Computer Vision

Class 01

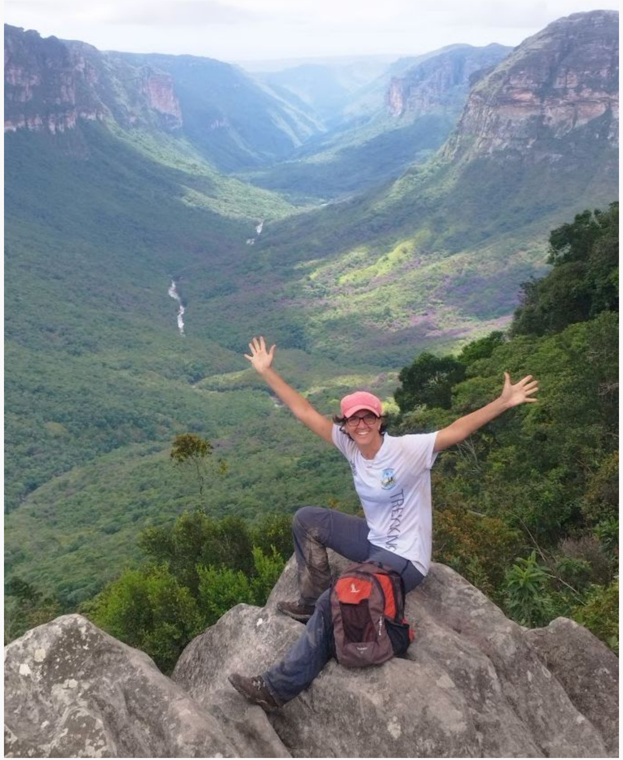
Raquel Frizera Vassallo

Contents of this course

That is what we are going to see in this course

22/09/25 10:30 – 12:30	23/09/25 13:00 – 15:00	29/09/25 13:00 – 17:00	30/09/25 15:00 – 17:00	06/10/25 13:00 – 17:00	10/10/25 10:30 – 12:30	13/10/25 13:00 – 17:00	17/10/25 13:00 – 17:00
Introduction Rigid Body Motion					Stereo Vision		
	Rigid Body Motion Exercise	Pinhole Projection Model Exercise		Homography Exercise		Essential and Fundamental Matrices	Triangulation
		Calibration	Homography	Aruco Exercise		Exercise	Exercise

Who am I?



Raquel Frizera Vassallo

raquel.vassallo@ufes.br

Associate professor at Federal University of Espírito Santo - UFES

Background: Electrical Engineering

Master and PhD - Robotics and Computer Vision

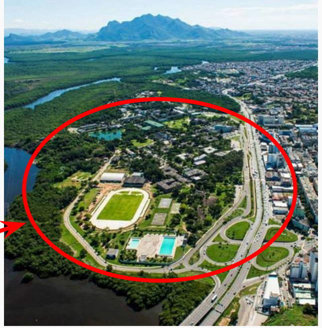
Topics of Interest:

- Computer Vision & Mobile Robotics
- Intelligent or Smart spaces
- Human-robot-environment interaction
- Computer Vision in aerial robotics

- Hiking and Trekking (Hobby)

Where am I from?

UFES
UNIVERSIDADE FEDERAL
DO ESPÍRITO SANTO



Espírito Santo (state) has ~4,1 million inhabitants

Vitória (capital) is an island, with ~345,000 inhabitants and 474 years old.

UFES (Federal University of Espírito Santo):

- Students: 25,000 - undergraduate education, as well as graduate programs (academic and professional master's and doctoral degrees)
- Professors: 2,000
- Administrative staff: 2,500
- Library, museums, theater, sport center, university restaurant

Class 01

01 Introduction

Motivation
Applications
Basic Background
Exercises

02 Rigid Body Motion

Homogeneous Coordinates
Translation and Rotation
Exercises

01

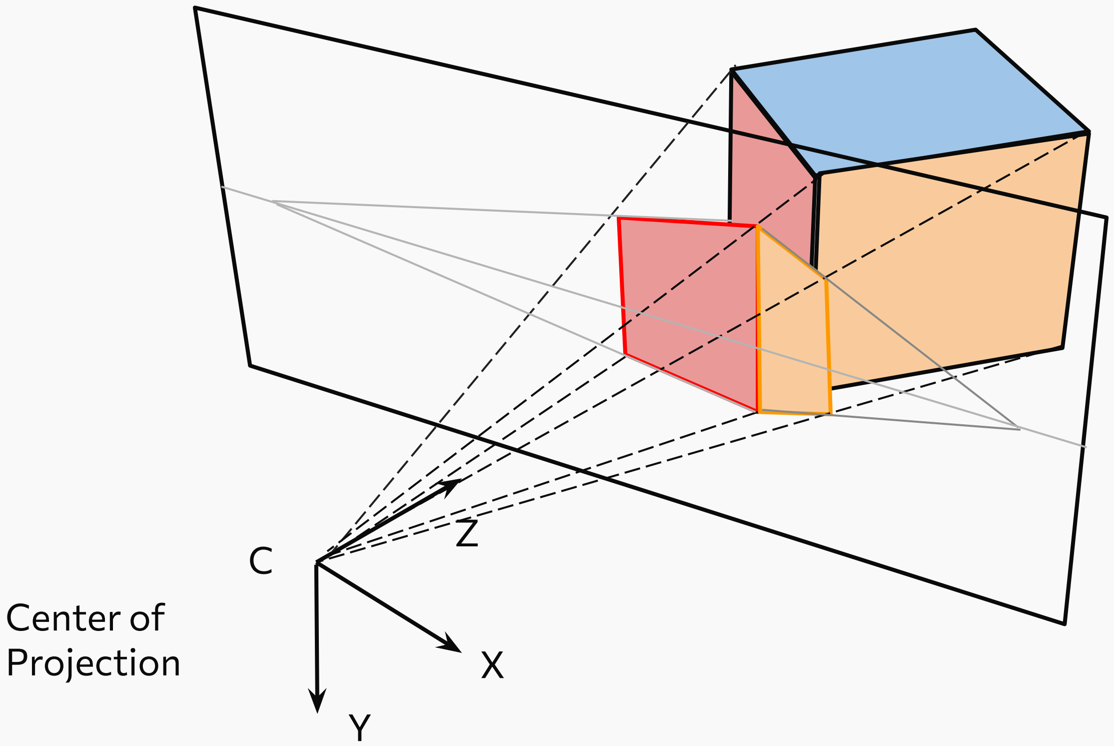
Introduction

Motivation
Applications
Basic Background
Exercises

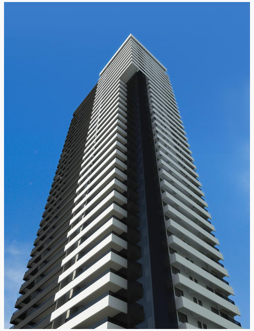
What do we want with Computer Vision?

