Computer Graphics

4 - Affine Space / Frame / Matrix

Yoonsang Lee
Hanyang University

Spring 2025

Outline

- Affine Space - Point vs. Vector
- Coordinate System & Reference Frame
- Affine Transformation Matrix
- Interpretation of Composite Transformations

Affine Space - Point vs. Vector

Affine Space - Point vs. Vector

- Conceptually, *points* and *vectors* are quite different.

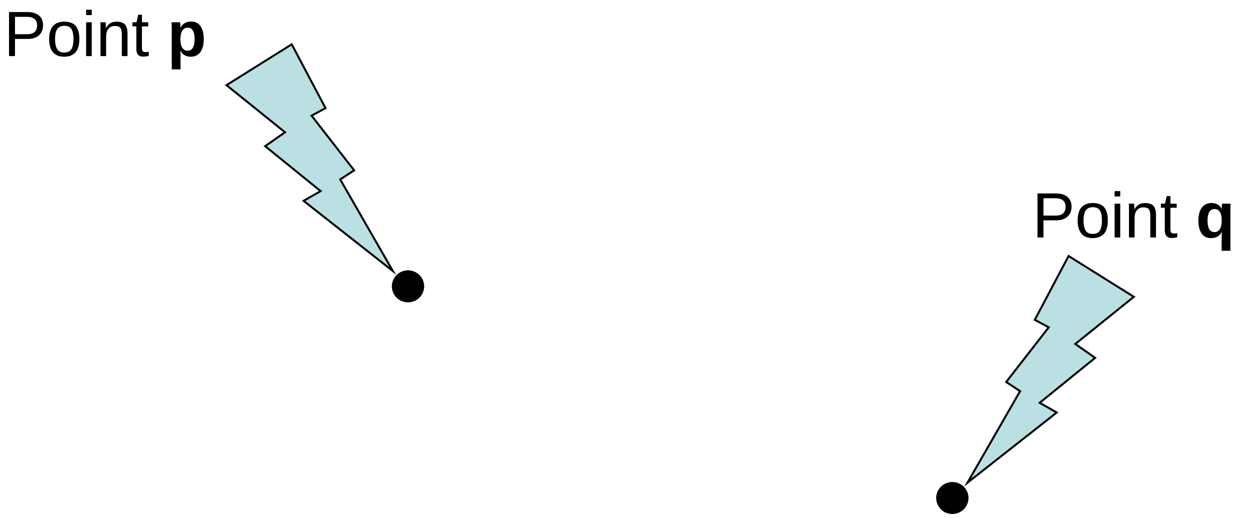
- Homogeneous coordinates can be used to express this difference.

- We will see affine space and the difference between points and vectors, and their relationship with the homogeneous coordinates.

- This concept has been called *coordinate invariant* or *coordinate-free* geometric programming.

  - Many of the following slides for this part are from: http://mrl.snu.ac.kr/courses/CourseGraphics/index_2017spring.html

