of the states also defines a local coordinate frame



$$\mathbf{x} = (t_x, t_y, \theta)$$

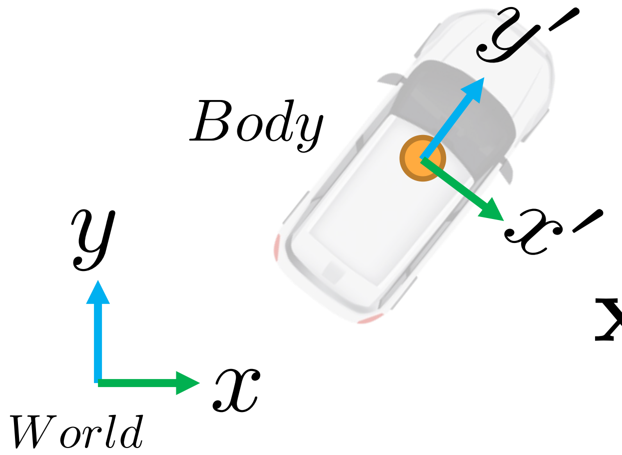
Body Frame



$$\mathbf{x} = (t_x, t_y, \theta)$$

Body Frame

- Can we get the pedestrian's position in the world frame?



Pedestrian's location in body frame:

$$\mathbf{p}' = (p'_x, p'_y)^T$$

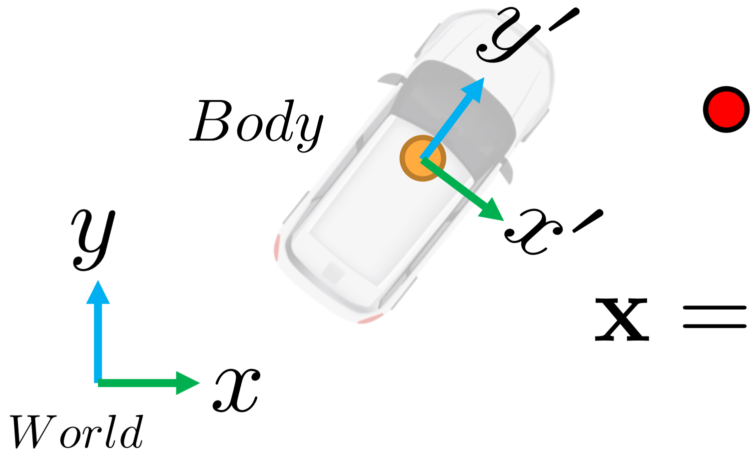
$$\mathbf{x} = (t_x, t_y, \theta)$$

Rigid Transform between Frames

$$\mathbf{p} = \mathbf{R}\mathbf{p}' + \mathbf{t} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} p'_x \\ p'_y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Step 1: rotate by theta

Step 2: translate



$$\mathbf{p}' = (p'_x, p'_y)^T$$

$$\mathbf{x} = (t_x, t_y, \theta)$$

Properties of Rigid Transform

$$\mathbf{p}' = \mathbf{R}\mathbf{p} + \mathbf{t} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} p'_x \\ p'_y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Step 1: rotate by theta

Step 2: translate

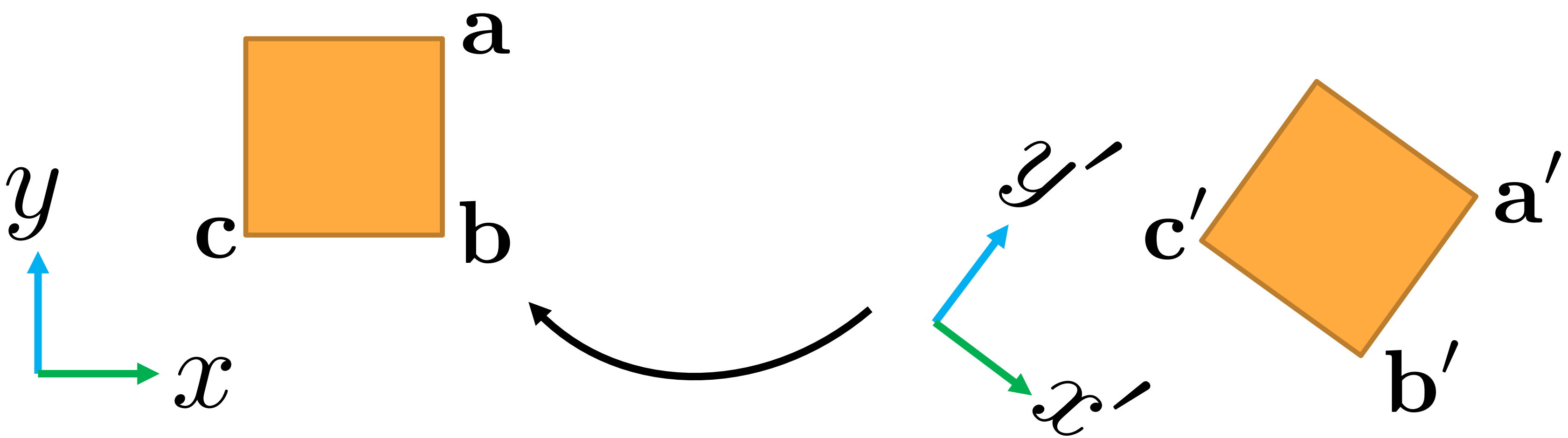
$$\mathbf{R}^T\mathbf{R} = \mathbf{R}\mathbf{R}^T = \mathbf{I}, \det \mathbf{R} = 1$$

Please validate the two properties offline

Properties of Rigid Transform

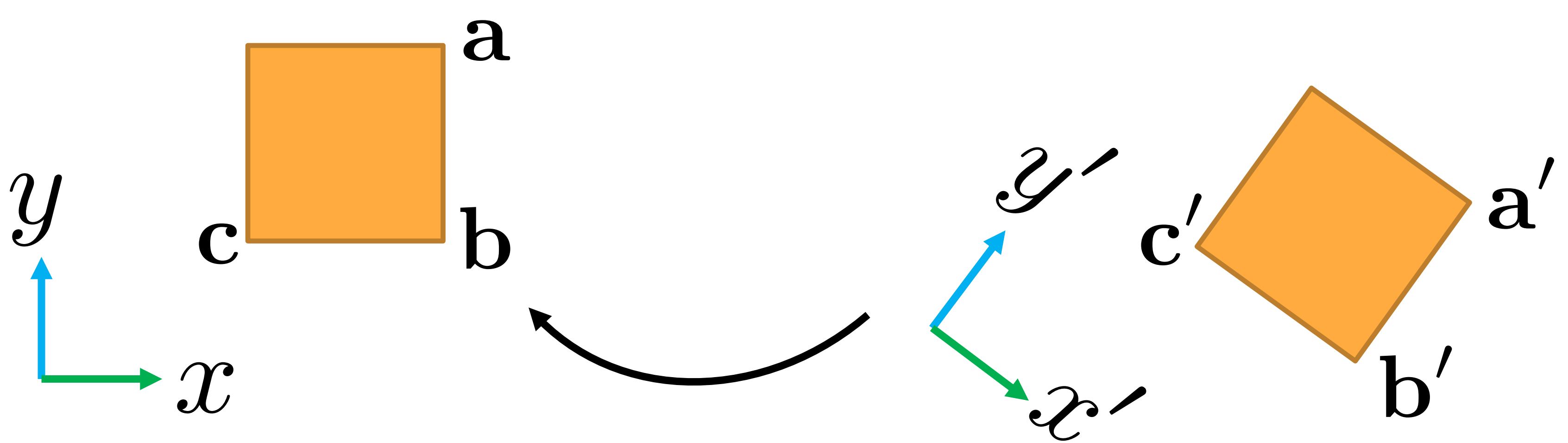
- Euclidean distance between any pair of two points is preserved:

$$(\mathbf{a}-\mathbf{b})^T(\mathbf{a}-\mathbf{b}) = (\mathbf{a}'-\mathbf{b}')^T(\mathbf{a}'-\mathbf{b}')$$



Properties of Rigid Transform

- Orientation-preserving or no reflection: any rotation between vectors is preserved:



Homogenous Coordinate

$$\begin{bmatrix} p_x \\ p_y \end{bmatrix}$$

⬇

$$\begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix}$$

Why matters?

Homogenous Coordinate

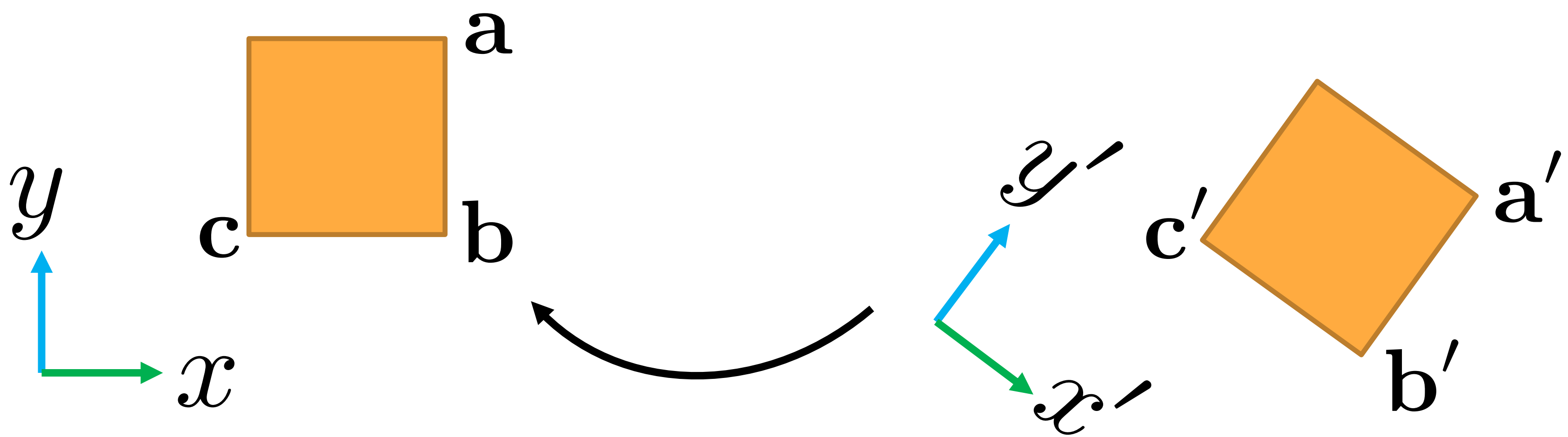
$$\begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} p'_x \\ p'_y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

⇓

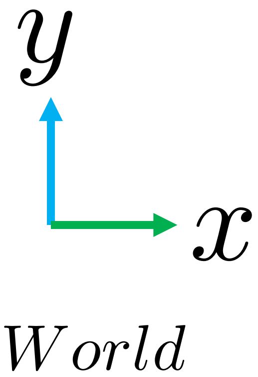
$$\begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p'_x \\ p'_y \\ 1 \end{bmatrix}$$

Homogenous Coordinate

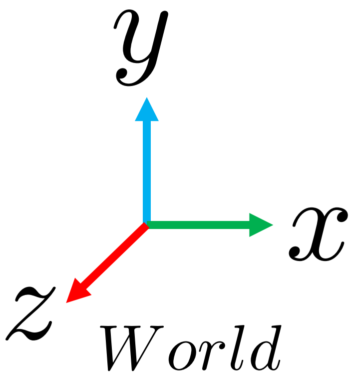
$$\hat{\mathbf{p}} = \mathbf{T}\hat{\mathbf{p}}' = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \hat{\mathbf{p}}'$$



3D Rigid Transform

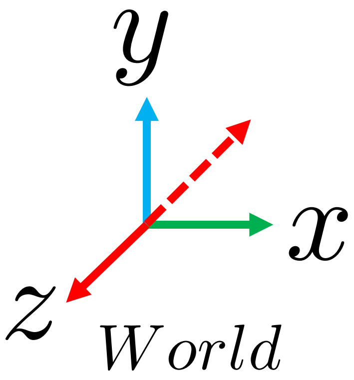


3D Rigid Transform

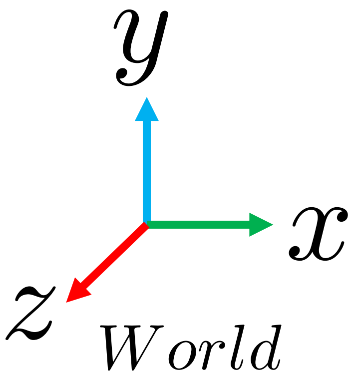


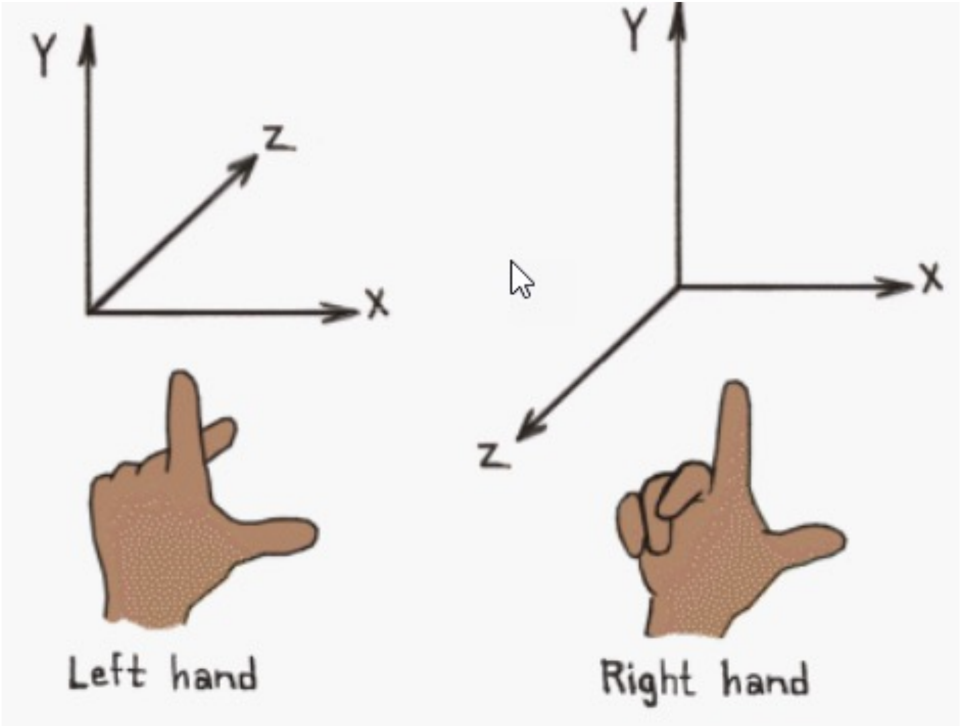
3D Rigid Transform

Which direction for z?