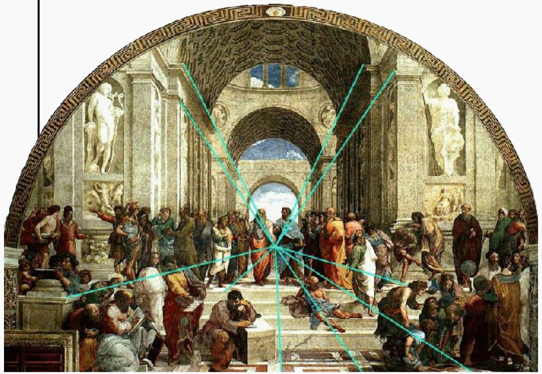
Usually we want to recover interesting information from images or videos that can be used to make decisions, to detect objects, elements, movement or actions, to recover 3D information as shape, size, localisation.

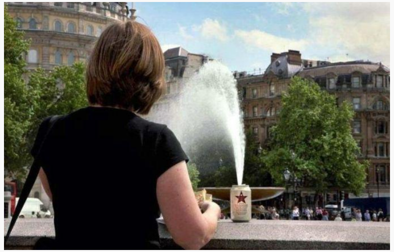
Perspective Vision: One camera



Perspective Vision
One camera

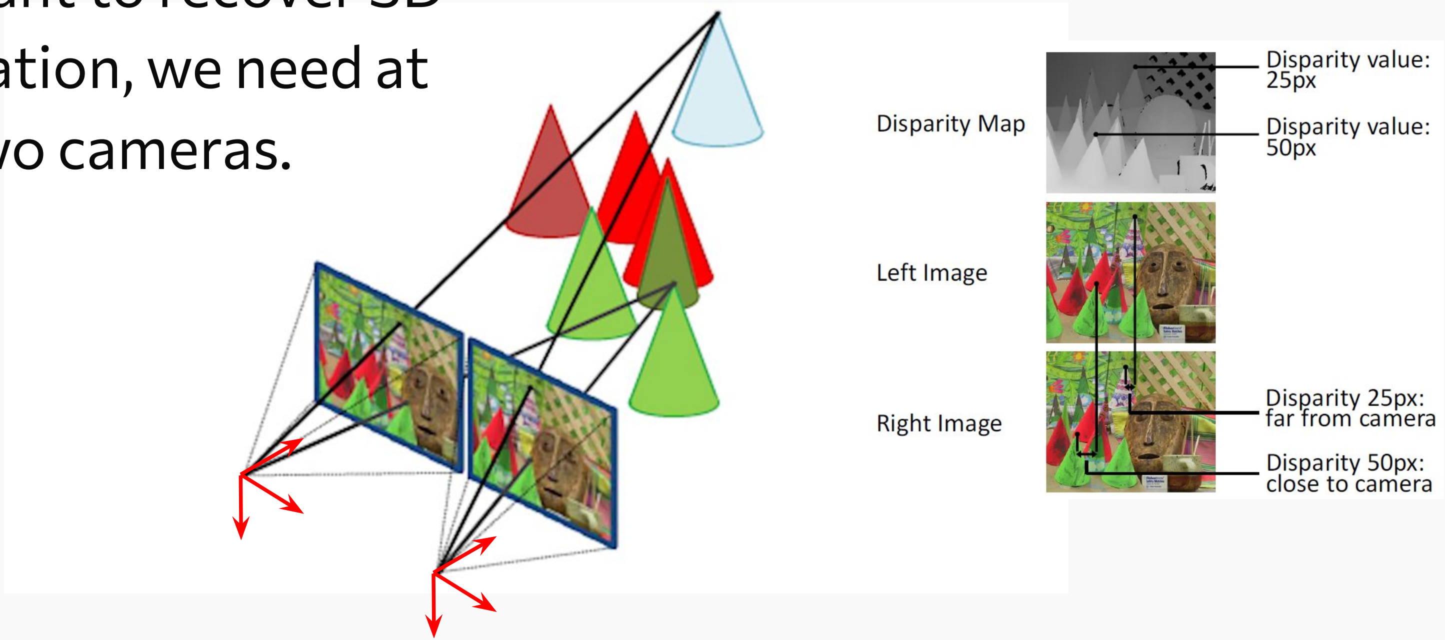






Perspective Vision: Two cameras - Stereo Vision

If we want to recover 3D information, we need at least two cameras.



Some interesting videos

Assumptions

https://www.youtube.com/watch?v=zNbF006Y5x4&authuser=0

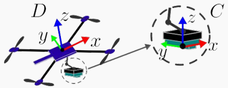
Stereoscopic Vision - Brain Games

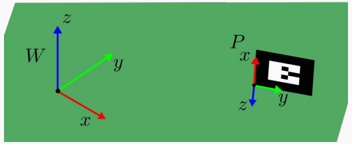
https://www.youtube.com/watch?v=nKdUD8lIGJY

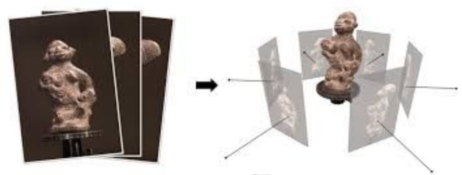
How does Stereo Vision work?

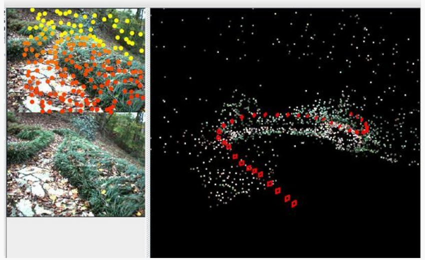
https://www.youtube.com/watch?v=yfjMJfXMBcY

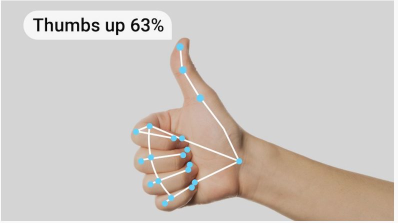
Applications

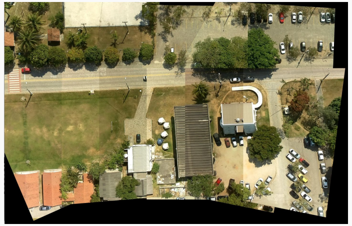


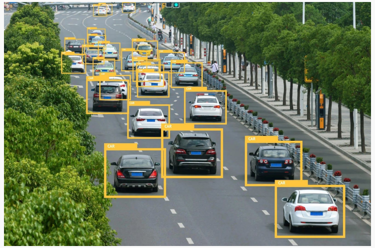


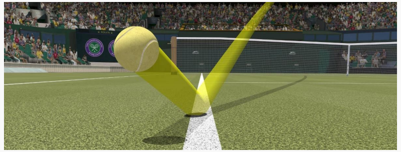






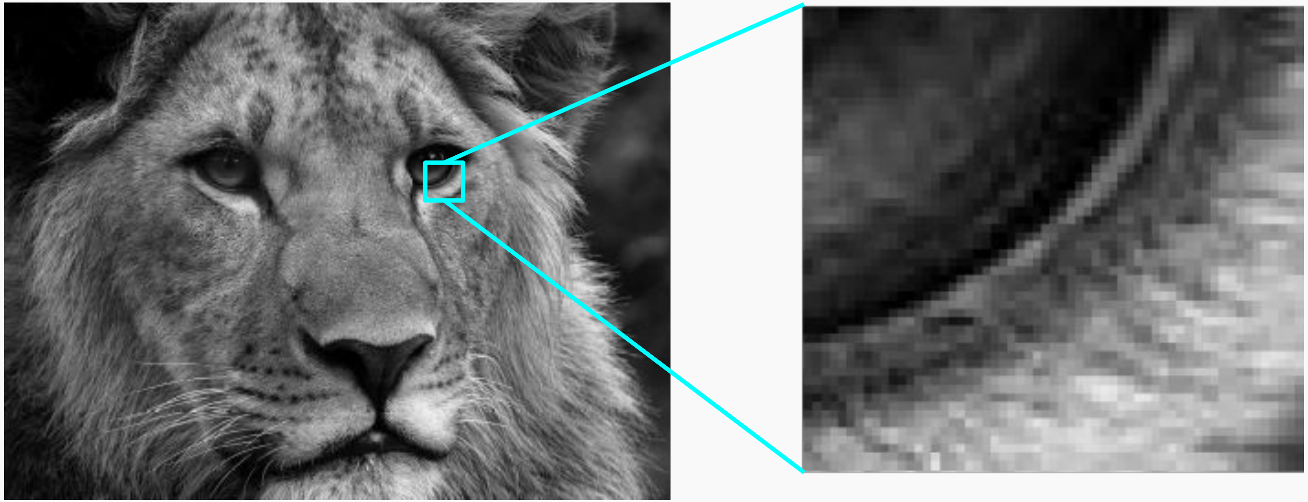




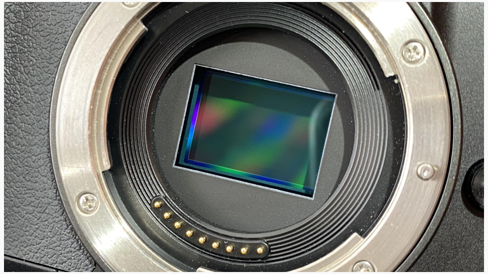


Basic Background
Gray levels and pixel

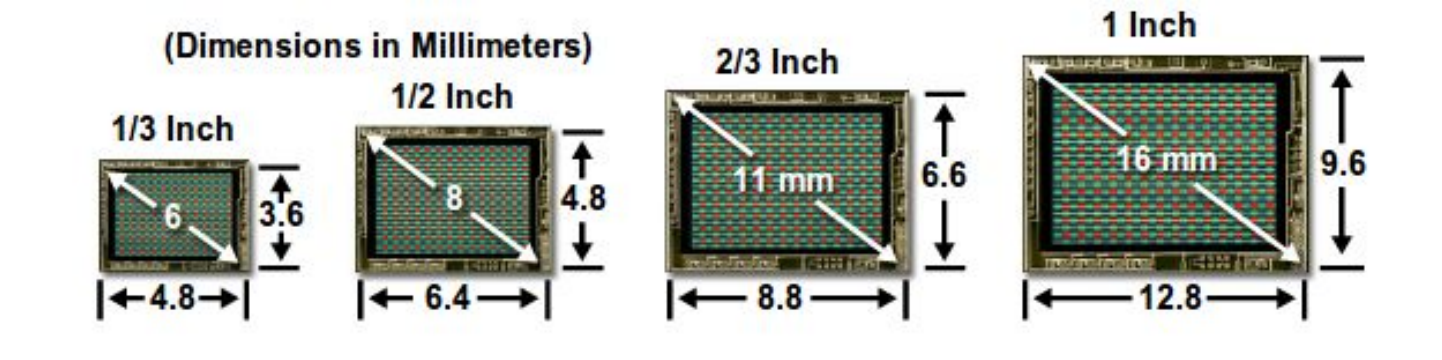
- Gray levels: 0 is "Black", 255 is "White", numbers in between are levels of gray
- Pixel: each number is shown as a shaded small square



Basic Background
Digital Images



- Sampling and integrating continuous (analog) data in a spatial discrete domain
- Uses one or several matrix: $M_{cols} \times N_{rows}$ of sensor elements (phototransistors) - CCD or CMOS
- Sensor cell converts measured light into an electric charge coded as a number
- Typically configured with square pixels assembled into rectangular area arrays, with an aspect ratio of 4:3 being most common



Basic Background
Image Resolution and Bit Depth

- **Aspect Ratio:** Each phototransistor is a rectangular cell.
  Ideally, the aspect ratio should be equal to 1 (i.e. square cells)
- **Resolution:** Number of sensor elements.
  Example: 4 Mpixel camera (4.000.000 pixel) in some image format. Without further mentioning, the number of pixels means “color pixel”
- **Bit Depth**: Number of bits per pixel.
  Common goal: more than just 8 bits per pixel value in one channel.

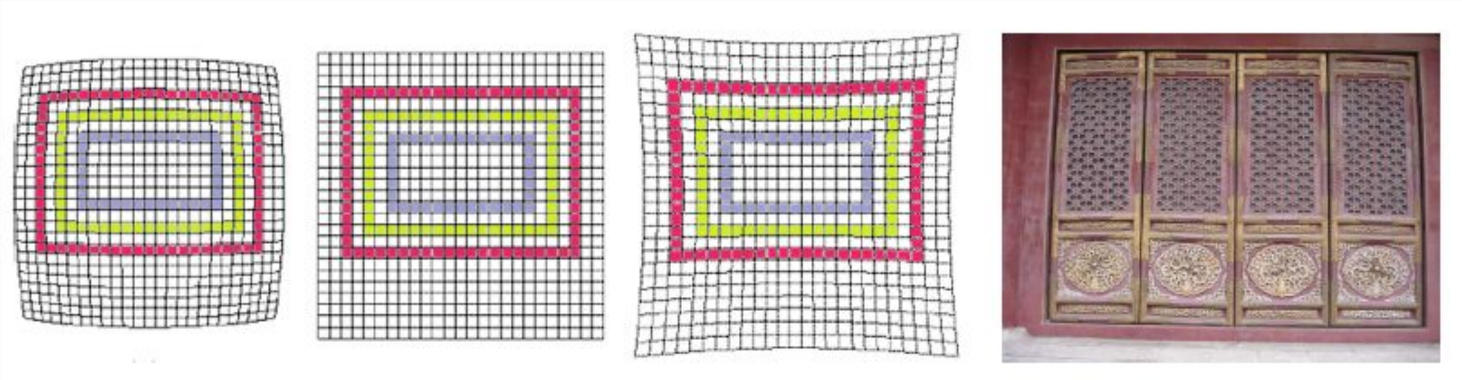
  E.g. 16 bits per pixel in a gray-level image for motion or stereo analysis.