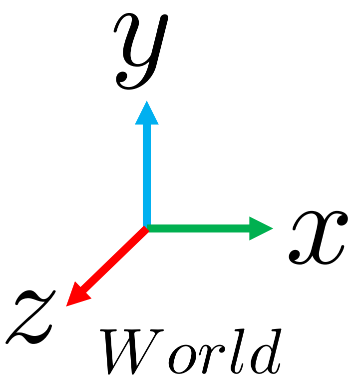
3D Rigid Transform



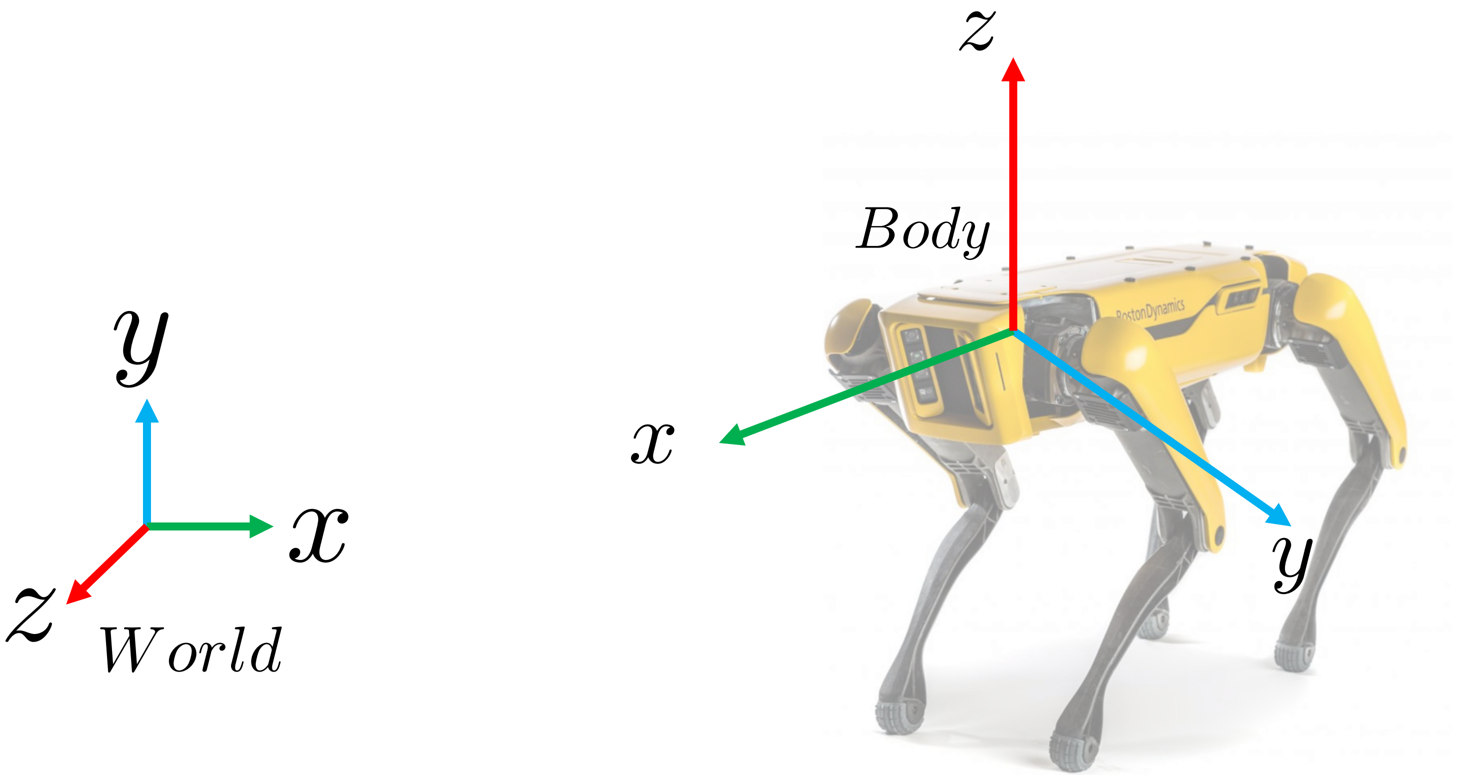


Roboticists and CVers mostly use right hand

3D Rigid Transform

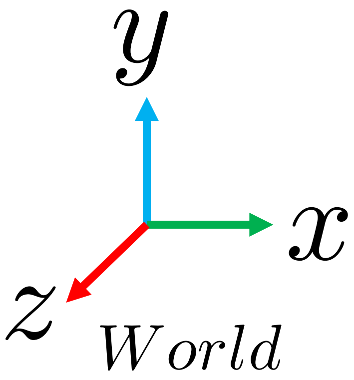


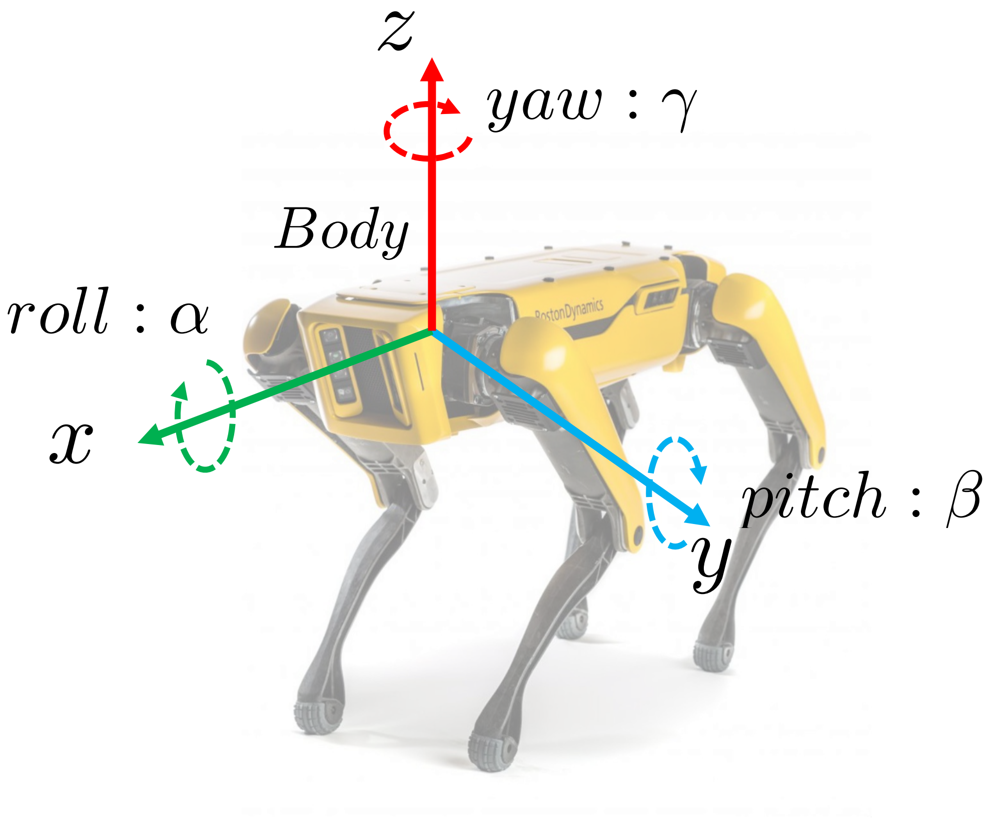
3D Rigid Transform



How many numbers do we need to represent a 3D rigid transform?

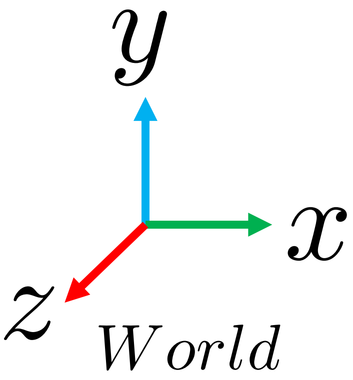
3D Rigid Transform

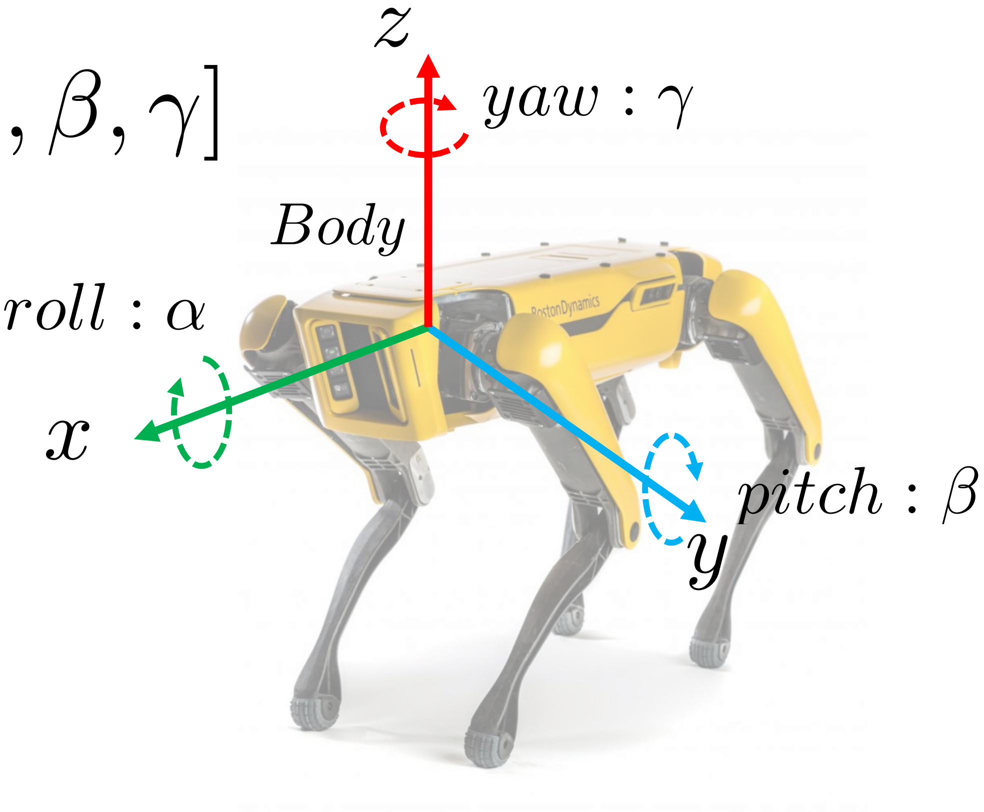




3D Rigid Transform

$$\mathbf{x} = [t_x, t_y, t_z, \alpha, \beta, \gamma]$$

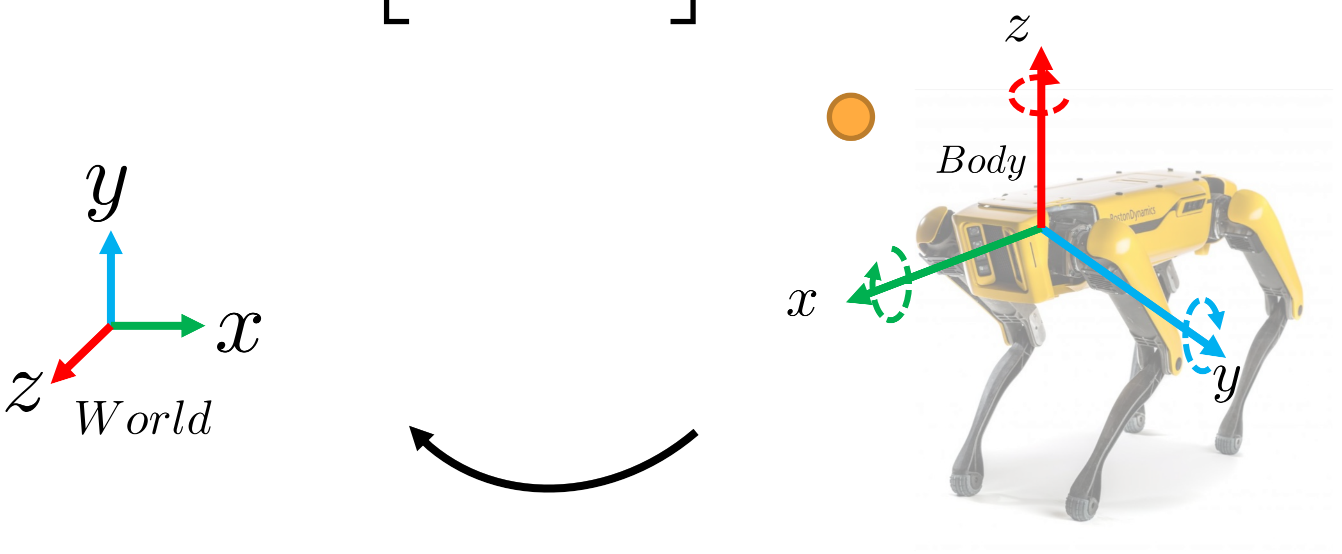




3D Rigid Transform

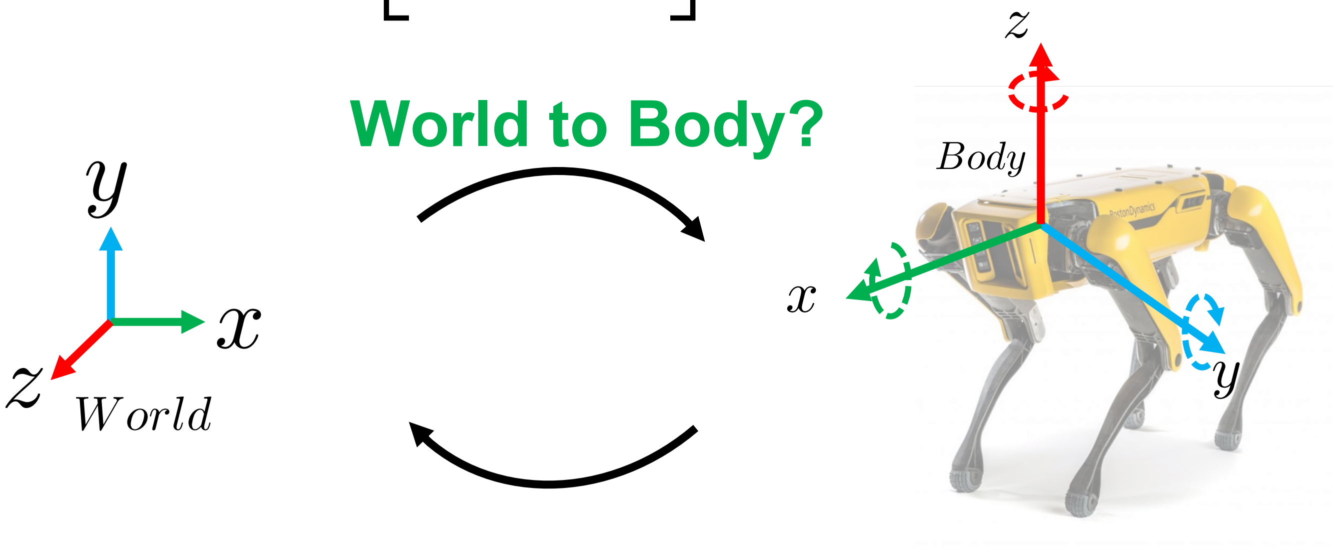
$$\hat{\mathbf{p}} = \mathbf{T}\hat{\mathbf{p}}' = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \hat{\mathbf{p}}'$$

$$(p'_x, p'_y, p'_z, 1)^T$$



3D Rigid Transform

$$\hat{\mathbf{p}} = \mathbf{T}\hat{\mathbf{p}}' = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \hat{\mathbf{p}}'$$



Inverse of Rigid Transform

$$\mathbf{T}^{-1} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

Try to verify the correctness!

$$\mathbf{R}^{-1} = \mathbf{R}^T, \mathbf{T}^{-1} \neq \mathbf{T}^T$$

Rotation Matrix

$$\{\mathbf{R} | \mathbf{R} \in \mathbb{R}^{3\times3}, \mathbf{R}^T\mathbf{R} = \mathbf{R}\mathbf{R}^T = \mathbf{I}, \det \mathbf{R} = 1\}$$

Orthogonal

Right hand coordinate system

Rotation Matrix

$$\{\mathbf{R} | \mathbf{R} \in \mathbb{R}^{3\times 3}, \mathbf{R}^T\mathbf{R} = \mathbf{R}\mathbf{R}^T = \mathbf{I}, \det \mathbf{R} = 1\}$$

Orthogonal

Right hand coordinate system

- Preserving Length: $$||\mathbf{R}\mathbf{v}|| = ||\mathbf{v}||$$
- Preserving Orientation: $$\mathbf{R}\mathbf{a} \times \mathbf{R}\mathbf{b} = \mathbf{R}(\mathbf{a} \times \mathbf{b})$$

Could you prove these?