Basic Background
Image Values

- A binary image has two values, traditionally 0 = white and 1 = black (black objects on white background or vice-versa)
- Color images in the RGB color model have 3 channels: red, green, and blue
  Image values are represented as $(u_R; u_G; u_B)$
- Values in each channel are in the set $\{0; 1; ...; G_{max}\}$ (like a gray-value image).

Basic Background
Lens Distortion

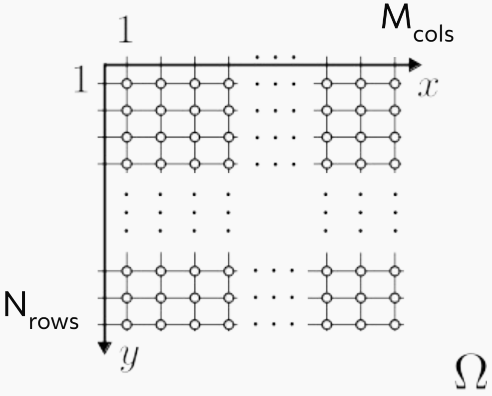
- Optic lenses contribute to radial lens distortion on the projection process
- Barrel transform or pincushion transform



Left to right: Barrel transform, ideal rectangular image, pincushion transform, and projective and lens distortion combined in one image

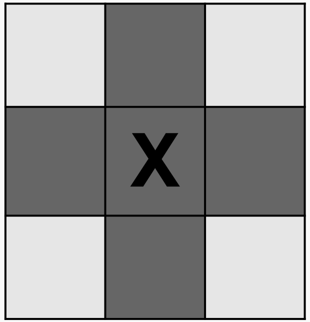
Basic Background
Image Coordinate System

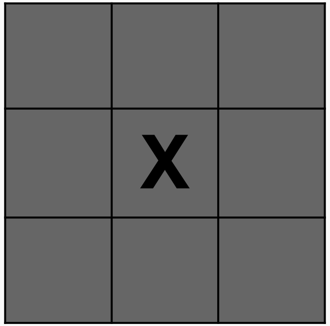
- The origin is at the top left corner
- The x-axis increases from left to right
  Each column contains grid points: $\{(x, 1), (x, 2), \dots, (x, N_{rows})\}$
- The y-axis increases from top to bottom
  Each row contains grid points: $\{(1, y), (2, y), \dots, (M_{cols}, y)\}$



Basic Background
Adjacency

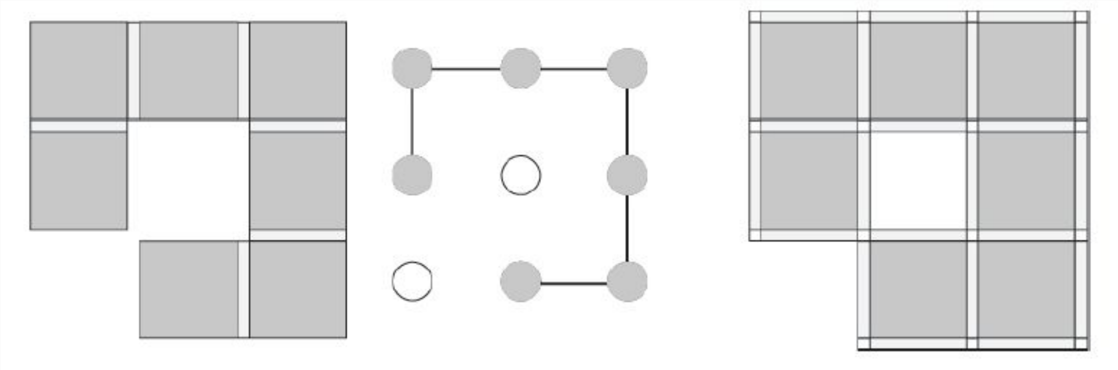
- Two types: 4 or 8-adjacency.
- Two pixel locations p and q are adjacent if:
    - 4-adjacency: their tiny shaded squares share an edge;
    - 8-adjacency: their tiny shaded squares share an edge or corner.





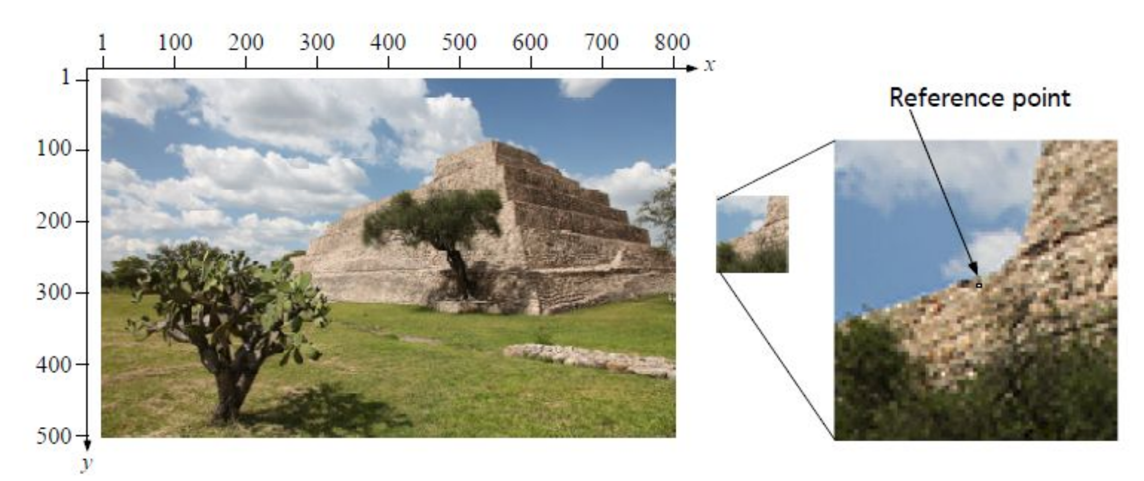
Basic Background
Connected or not connected sets of pixels

- Depends on adjacency
- Left and middle: considering 4-adjacency, the two white pixels are not connected. We have an open blob
- Right: considering 8-adjacency, the two white pixels are “crossing” this loop. We have a closed blob.



Basic Background
Image Window

- The reference point of a window is usually at the window's center
- A window $W_p(I)$ is a subimage of image I of size M x N positioned with respect to a reference point p
- Default: odd number M = N and p at the center of the window



02

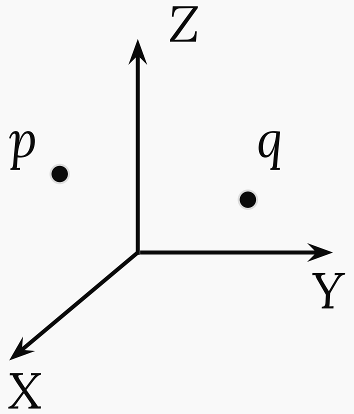
Rigid Body Motion

Homogeneous coordinates
Translation
Rotation

Points and Vectors

- A point can be defined by three Cartesian Coordinates in the Euclidean Space.

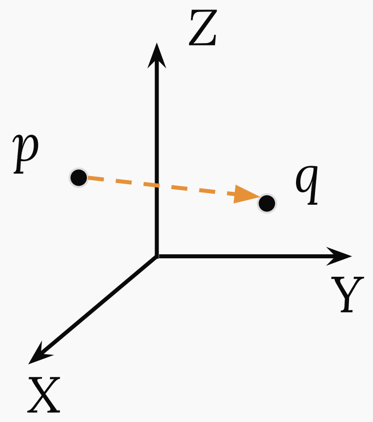
$$p = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} \in \mathbb{R}^3 \qquad q = \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} \in \mathbb{R}^3$$



Points and Vectors

- A point can be defined by three Cartesian Coordinates in the Euclidean Space.
- A vector can be defined by a pair of points (p,q).
- Both of them are represented by three coordinates.

$$p = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} \in \mathbb{R}^3 \qquad q = \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} \in \mathbb{R}^3$$



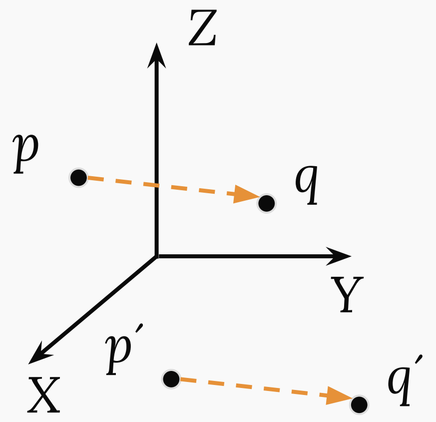
$$v = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} X_2 - X_1 \\ Y_2 - Y_1 \\ Z_2 - Z_1 \end{bmatrix} \in \mathbb{R}^3$$

Points and Vectors

- A point can be defined by three Cartesian Coordinates in the Euclidean Space.
- A vector can be defined by a pair of points (p,q).
- Both of them are represented by three coordinates.

$$p = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} \in \mathbb{R}^3 \qquad q = \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} \in \mathbb{R}^3$$

$$v = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} X_2 - X_1 \\ Y_2 - Y_1 \\ Z_2 - Z_1 \end{bmatrix} \in \mathbb{R}^3$$

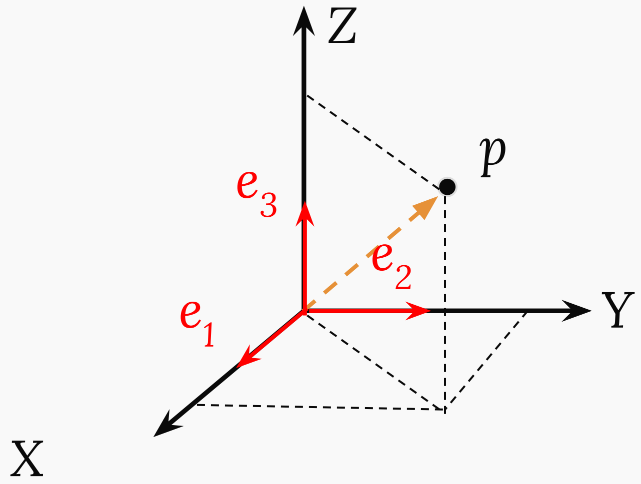


IMPORTANT:
Vector and Point are different!

Euclidean Space

Base Vectors:

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$



Coordinates of point p in 3D space: $$p = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \in \mathbb{R}^3$$

Inner Product of two vectors

$$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad v = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

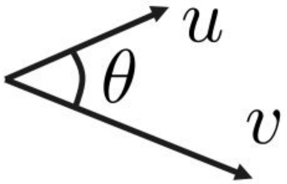
$$\langle u, v \rangle \doteq u^T v = u_1 v_1 + u_2 v_2 + u_3 v_3$$

Used to measure distances and vector norm

$$\|u\| \doteq \sqrt{u^T u} = \sqrt{u_1^2 + u_2^2 + u_3^2}$$

Used to measure angles

$$\cos(\theta) = \frac{\langle u, v \rangle}{\|u\| \, \|v\|}$$

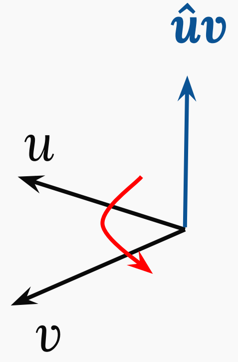


If the inner product of two vectors is zero, then the vectors are orthogonal to each other

Cross Product of two vectors

$$u \times v \doteq \begin{bmatrix} u_2v_3 - u_3v_2 \\ u_3v_1 - u_1v_3 \\ u_1v_2 - u_2v_1 \end{bmatrix} \in \mathbb{R}^3$$

Resulting vector is perpendicular to the original ones



$$u \times v \doteq \hat{u}v, \quad u, v \in \mathbb{R}^3$$

$$\hat{u} = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 3}$$

Attention:

$$u \times (\alpha v + \beta w) = \alpha u \times v + \beta u \times w, \forall \alpha, \beta \in \mathbb{R}.$$
$$\langle u \times v, u \rangle = \langle u \times v, v \rangle = 0$$
$$u \times v = -v \times u$$

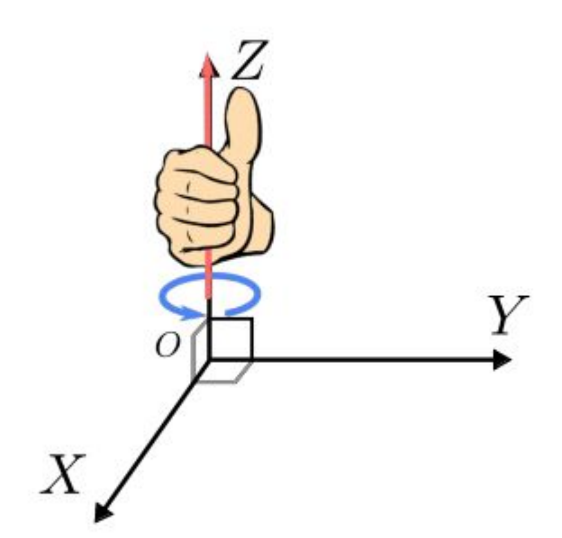
$\hat{u}$ is a skew-symmetric matrix. Thus $\hat{u}$ is square and $\hat{u}^T = -\hat{u}$