Rotation Matrix

- Rotating a Vector:

$$\mathbf{p}' = \mathbf{R}\mathbf{p}$$

- Composition:

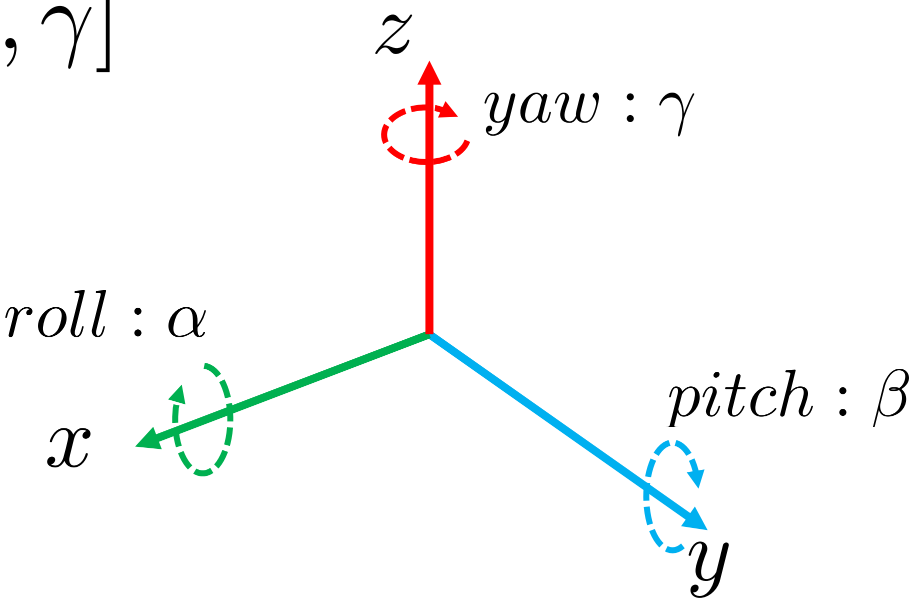
$$\mathbf{R}' = \mathbf{R}_2\mathbf{R}_1 \qquad \cancel{\mathbf{R}_1 + \mathbf{R}_2}$$

- Not compact: 3x3 numbers vs 3-DoF.
- Optimization/interpolation is not straightforward:

Euler Angles

- Three elemental rotations sequentially applied on each axes.

$$[\alpha, \beta, \gamma]$$



Euler Angles: Order Matters

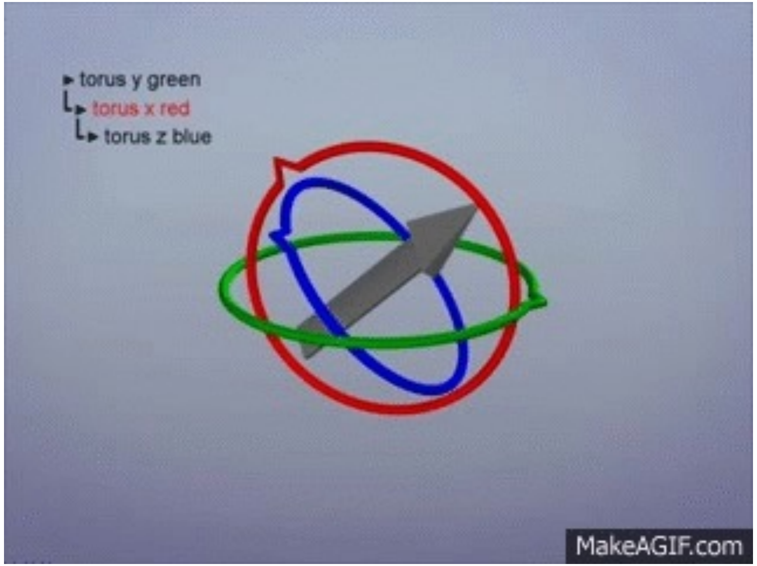
- Need to specify the order. In total there are **twelve** valid combinations.
- (Roll, Pitch, Yaw) is a special case:

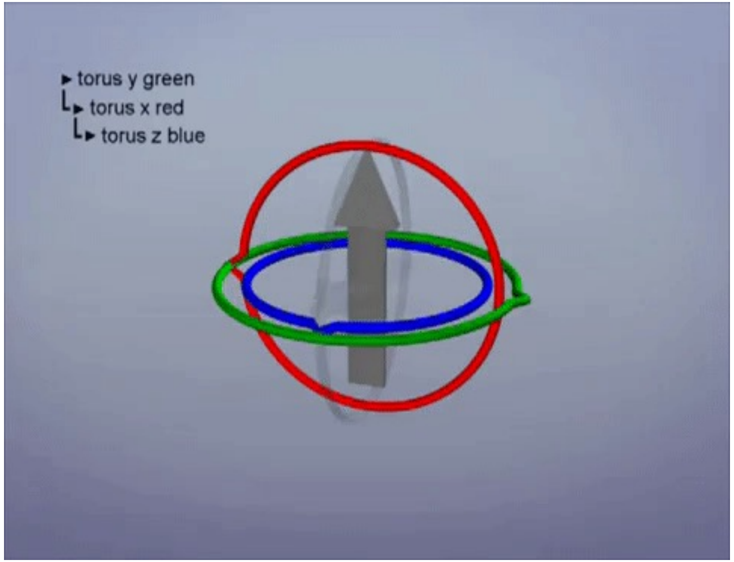
$$R = R_z(\gamma)\,R_y(\beta)\,R_x(\alpha) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

Euler Angles: Gimbal Lock

- Loss of one degree of freedom in a three-dimensional, three-gimbal mechanism

Rotation along y and z becomes the same!



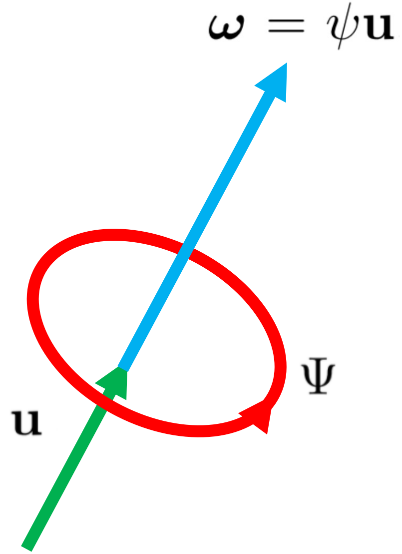


Axis-angle

- 4-number representation (3d unit vector + 1d angle)
- Ambiguities: (-angle, -axis) is the same as (angle, axis)
- Minimal version: Euler vector (3d arbitrary vector)
- Conversion to rotation (Rodriguez formula):

$$\mathbf{R} = \mathbf{I} + [\mathbf{u}]_\times \sin\psi + [\mathbf{u}]_\times^2 (1 - \cos\psi)$$

$$[\mathbf{u}]_\times = \begin{bmatrix} 0 & -u_z, & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}$$



Could you derive it?

Axis-angle

- Suffer from the "edges"

$$r_1 = \begin{pmatrix} 0 \\ 0 \\ 179^\circ \end{pmatrix} \quad r_2 = \begin{pmatrix} 0 \\ 0 \\ -179^\circ \end{pmatrix} \quad r_1 - r_2 = \begin{pmatrix} 0 \\ 0 \\ 358^\circ \end{pmatrix}$$

Actual angular difference is only 2 deg.

- Interpolation and composition is hard.
- Rotating a vector is not straightforward. We have to convert it back to matrix.

Unit Quaternions

- Quaternion:

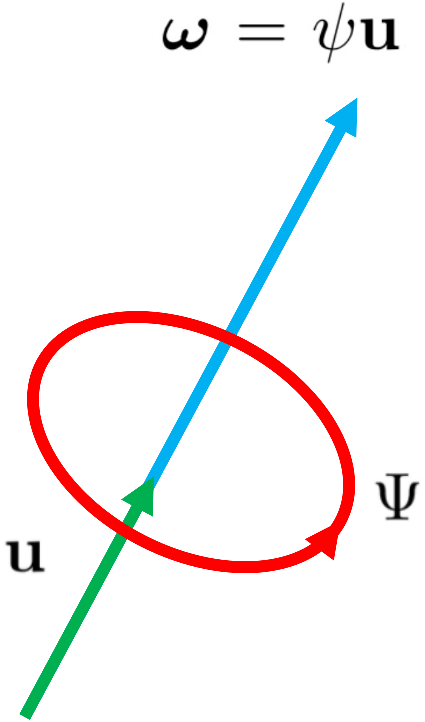
$$\mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$$

- Unit Quaternion as Rotation Representation:

$$\mathbf{q} = \left(\sin\frac{\Psi}{2}\mathbf{u}, \cos\frac{\Psi}{2}\right)$$

imagery   Real

$||q|| = 1$



Pros:

- Continuous
- Numerically stable
- Relatively compact
- Rotating a vector is efficient

https://en.wikipedia.org/wiki/Quaternions_and_spatial_rotation

Rotations Cheat Sheet

	Parameters	Singularities	Composition and Action
Matrix	9, orthogonality constraints	No	Easy
Euler Angle	3, [0, 2pi] or [0, pi]	Gimbal lock	Hard
Axis-Angle	4, unit axis vector	theta = 0	Hard
Rotation Vector	3, || v || < pi	Double representation	Hard
Quaternions	4, unit quaternion	Double representation, q -q	Easy

http://motion.cs.illinois.edu/RoboticSystems/3DRotations.html

Camera: History

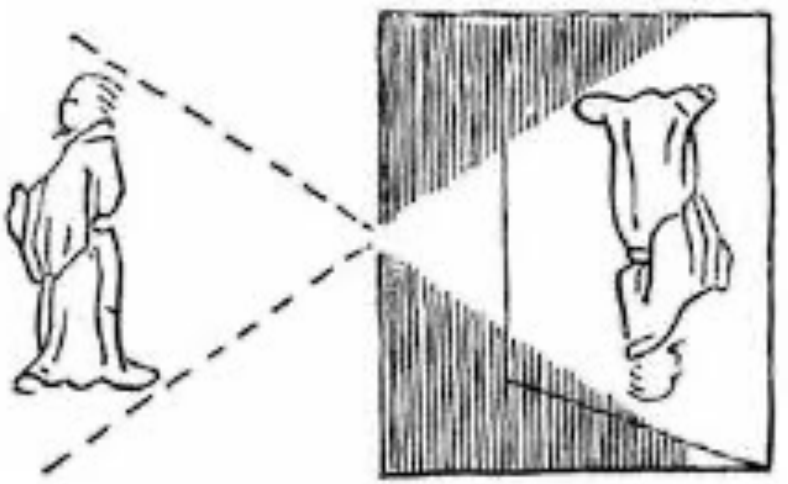
First mention …

Chinese philosopher Mozi
(470 to 390 BC)

景光之人煦若射，下者之入也高，高者之入也下

——- 墨子

The shadow resembles the human figure. Light that hits the feet passes through a small hole and projects upward, forming an image above. Light that hits the head passes through the hole and projects downward.

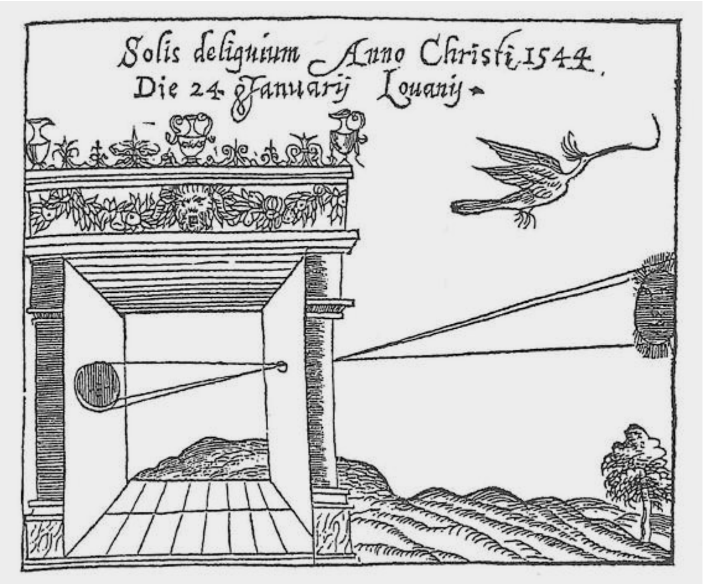


Camera: History

First mention …

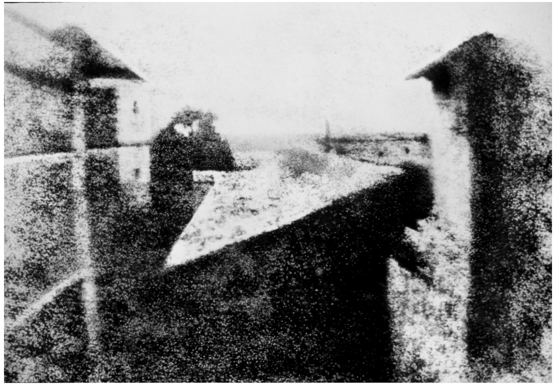
Chinese philosopher Mozi
(470 to 390 BC)

First camera …



Greek philosopher Aristotle
(384 to 322 BC)

Early Cameras



View from the Window at Le Gras (1825), the earliest surviving photograph

Kodak (1888) roll-film hand camera

Rectaflex, the first pentaprism SLR for eye-level viewing

Pinhole Camera

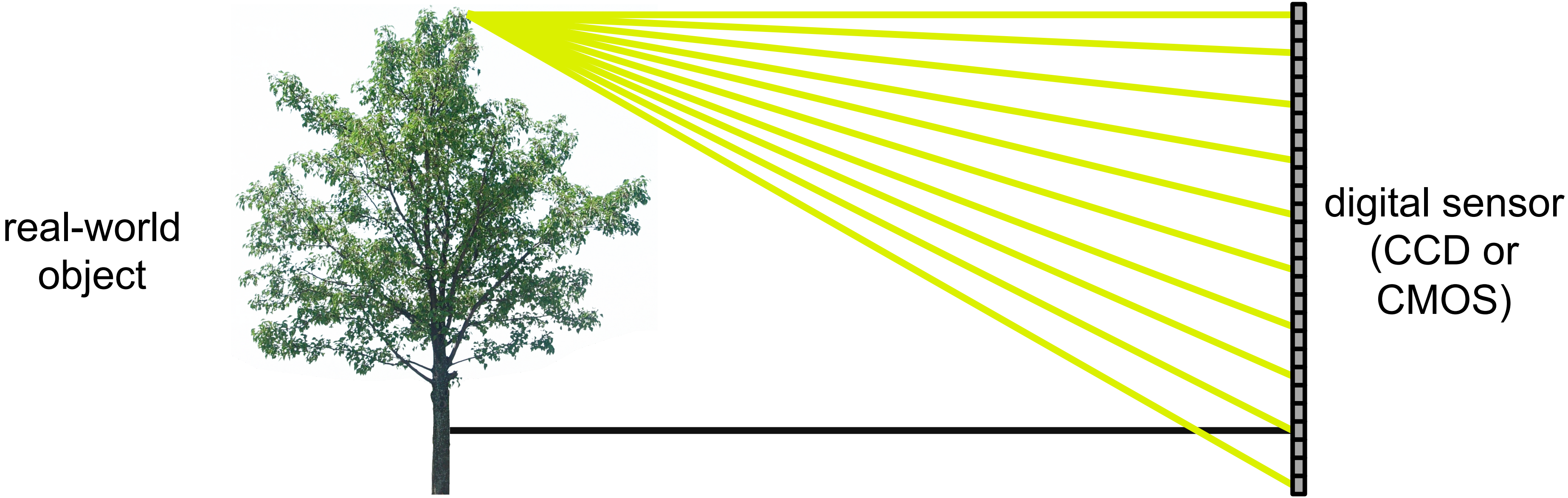
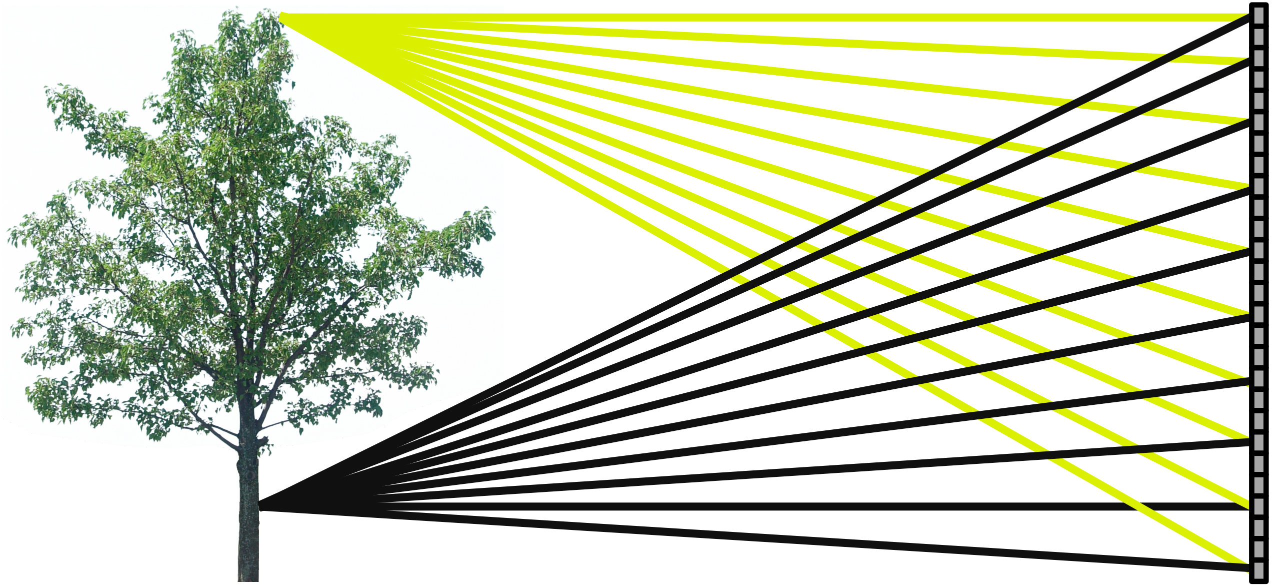


Image credit: Kris Kitani

Pinhole Camera

real-world
object



digital sensor
(CCD or
CMOS)

What does the image on the sensor look like?