Coordinate Axis – Right-hand Rule

$$e_1 = [1, 0, 0]^T \qquad e_1 \times e_2 = e_3$$

$$e_2 = [0, 1, 0]^T \qquad e_2 \times e_3 = e_1$$

$$e_3 = [0, 0, 1]^T \qquad e_3 \times e_1 = e_2$$

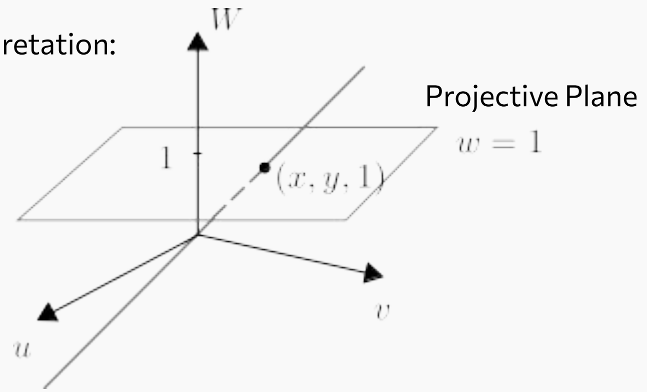


Homogeneous Coordinates for the 2D Euclidean Space

- (wx, wy, w) represent the same 2D point for different w
- (wx, wy, w) divided by w gives (x, y, 1) → Cartesian Coordinates (x, y)
- (0, 0, 0) does not exist
- If the last coordinate is 0 (zero) → point at infinity (ideal point) or a vector (direction)

$$\vec{v} = q - p = \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} - \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ 0 \end{bmatrix}$$

Graphical interpretation:



Homogeneous Coordinates for the 3D Euclidean Space

- (wx, wy, wz, w) represent the same 3D point for different w
- (wx, wy, wz, w) divided by w gives (x, y, z, 1) → Cartesian Coordinates (x, y, z)
- (0, 0, 0, 0) does not exist
- If the last coordinate is 0 (zero) → point at infinity (ideal point) or a vector (direction)

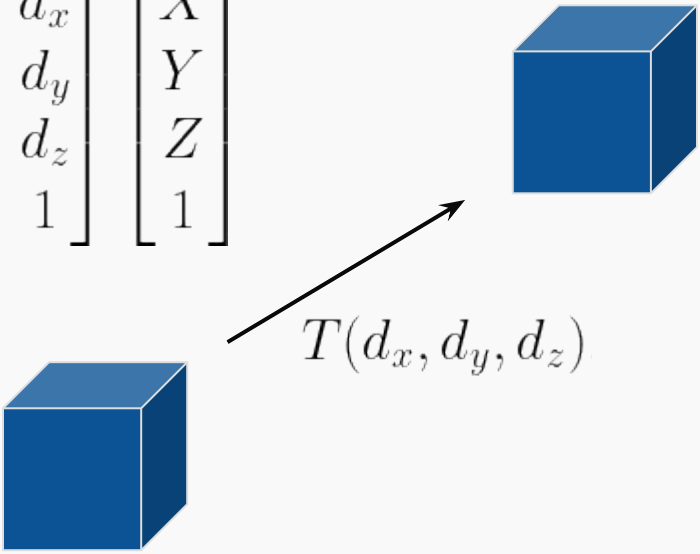
$$\vec{v} = q - p = \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} - \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ 0 \end{bmatrix}$$

Benefits:
Homogeneous coordinates allow performing point transformations as multiplication of matrices.

Translation

$$
\begin{aligned}
X' &= X + d_x \\
Y' &= Y + d_y \\
Z' &= Z + d_z
\end{aligned}
\qquad
\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}
$$

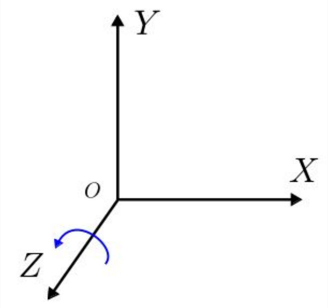
$$\boxed{P' = T(d_x, d_y, d_z).P}$$



Important: $T(-dx, -dy, -dz) = T(dx, dy, dz)^{-1}$

Rotation

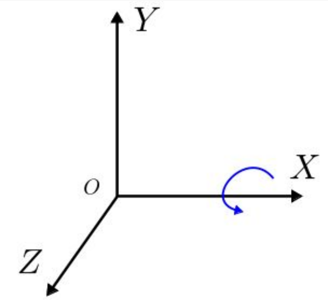
Around Z-axis



$$P' = R_z(\theta).P$$

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 & 0 \\ sin(\theta) & cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

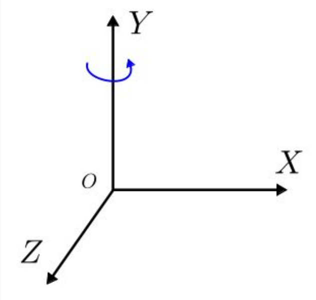
Around X-axis



$$P' = R_x(\theta).P$$

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) & 0 \\ 0 & sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Around Y-axis



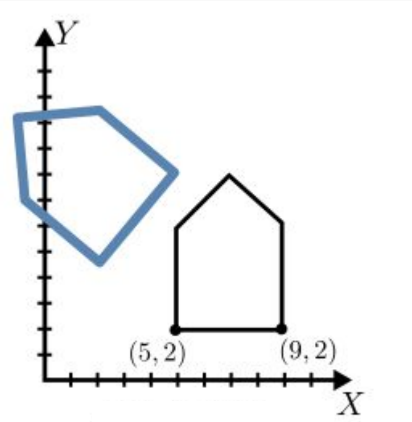
$$P' = R_y(\theta).P$$

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} cos(\theta) & 0 & sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(\theta) & 0 & cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