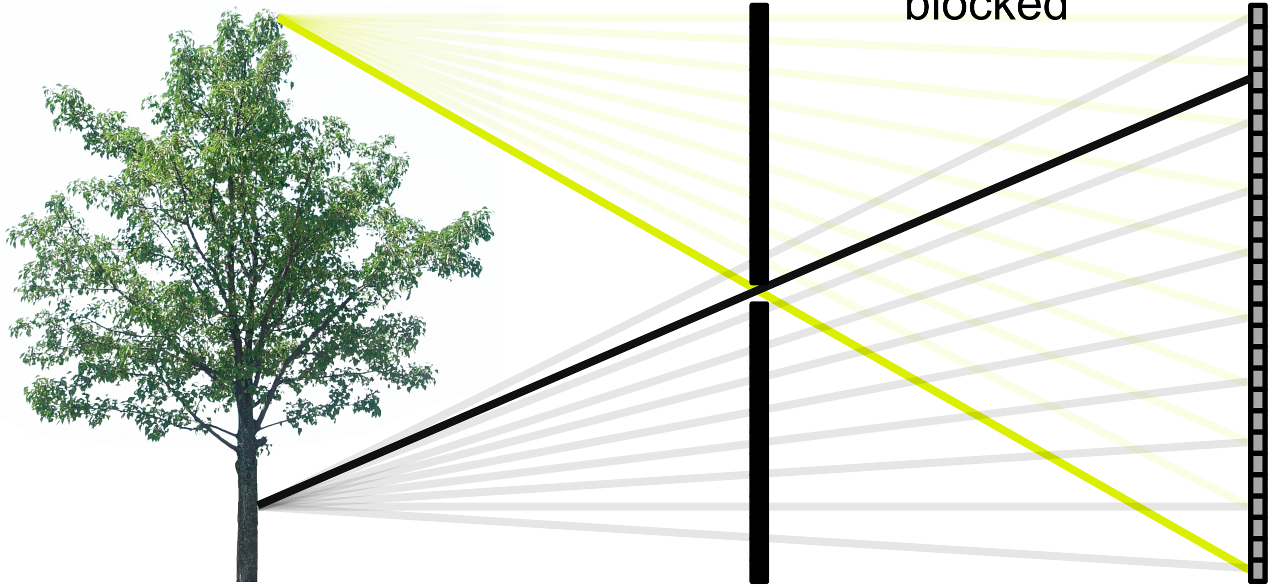
Image credit: Kris Kitani

Bare-Sensor Imaging

All scene points contribute to all sensor pixels

Image credit: Kris Kitani

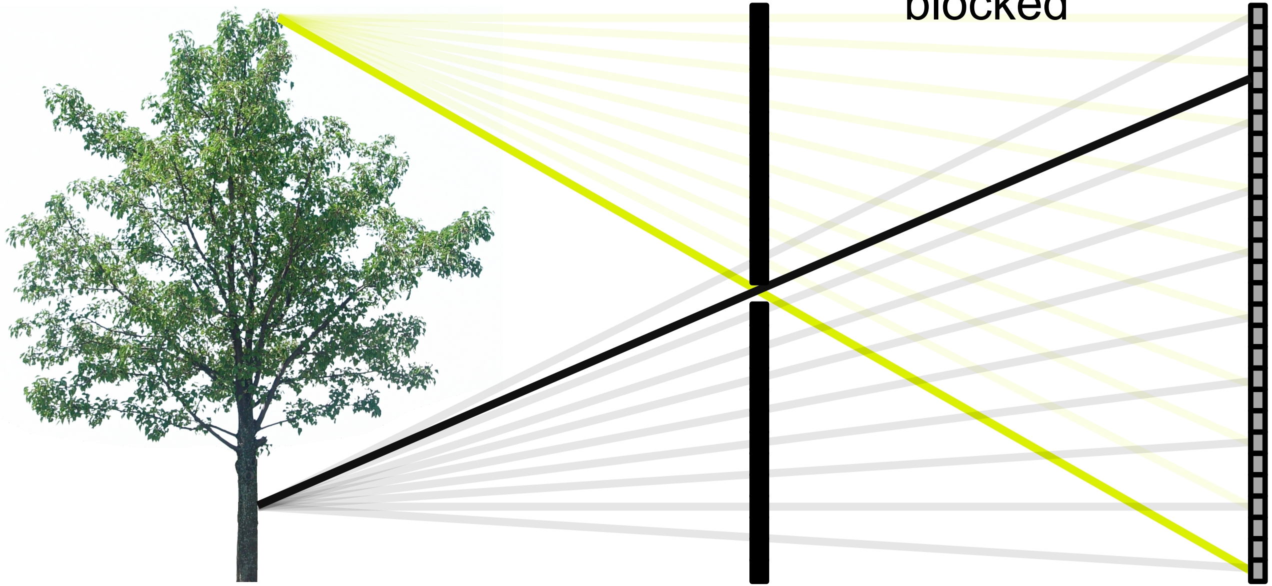
Pinhole Camera



Any drawbacks?

Image credit: Kris Kitani

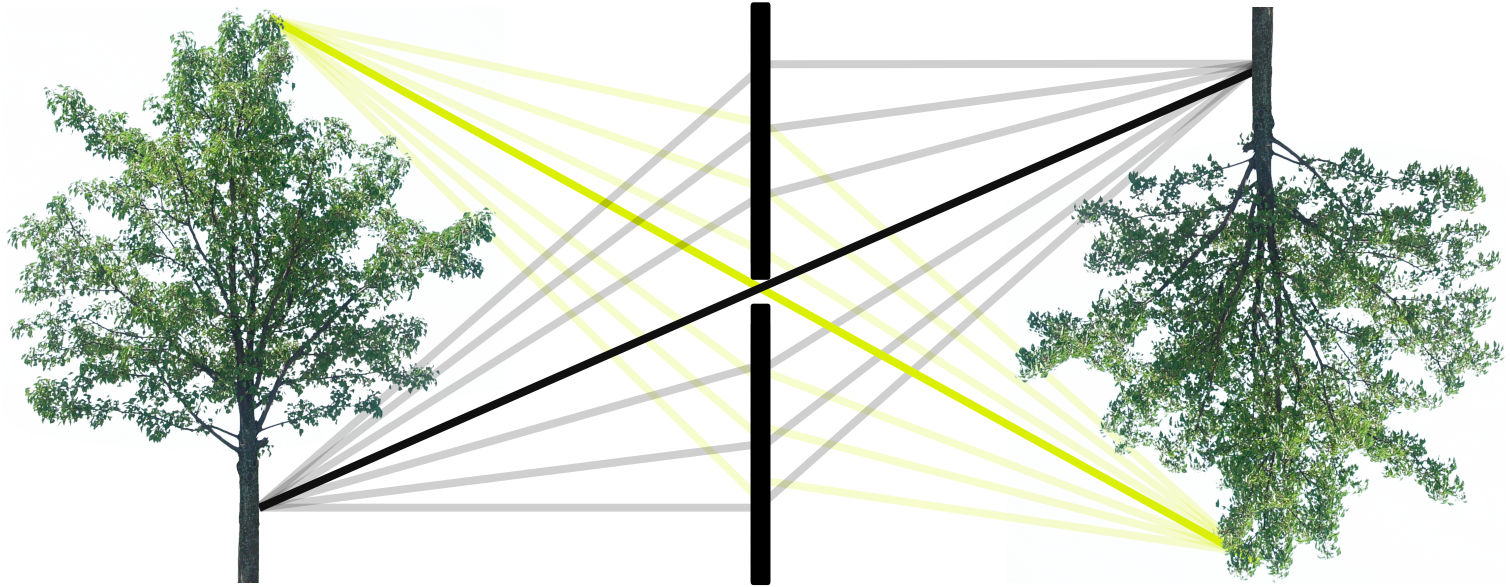
Pinhole Camera



What does the sensor image look like now?

Image credit: Kris Kitani

Pinhole Camera



Any drawbacks?

Image credit: Kris Kitani

Pinhole Camera

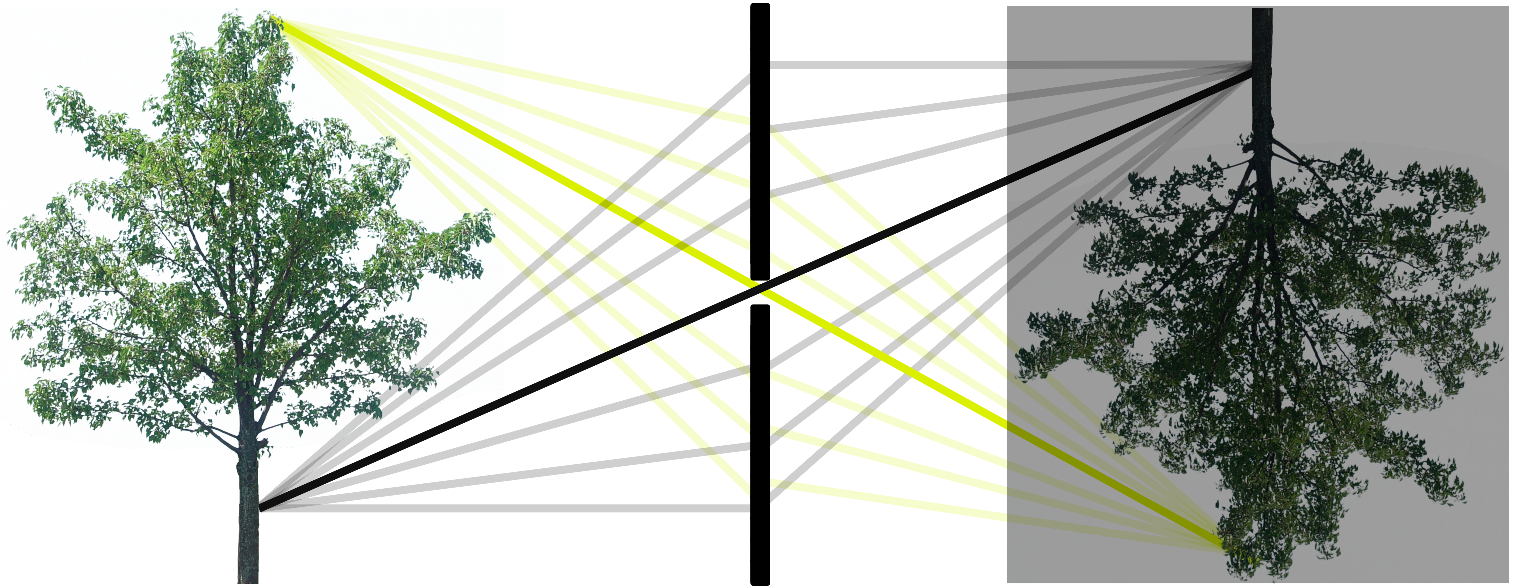


Image credit: Kris Kitani

Pinhole Camera

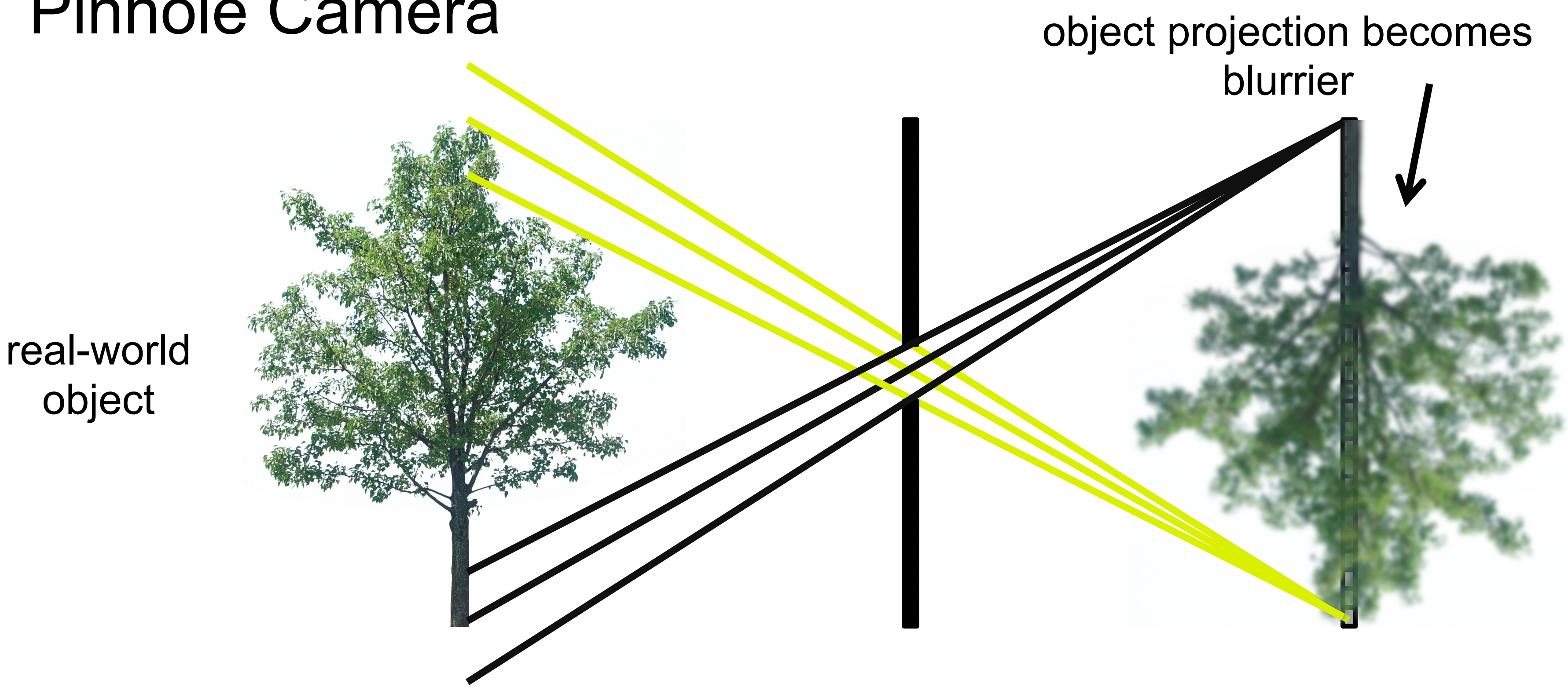


Image credit: Kris Kitani

Lens Camera

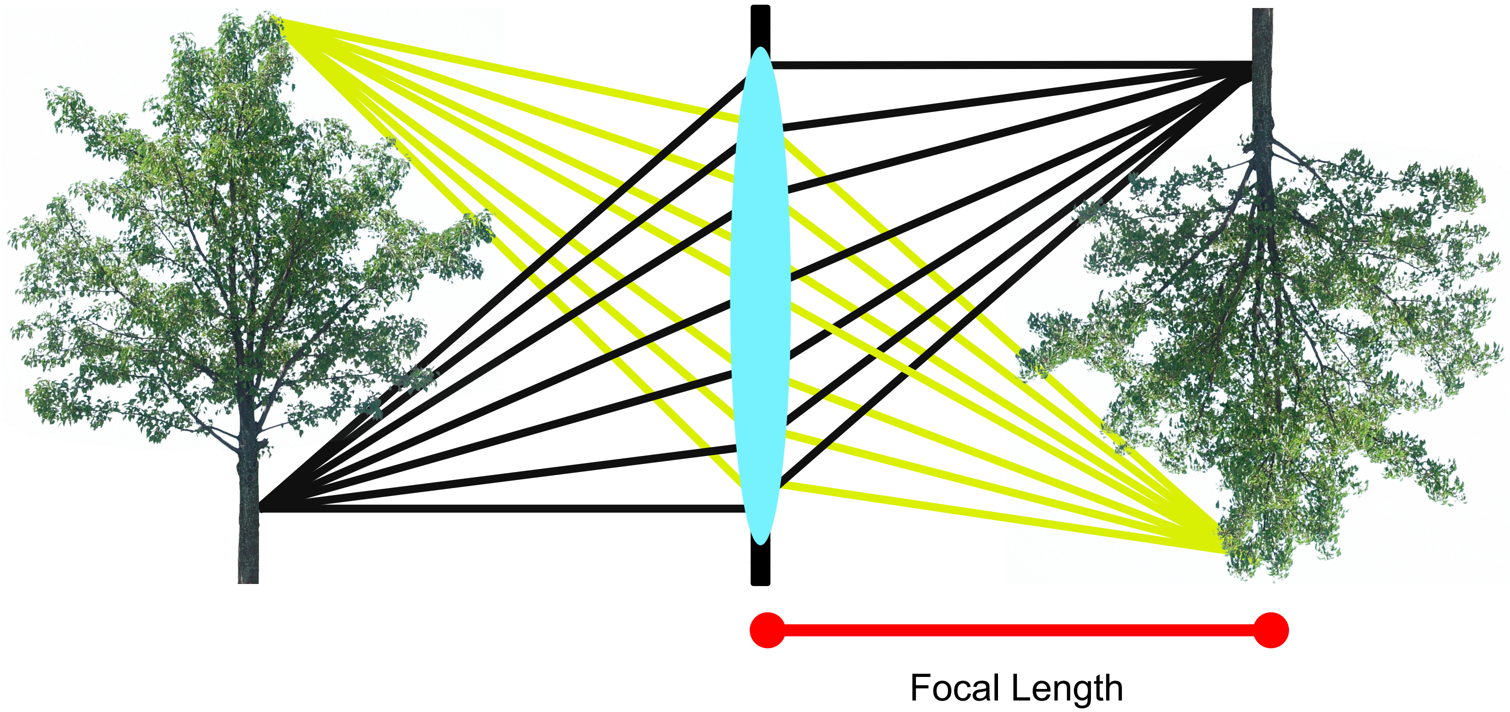


Image credit: Kris Kitani

Lens can vary

Image credit: Pentax

Choose the right focal length for your project

Focal Length: Short or Long?