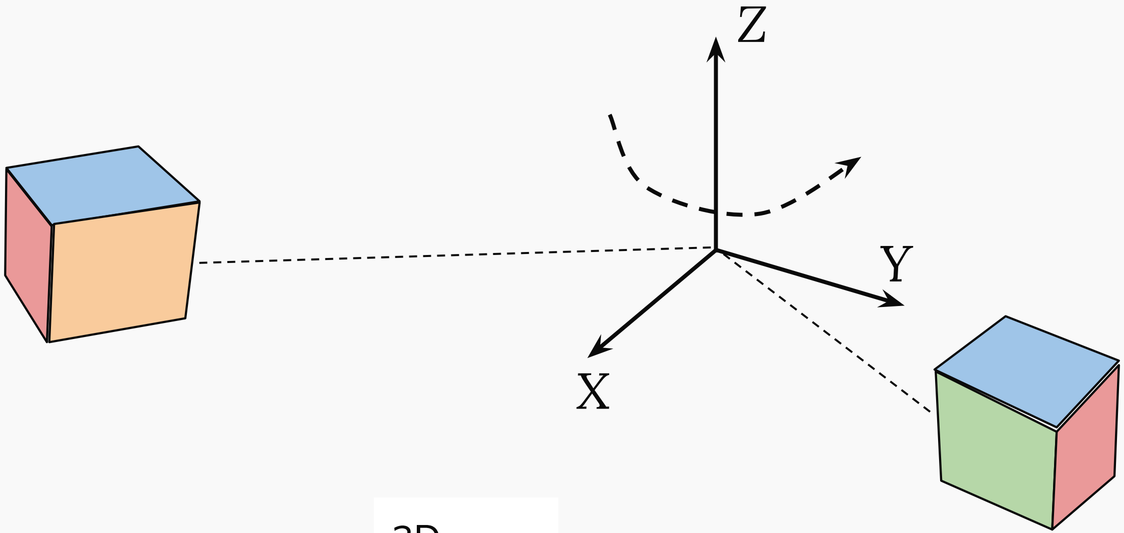
Important: $R_z(-\theta) = R_z(\theta)^{-1} = R_z(\theta)^T$ $\quad R_x(-\theta) = R_x(\theta)^{-1} = R_x(\theta)^T$ $\quad R_y(-\theta) = R_y(\theta)^{-1} = R_y(\theta)^T$

Rotation

ATTENTION: Rotation around the frame axes happens with respect to the origin

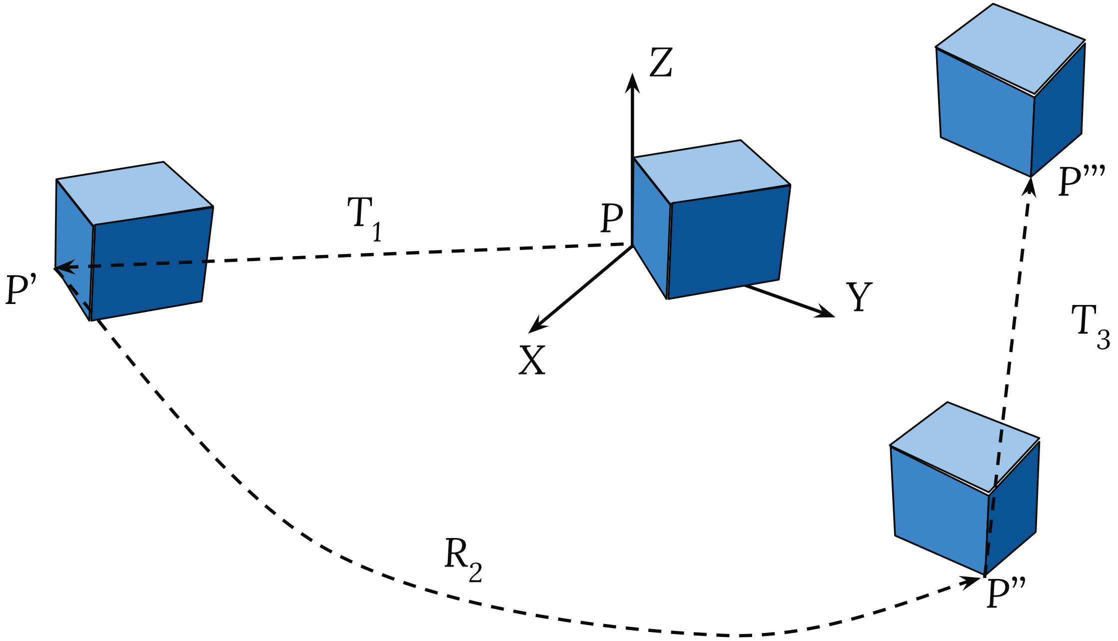


2D



3D

Composition of transformations



$$P' = T_1.P$$

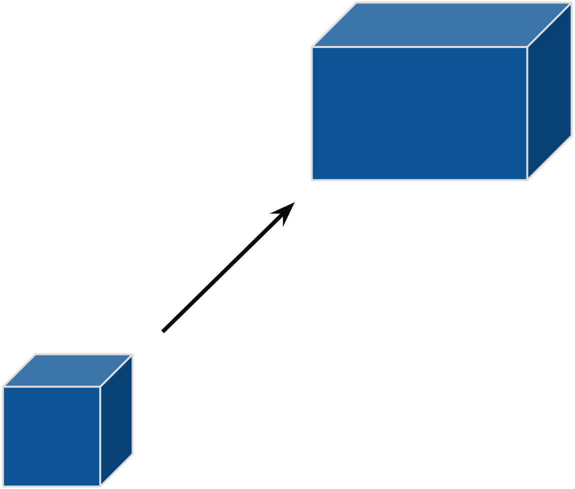
$$P'' = R_2.P'$$

$$P''' = T_3.P''$$

$$P''' = \underbrace{T_3.R_2.T_1}.P$$

$$P''' = M.P$$

Scaling

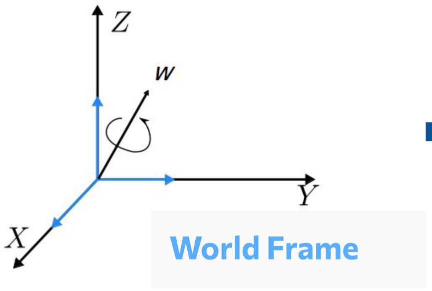


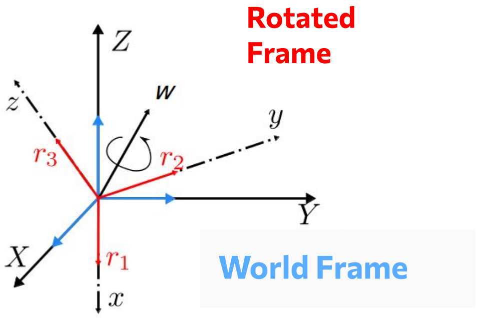
$$P' = S(s_x, s_y, s_z).P$$

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Rotation

Rotation around a general axis





r_1, r_2 and r_3 form an orthogonal frame.