Camera detecting front vehicles in highway beyond 200m.

Drone navigate in cluttered environment

Image credit: glfcat

Depth of focus

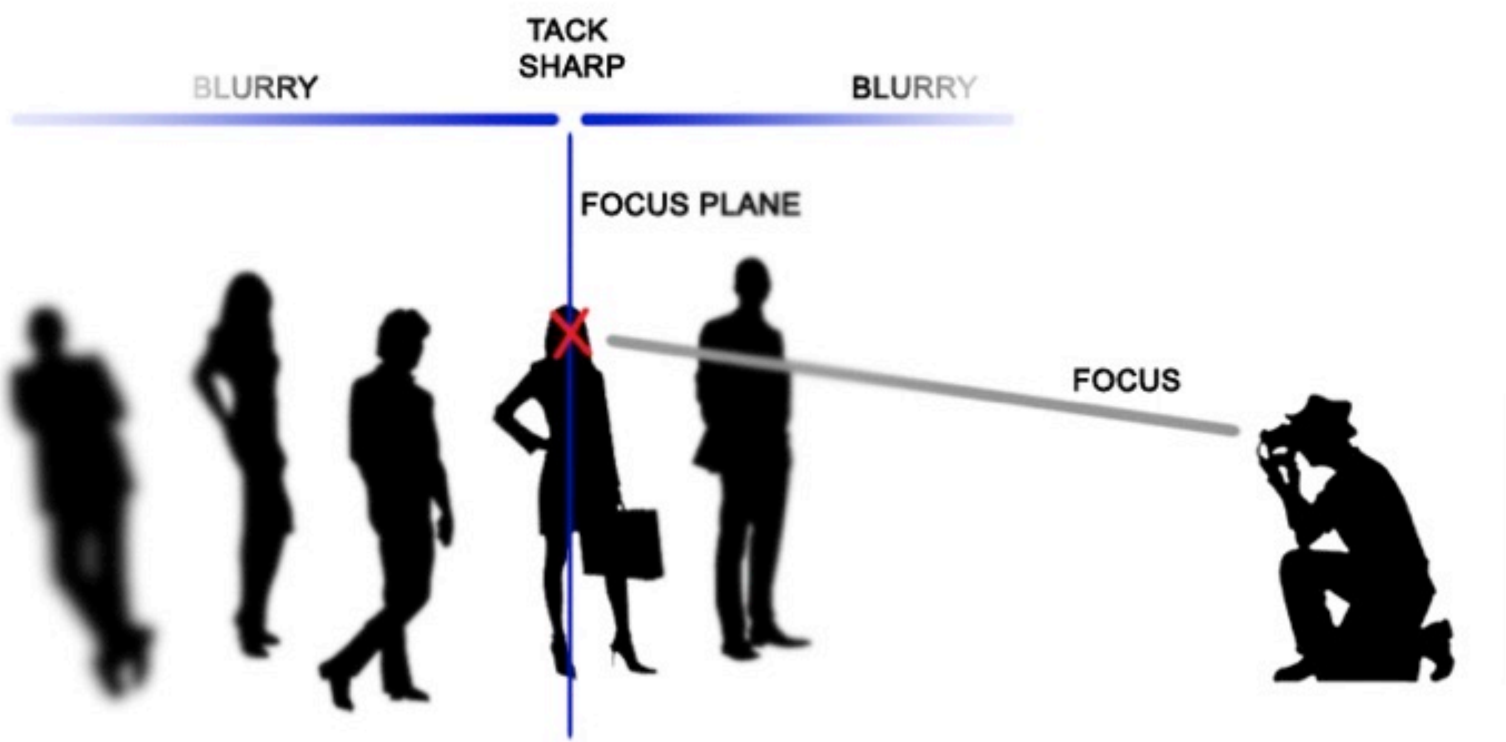


Image credit: https://digital-photography-school.com/

Aperture size

Image credit: https://digital-photography-school.com/

Aperture size

599 USD

1798 USD

Image credit: B&H

Short or Long?





Digital Camera Imaging Process

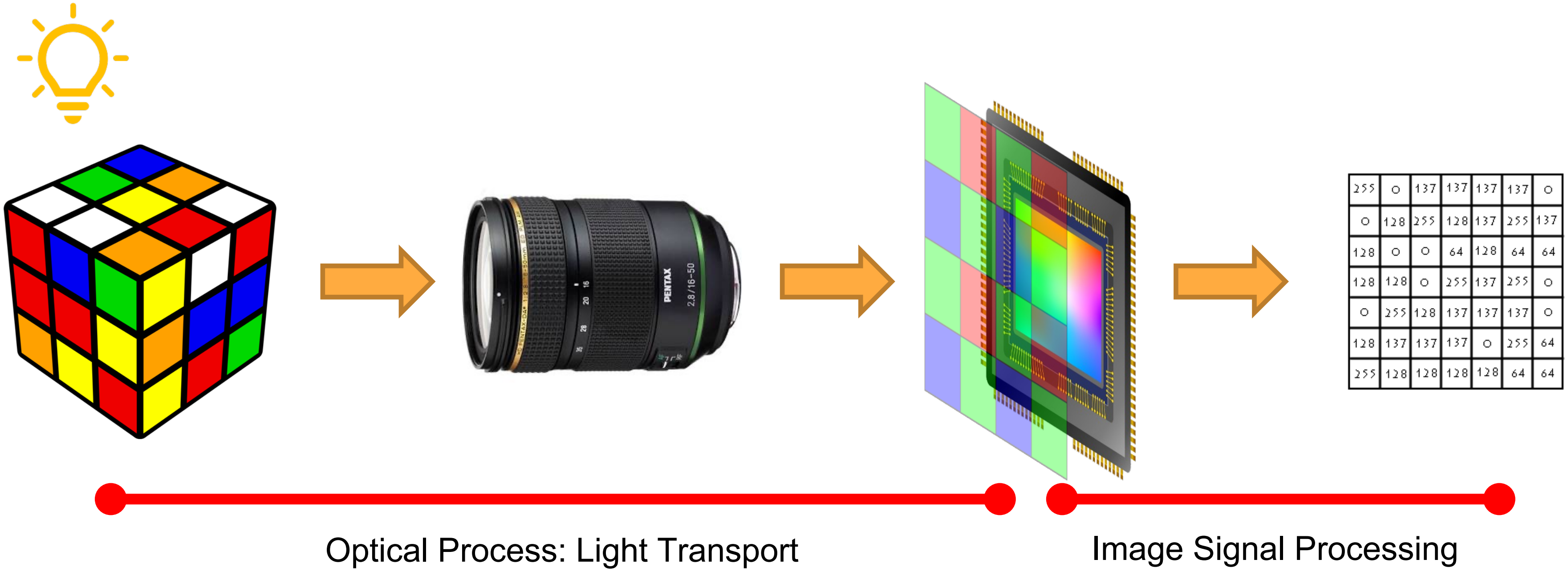
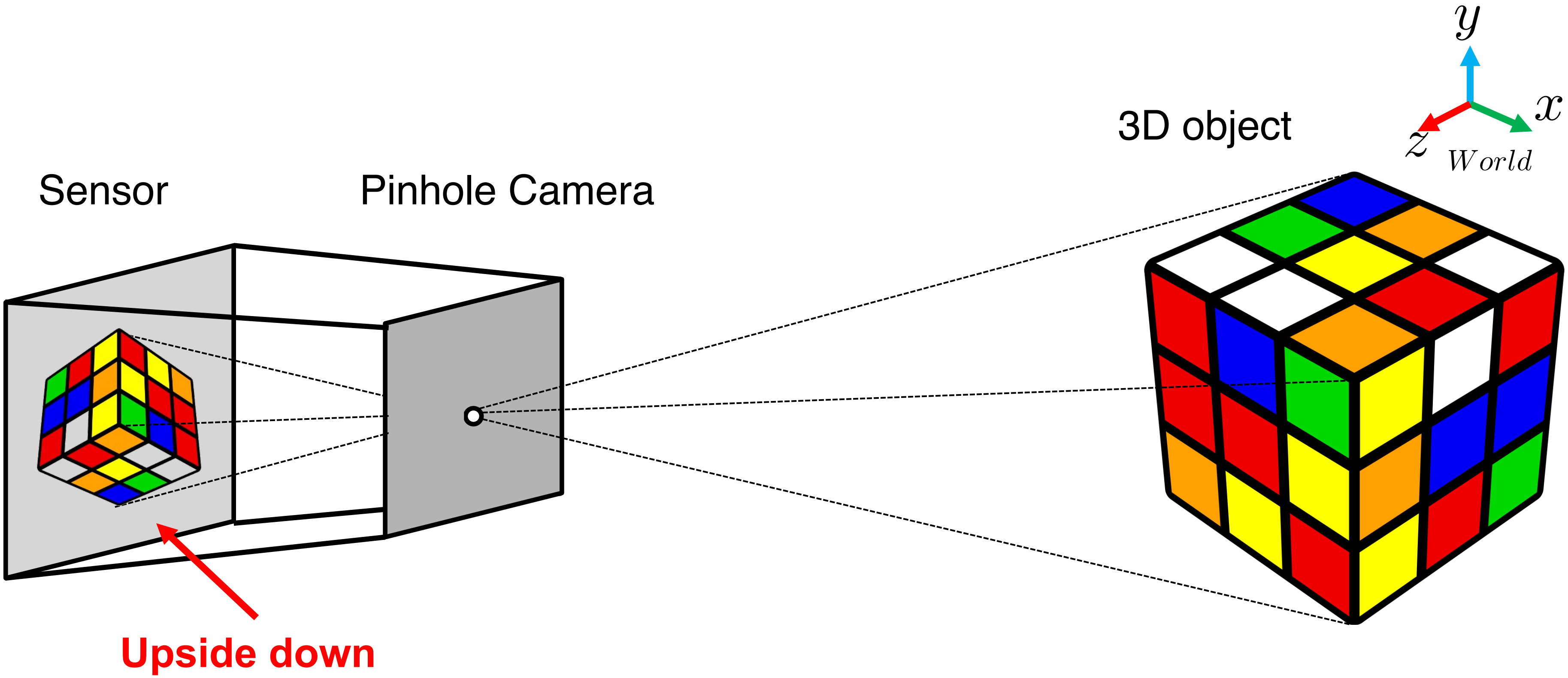


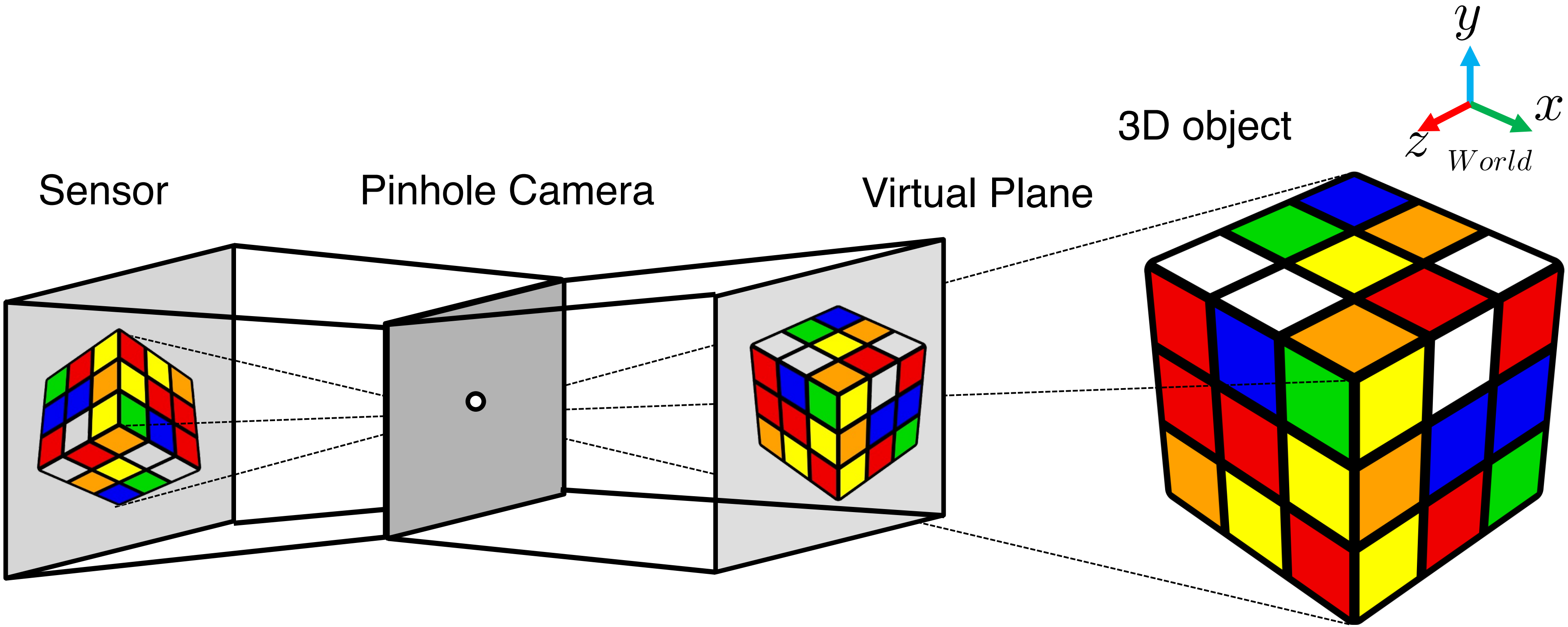
Image credit: pentax, wikiipedia

Camera Models

Camera Models

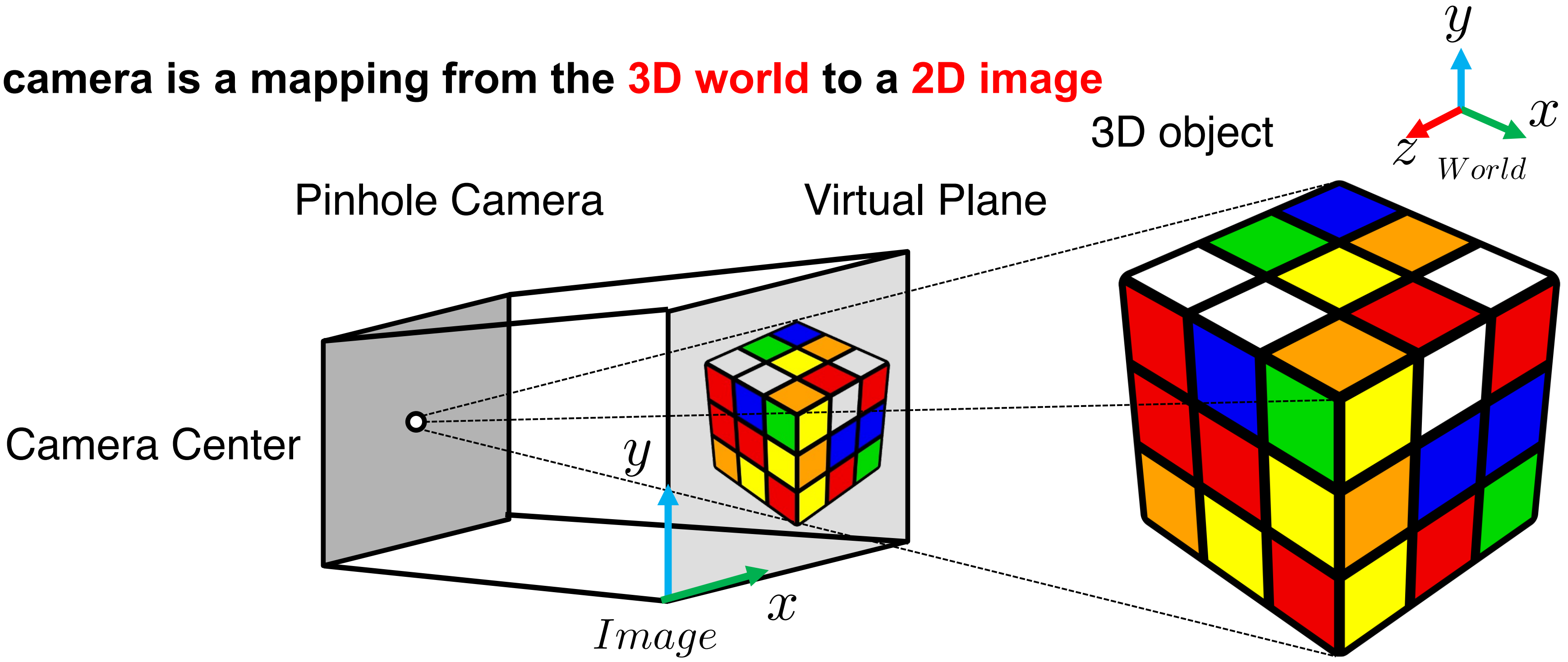


Camera Models



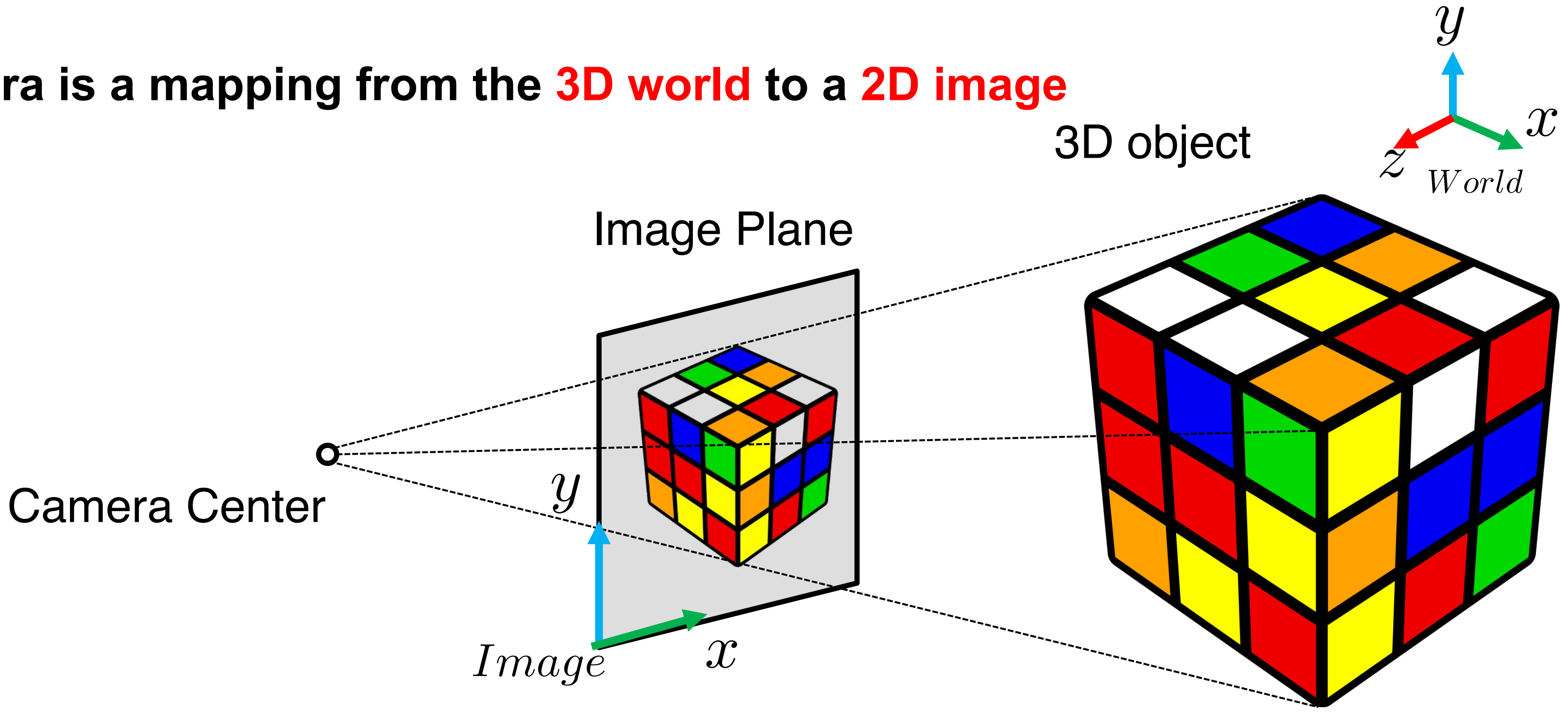
Camera Models

A camera is a mapping from the 3D world to a 2D image



Camera Models

A camera is a mapping from the 3D world to a 2D image



Camera as a Coordinate Transform

homogeneous coordinates

$$x = \mathbf{P}\mathbf{X}$$

2D image point

camera matrix

3D world point

What are the dimensions of each variable?

Camera as a Coordinate Transform

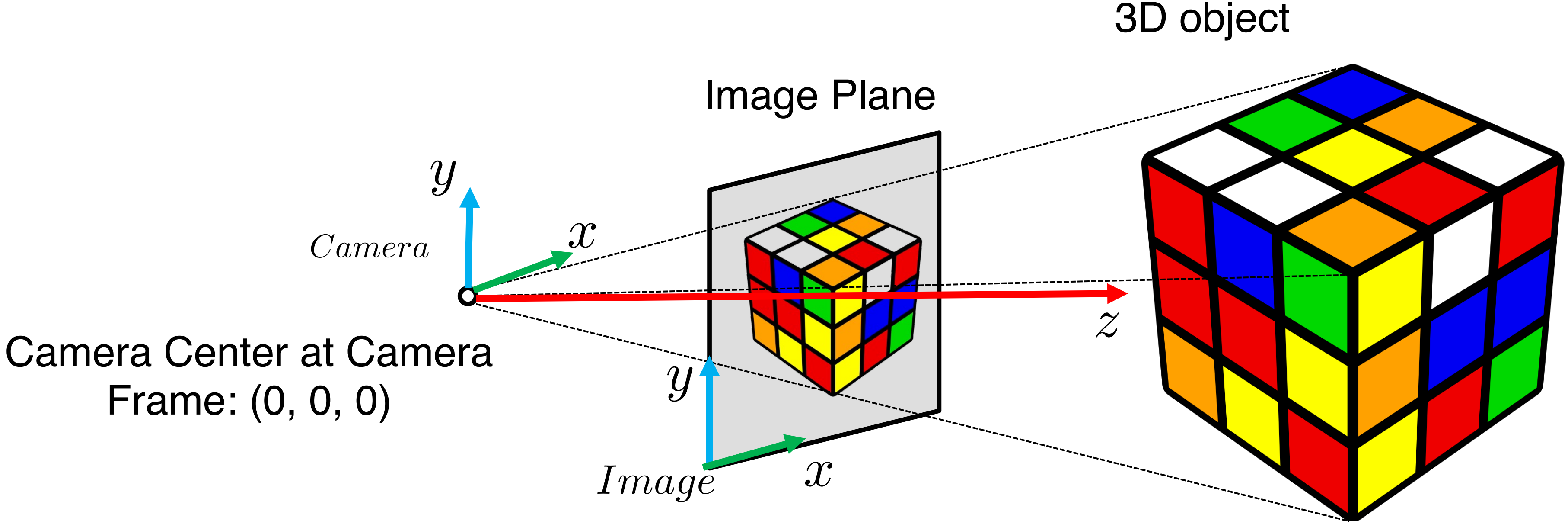
$$\boldsymbol{x} = \mathbf{PX}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

homogeneous image coordinates	camera projection matrix	homogeneous world coordinates
3 x 1	3 x 4	4 x 1

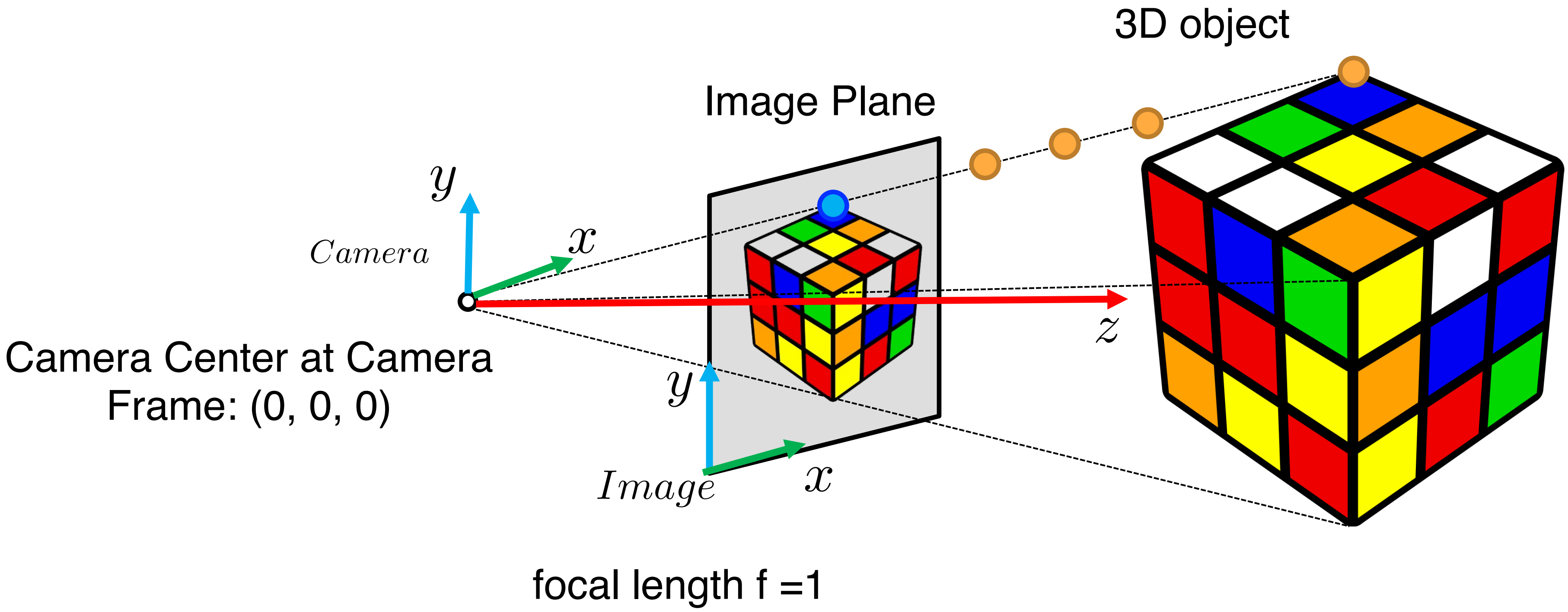
Camera Models

Now let's introduce the camera coordinate frame

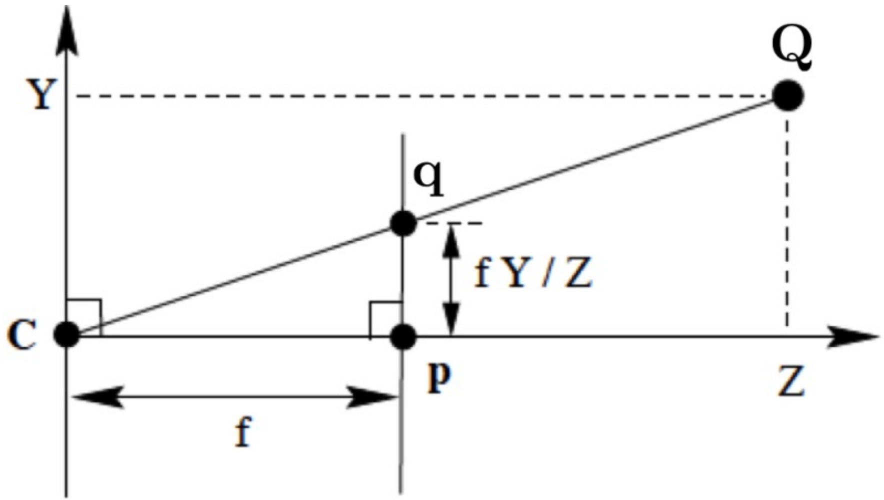


Camera Models

Points along the same projection ray maps to the same 2D point



Similar Triangles



$$[X \quad Y \quad Z]^\top \mapsto [fX/Z \quad fY/Z]^\top$$

Image credit: Andre Zisserman

Similar Triangles

Relationship from similar triangles:

$$[X \quad Y \quad Z]^\top \mapsto [fX/Z \quad fY/Z]^\top$$

General camera model *in homogeneous coordinates*:

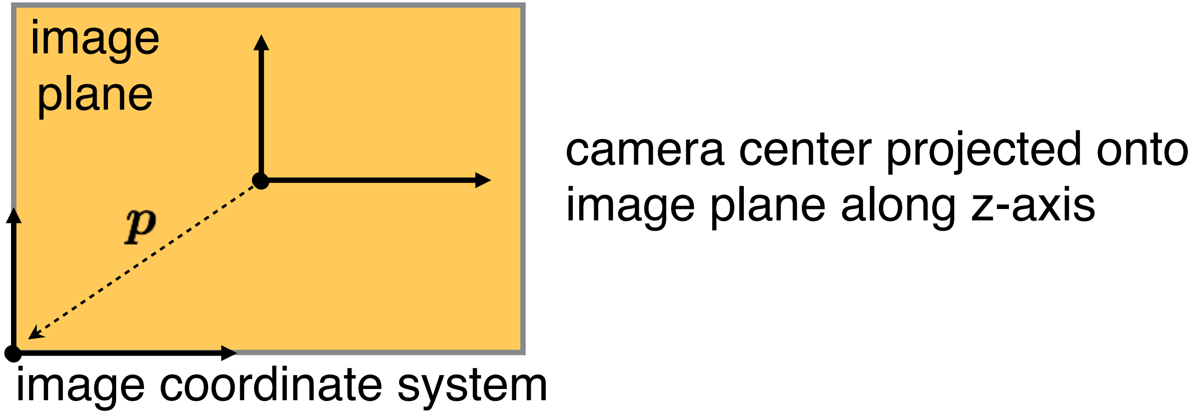
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

What does the pinhole camera projection look like?

$$\mathbf{P} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Perspective Projection

In particular, the camera origin and image origin may be different:

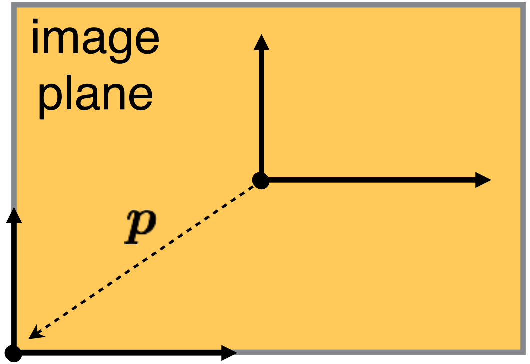


How does the camera matrix change?

$$\mathbf{P} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Perspective Projection

In particular, the camera origin and image origin may be different:



camera center projected onto image plane along z-axis

image coordinate system

principal point

How does the camera matrix change?