Orthogonal Matrix $\rightarrow R^T = R^{-1}$
Det = +1 $\rightarrow$ Special Orthogonal

$$R^T R = I, det(R) = +1$$

A point P_{world} in the fixed frame can be represented as a linear combination of r_1, r_2 and r_3 (Rotated frame).

$$P_{world} = X_R.r_1 + Y_R.r_2 + Z_R.r_3$$

where X_R, Y_R and Z_R represent the coordinates of the point P in the rotated frame.

$$P_{world} = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix} \begin{bmatrix} X_R \\ Y_R \\ Z_R \end{bmatrix} = R.P_R$$

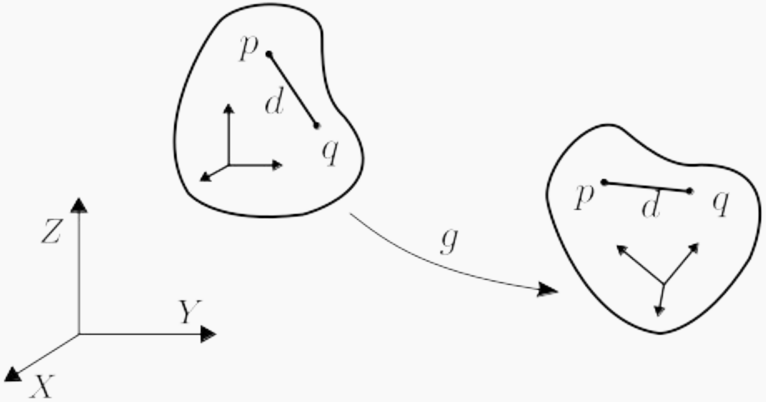
$$R \doteq [r_1, r_2, r_3] \in \mathbb{R}^{3\times 3}$$

$$\boxed{P_{world} = R.P_R} \Longleftrightarrow \boxed{P_R = R^{-1} P_{world}}$$

ATTENTION

All transformations defined in 3D can be applied in 2D, except for rotation around a general rotation axis. In this case, the rotation always occurs within the plane, that is, around an axis perpendicular to the given 2D plane.

Rigid-body Motion



The distance between any two points on the body does not change over time as the object moves.

It is sufficient to specify the motion of one point, and the motion of the coordinate frame attached to that point: Body Refrence Frame

Preserves distances and orientations

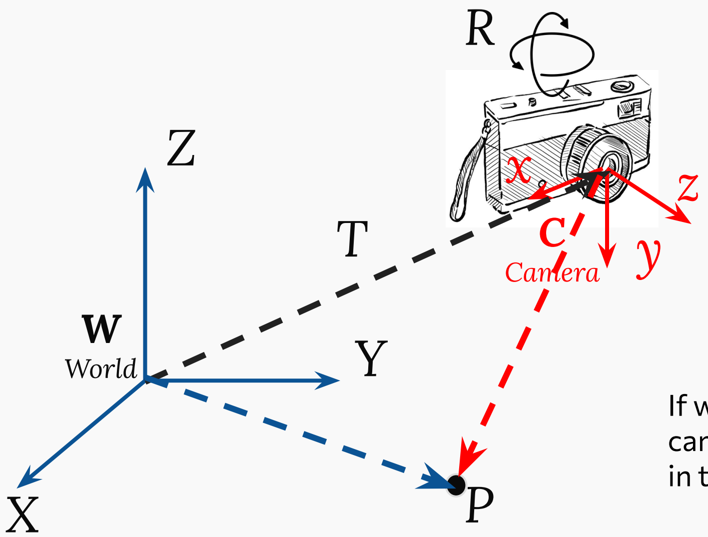
- Preserves both the inner product and cross product
- Preserves the triple product among three vectors, which corresponds to the volume of the parallelepiped spanned by the three vectors,
- Includes Translation and Rotation.

$$\|g_*(v)\| = \|v\|, \ \forall v \in \mathbb{R}^3$$

$$g_*(u) \times g_*(v) = g_*(u \times v), \ \forall u, v \in \mathbb{R}^3$$

$$\langle g_*(u), g_*(v) \times g_*(w) \rangle = \langle u, v \times w \rangle$$

Rigid-body Motion



Consider a world frame *W* and a camera frame *C*:

- We can consider W as a rigid frame and C as a moving frame or vice-versa.
- Translation corresponds to the vector that connects the origin of the frame *W* to that of the frame *C*;
- Rotation is the relative rotation between frames *W* and *C* (it is the rotation applied to the camera in relation to the world frame)

If we know a point P represented in the camera frame (Pc), its coordinates Pw in the world frame will be:

$$P_w = R.P_c + T$$

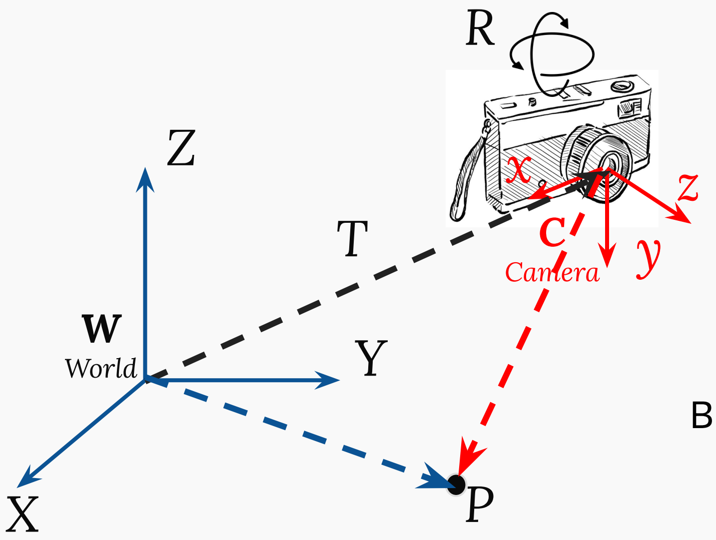
$$\widetilde{P}_w = T.R\,\widetilde{P}_c$$

in homogeneous coordinates

$$\widetilde{P}_w = \begin{bmatrix} R & T \\ \mathbf{0} & 1 \end{bmatrix} .\widetilde{P}_c$$

TO BE REMEMBERED: The translation and rotation applied to the camera to define its pose in the world frame also convert coordinates from the Camera Frame to the World Frame.

Rigid-body Motion



We can also convert from the world frame to the camera frame:

$$\tilde{P}_c = \begin{bmatrix} R & T \\ \mathbf{0} & 1 \end{bmatrix}^{-1} . \tilde{P}_w$$

But, attention:

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

Credits

R. Klette. Concise Computer Vision.
Springer-Verlag, London, 2014

Yi Ma, Stefano Soatto, Jana Kosecka e S. Shankar Sastry. An Invitation to 3D Vision: From Images to Geometric Models.
Springer, ISBN 0387008934