$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Decomposition

We can decompose the camera matrix like this:

$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array} \right]$$

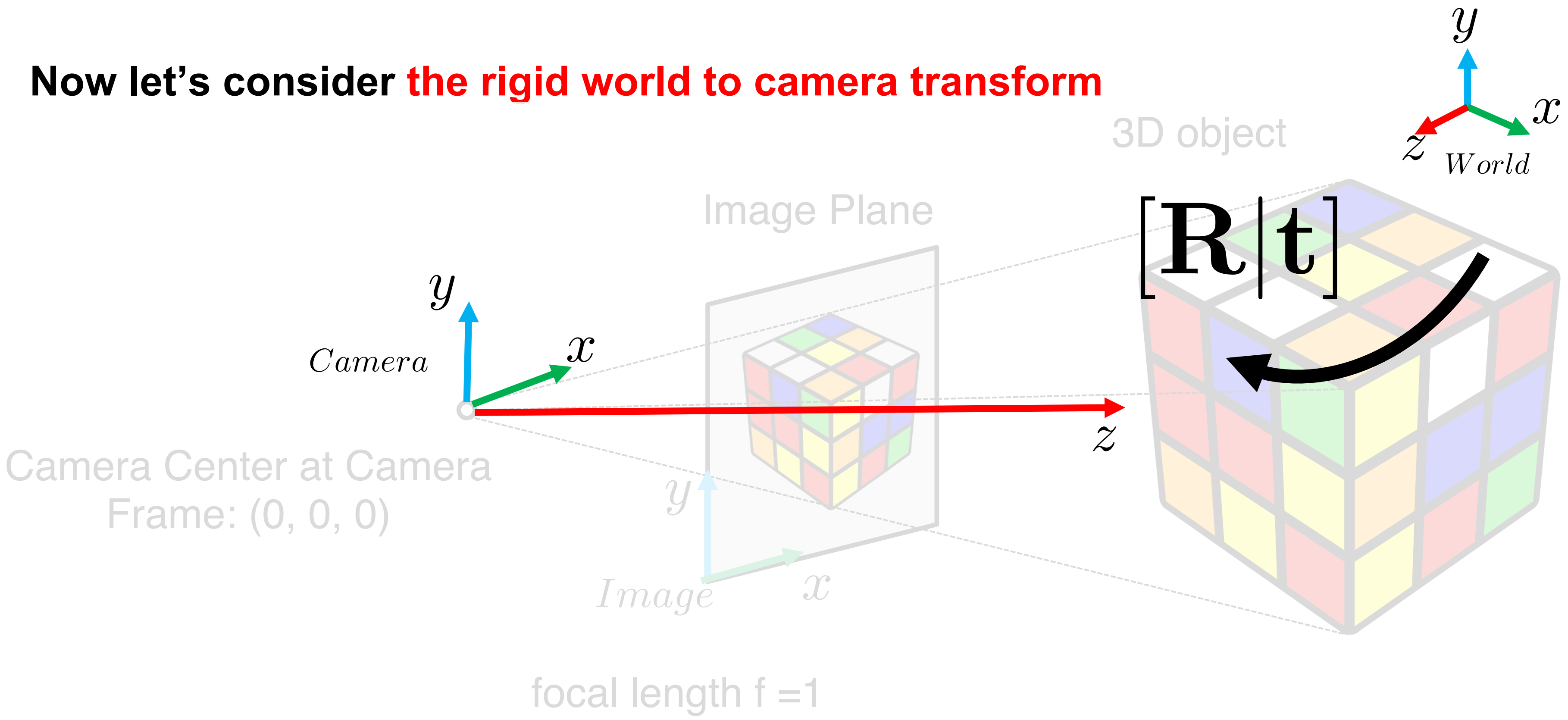
(homogeneous) transformation from 2D to 2D, accounting for not unit focal length and origin shift

(homogeneous) perspective projection from 3D to 2D, assuming principal axis is z-axis, perpendicular to image plane

Also written as: $\mathbf{P} = \mathbf{K}[\mathbf{I}|\mathbf{0}]$ where $\mathbf{K} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}$

Camera Models

Now let's consider the rigid world to camera transform



Projection Matrix from World to Image

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$$

$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc:c} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \end{array}\right]$$

intrinsic parameters

extrinsic parameters

$$\mathbf{R} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \qquad \mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

3D rotation

3D translation

Projection Matrix from World to Image

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$$

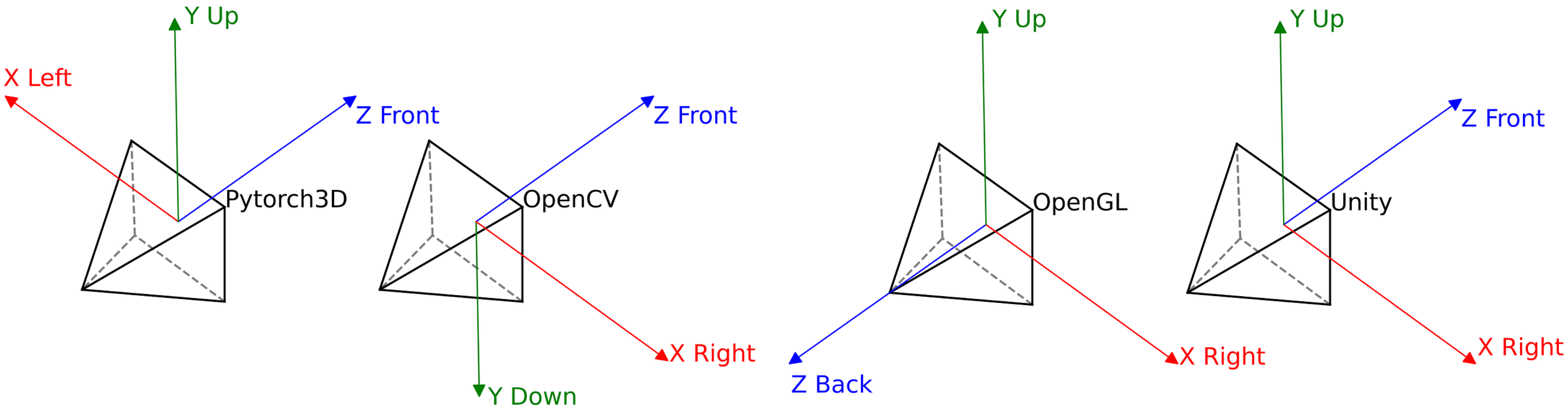
$$\mathbf{P} = \left[\begin{array}{ccc} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{array}\right] \left[\begin{array}{ccc:c} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \end{array}\right]$$

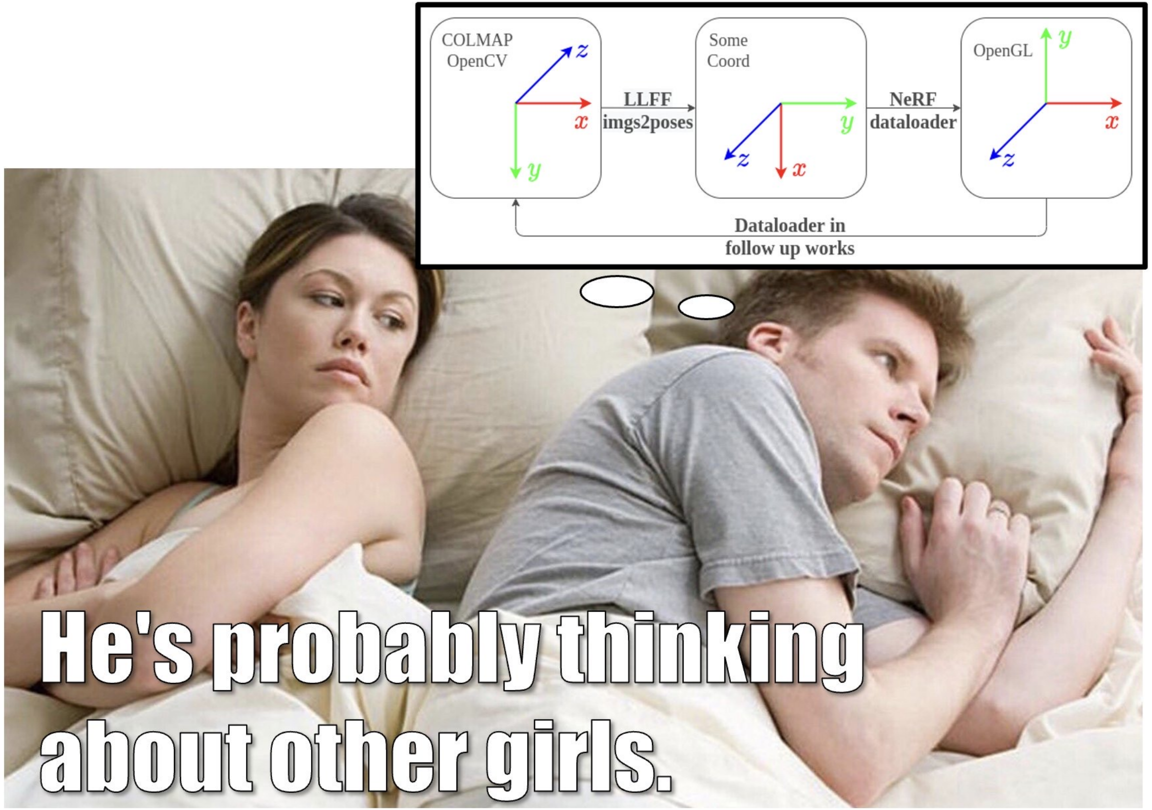
intrinsic parameters | extrinsic parameters

$$\mathbf{R} = \left[\begin{array}{ccc} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{array}\right] \qquad \mathbf{t} = \left[\begin{array}{c} t_1 \\ t_2 \\ t_3 \end{array}\right]$$

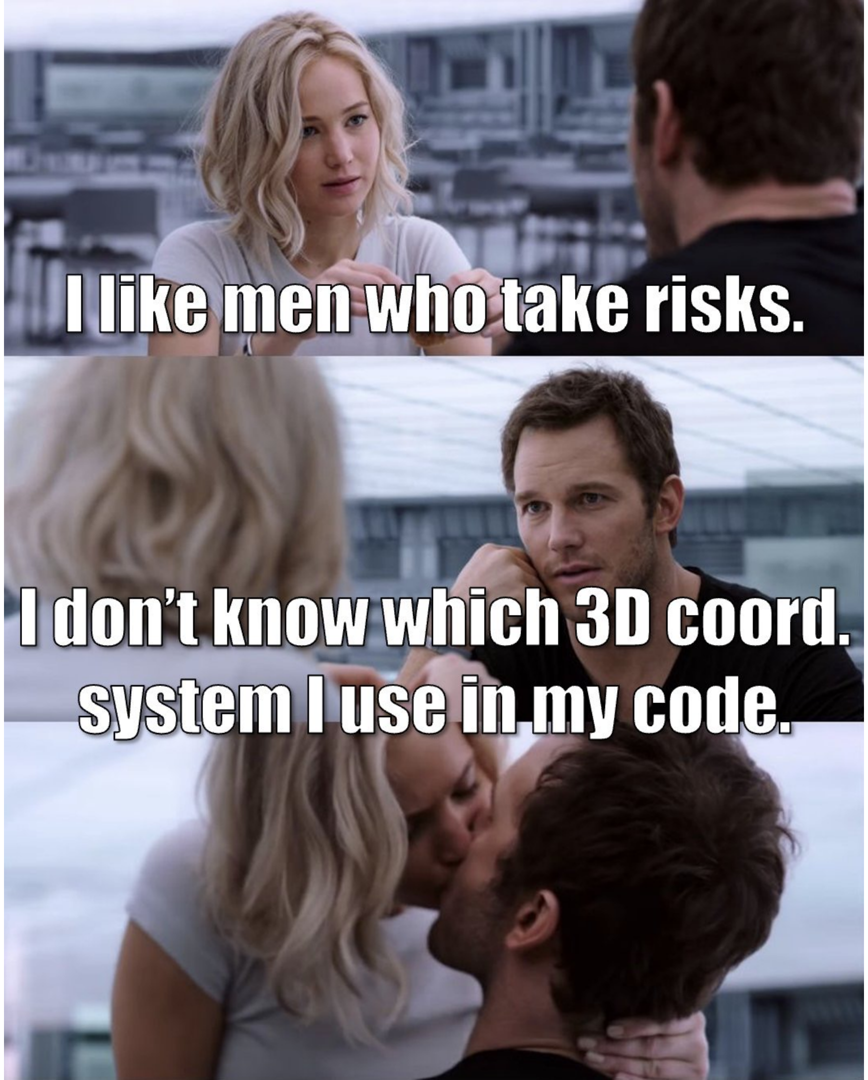
3D rotation | 3D translation

Be careful and camera coordinate definition!





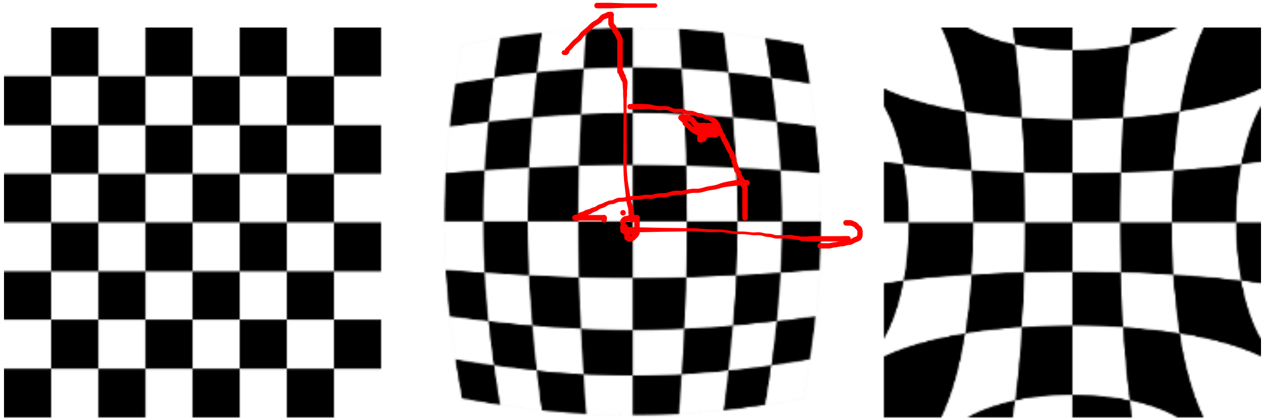
https://x.com/3dvconf



https://x.com/3dvconf

Camera Distortion

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



No distortion

Positive radial distortion (Barrel distortion)

Negative radial distortion (Pincushion distortion)

Image credit: OpenCV

Camera Distortion

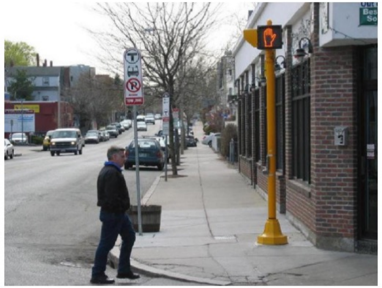
- Remember to **cv2.undistort** the image if you want to reason in 3D.

before

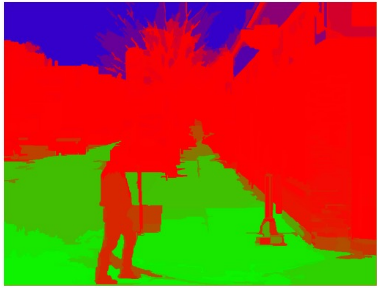
after

Image credit: OpenCV

Understanding perspectives helps recognition

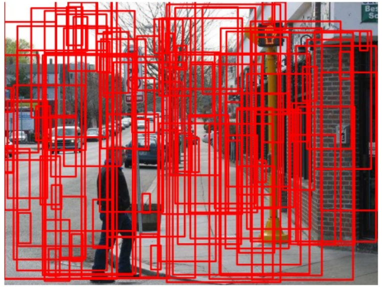


(a) input image

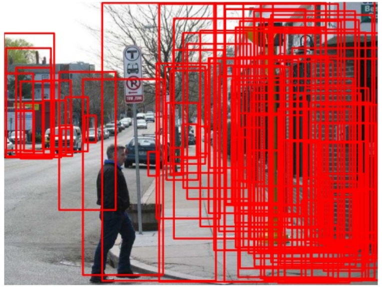


(c) surface orientation estimate

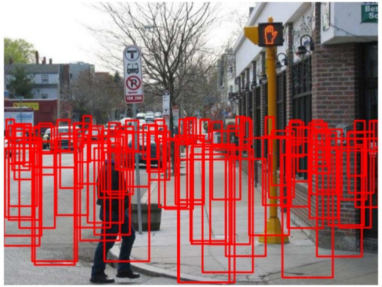
(e) P(viewpoint | objects)



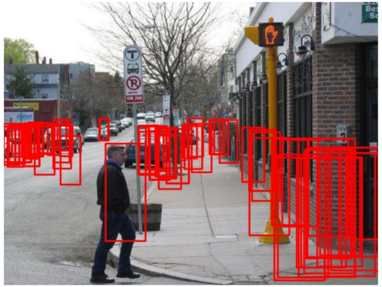
(b) P(person) = uniform



(d) P(person | geometry)



(f) P(person | viewpoint)



(g) P(person|viewpoint,geometry)

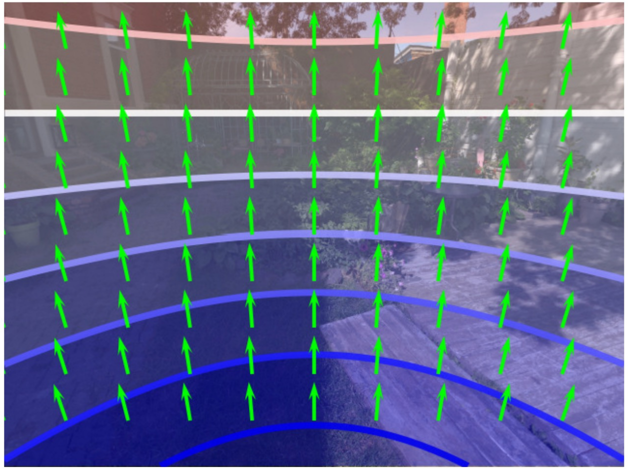
D. Hoiem, A.A. Efros, and M. Hebert, "Putting Objects in Perspective", CVPR 2006.

What’s new: more generic (but over-parameterized) camera model

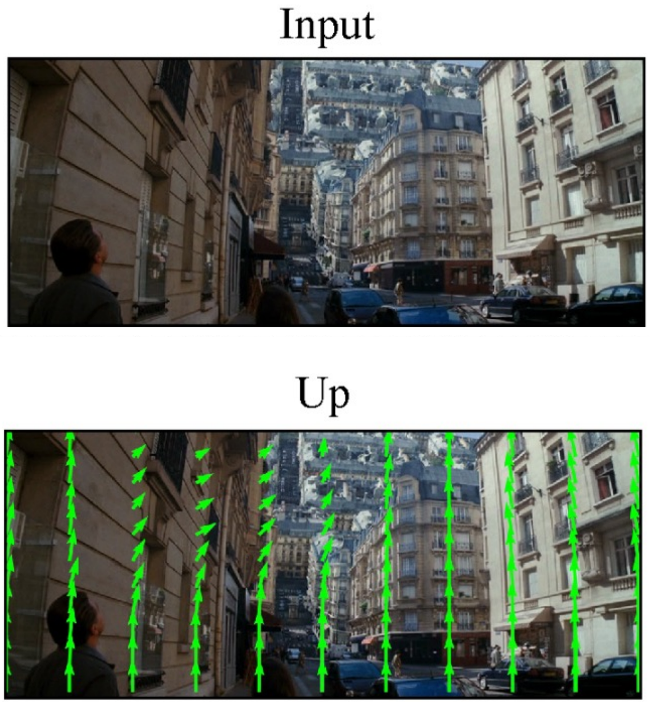
Perspective Fields on Pinhole Camera

Check out how Perspective Fields change w.r.t. pinhole camera parameters.

Roll 0
Pitch -20
FoV 70



For each pixel location, the Perspective Field consists of a unit ***Up-vector*** and ***Latitude***. The ***Up-vector*** is the projection of the up direction, shown in **Green** arrows. In perspective projection, it points to the vertical vanishing point. The ***Latitude*** of each pixel is defined as the angle between the incoming ray and the horizontal plane. We show it using contour line: -π/2 π/2. Note 0° is at the horizon.



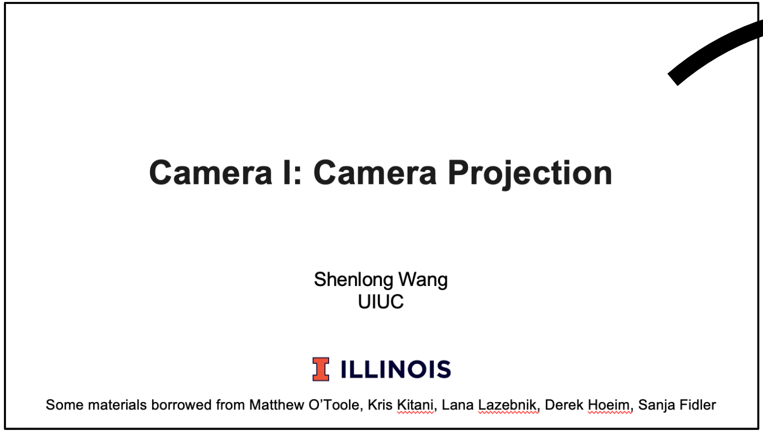
Linyi Jin, Jianming Zhang, Yannick Hold-Geoffroy, Oliver Wang, Kevin Matzen, Matthew Sticha, David F. Fouhey Perspective Fields for Single Image Camera Calibration. CVPR 2023.

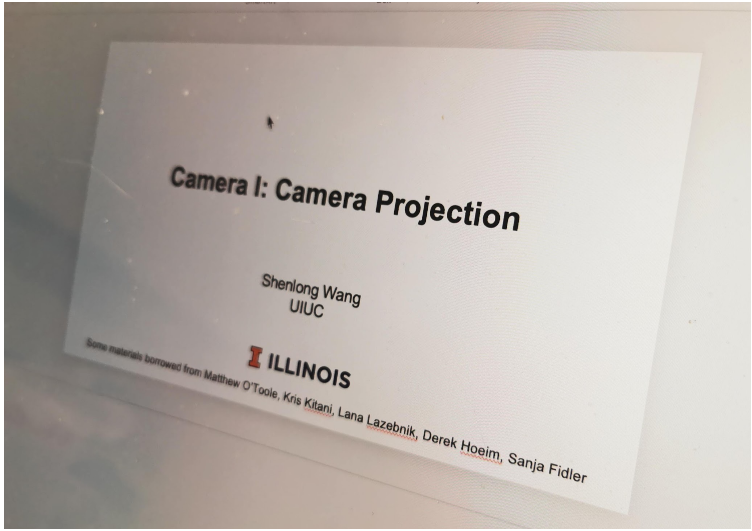
How to fail a 3D vision project?

- Use addition for rotation composition.
- Transpose a rigid transform and pretend you did an inversion.
- Do not know which 3D coordinate system was used.
- Use distorted images.
- ... many others

Homography

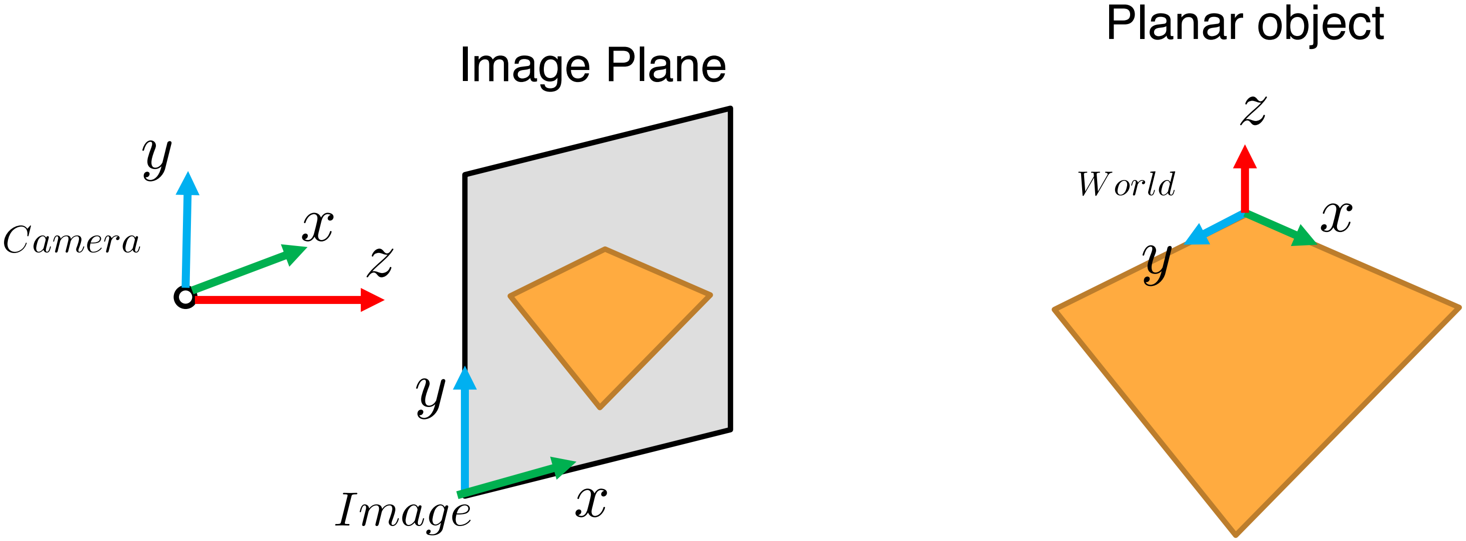
What if a planar object projected onto the camera plane?



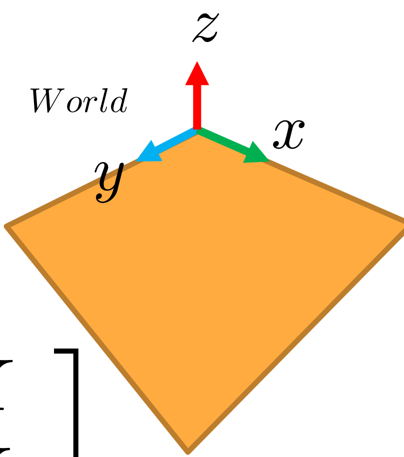


Homography

What if a planar object projected onto the camera plane?



Homography



What if a planar object projected onto the camera plane?

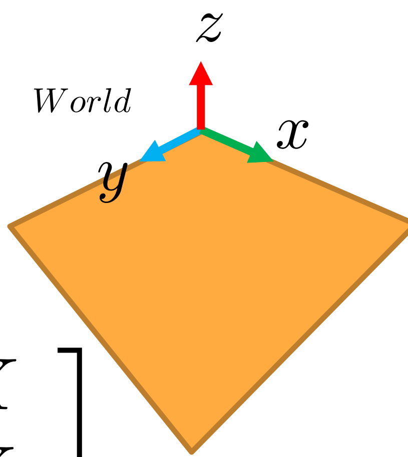
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

homogeneous image coordinates
3 x 1

camera projection matrix
3 x 4

homogeneous world coordinates
4 x 1

Homography



What if a planar object projected onto the camera plane?

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

homogeneous image coordinates
3 x 1

camera projection matrix
3 x 4

homogeneous world coordinates
4 x 1

Homography

What if a planar object projected onto the camera plane?

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

homogeneous
image coordinates
3 x 1

camera projection
matrix
3 x 4

homogeneous
world coordinates
4 x 1

Homography

What if a planar object projected onto the camera plane?

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

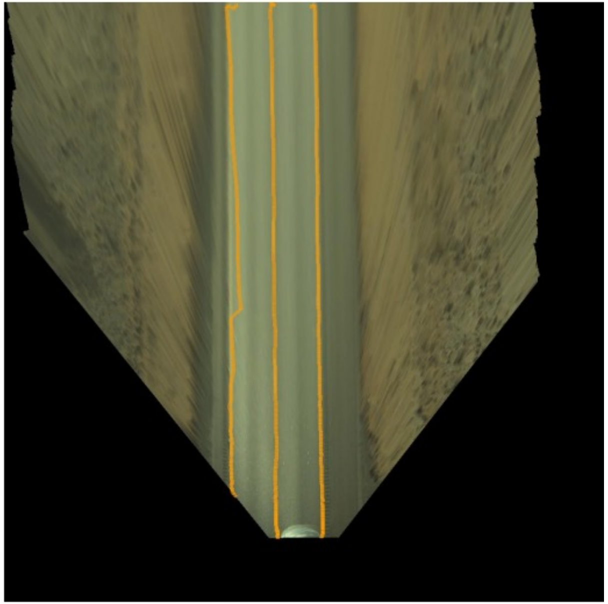
homogeneous image coordinates 3 x 1	homography matrix 3 x 4	homogeneous planar coordinates 3 x 1

Homography

NFL

Nintendo

Homography



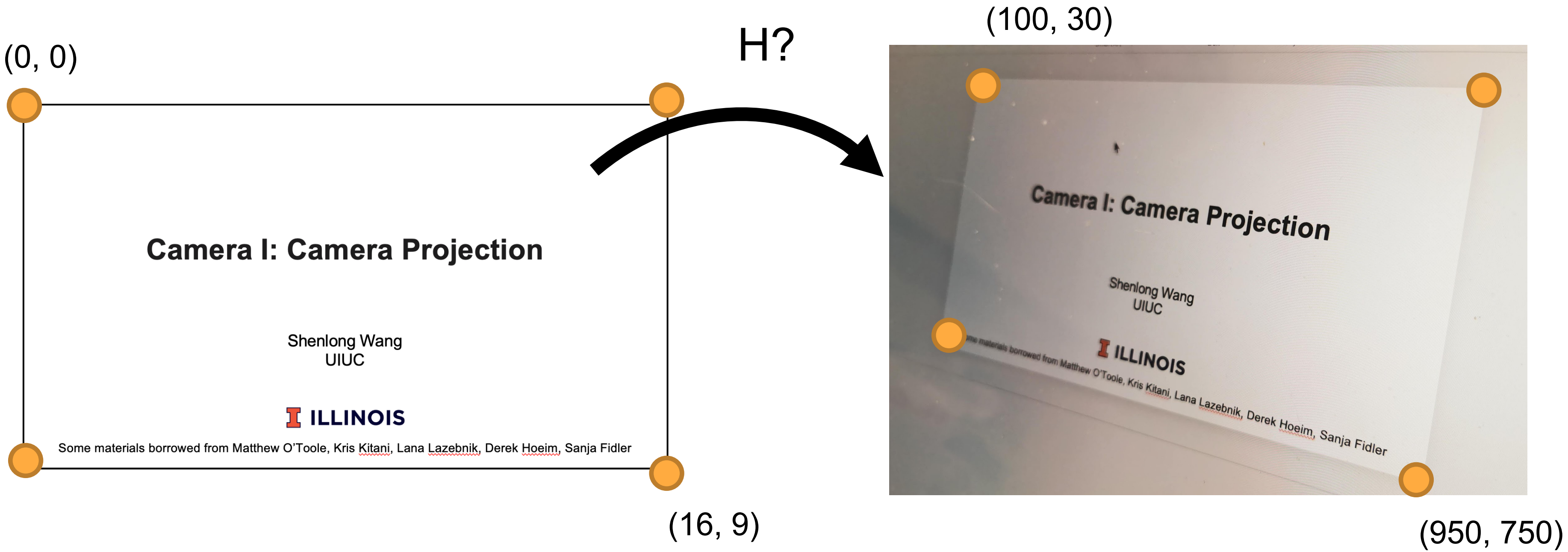
Bai et al. IROS 18

Homography

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

$$x' = \frac{h_1 X + h_2 Y + h_3}{h_7 X + h_8 Y + h_9} \qquad y' = \frac{h_4 X + h_5 Y + h_6}{h_7 X + h_8 Y + h_9}$$

Homography



(0, 0)
Camera I: Camera Projection
Shenlong Wang
UIUC
ILLINOIS
Some materials borrowed from Matthew O'Toole, Kris Kitani, Lana Lazebnik, Derek Hoeim, Sanja Fidler
(16, 9)
H?
(100, 30)
Camera I: Camera Projection
Shenlong Wang
UIUC
ILLINOIS
(950, 750)

Homography

$$x' = \frac{h_1 X + h_2 Y + h_3}{h_7 X + h_8 Y + h_9} \qquad y' = \frac{h_4 X + h_5 Y + h_6}{h_7 X + h_8 Y + h_9}$$

$$h_1 X + h_2 Y + h_3 - x'(h_7 X + h_8 Y + h_9) = 0$$
$$h_4 X + h_5 Y + h_7 - y'(h_7 X + h_8 Y + h_9) = 0$$

Homography

$$h_1X + h_2Y + h_3 - x'(h_7X + h_8Y + h_9) = 0$$

$$h_4X + h_5Y + h_7 - y'(h_7X + h_8Y + h_9) = 0$$

$$\begin{bmatrix} 0 & 0 & 0 & X & Y & 1 & -y'X & -y'Y & -y' \\ X & Y & 1 & 0 & 0 & 0 & -x'X & -x'Y & -x' \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ ... \\ h_9 \end{bmatrix} = 0$$

Homography

$$\begin{bmatrix} 0 & 0 & 0 & X_1 & Y_1 & 1 & -y'_1X_1 & -y'_1Y_1 & -y'_1 \\ X_1 & Y_1 & 1 & 0 & 0 & 0 & -x'_1X_1 & -x'_1Y_1 & -x'_1 \\ 0 & 0 & 0 & X_2 & Y_2 & 1 & -y'_2X_2 & -y'_2Y_2 & -y'_2 \\ X_2 & Y_2 & 1 & 0 & 0 & 0 & -x'_2X_2 & -x'_2Y_2 & -x'_2 \\ 0 & 0 & 0 & X_3 & Y_3 & 1 & -y'_3X_3 & -y'_3Y_3 & -y'_3 \\ X_3 & Y_3 & 1 & 0 & 0 & 0 & -x'_3X_3 & -x'_3Y_3 & -x'_3 \\ & & & & ... & & & & \\ 0 & 0 & 0 & X_n & Y_n & 1 & -y'_nX_n & -y'_nY_n & -y'_n \\ X_n & Y_n & 1 & 0 & 0 & 0 & -x'_nX_n & -x'_nY_n & -x'_n \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = \mathbf{0}$$

Homography

$$\mathbf{A}\mathbf{h} = \mathbf{0}$$

Homogeneous linear equations

$$\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} = \mathrm{svd}(\mathbf{A})$$

$\mathbf{h}$ last singular vector in V (corresponding to smallest singular values)

Steve Brunton: Singular Value Decomposition: Overview; Wikipedia

Today’s Agenda

- Coordinates & Axis
- Rigid Transforms & Rotations
- Camera Basics
- Perspective Geometry
- Homography

TODOs

- Join Slack: https://shorturl.at/jV1NL
- Fill in a quick survey form: https://forms.gle/mUmMZbx8ZwgUkT5W9
- Mini Quiz 1: Draw a diagram about rotations: https://forms.gle/sF1yLkbgRNmWwcyX7