Points

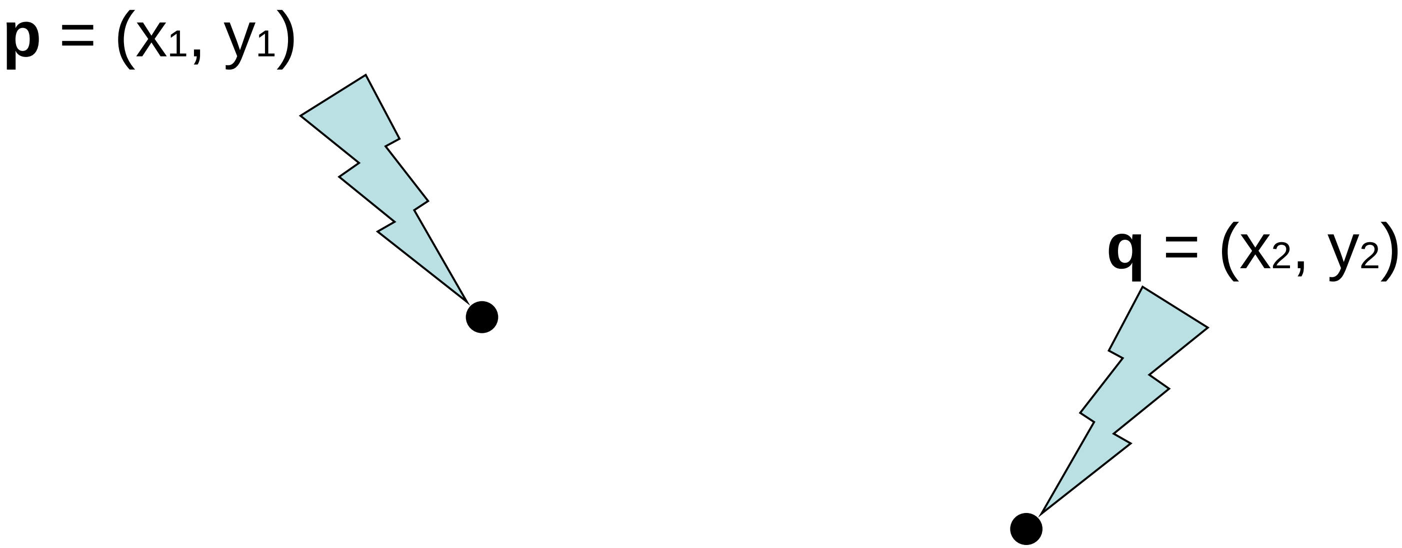


Point **p**

Point **q**

- What is the “sum” of these two "points" ?

If you assume coordinates, …

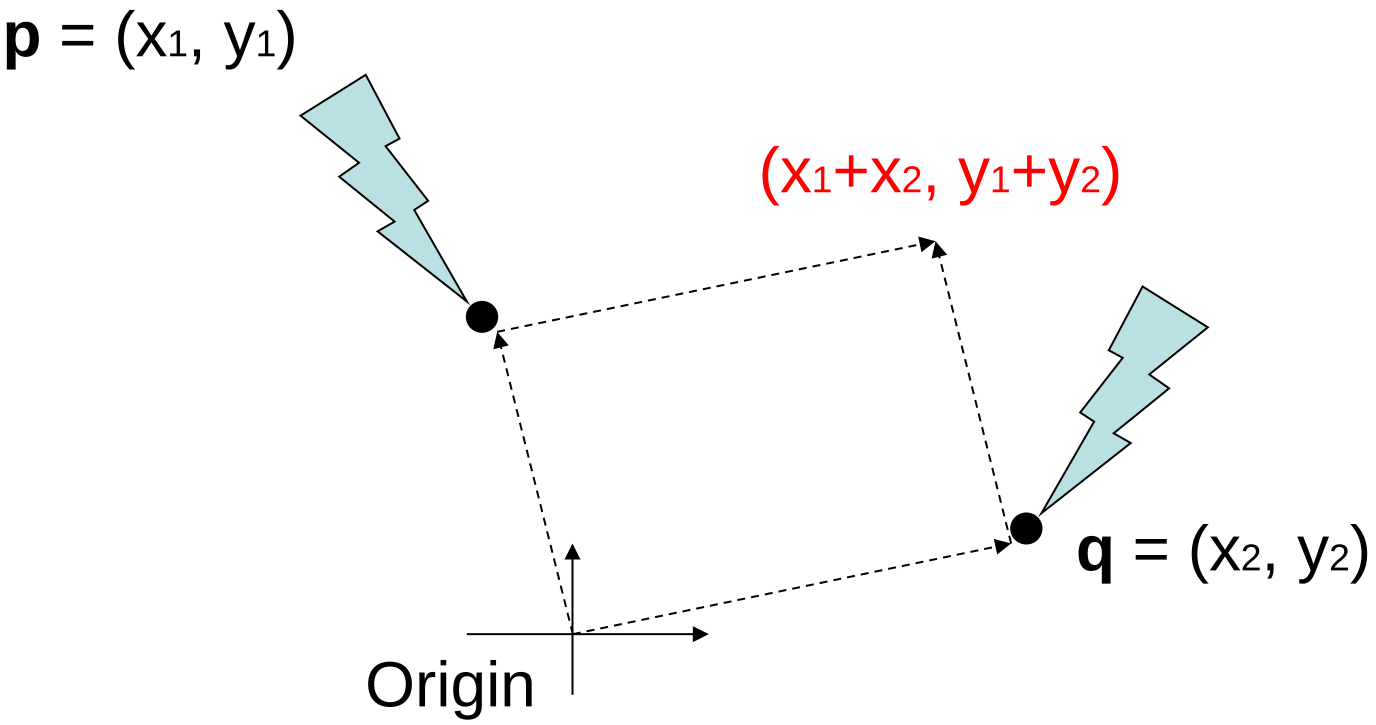


p = (x_1, y_1)

q = (x_2, y_2)

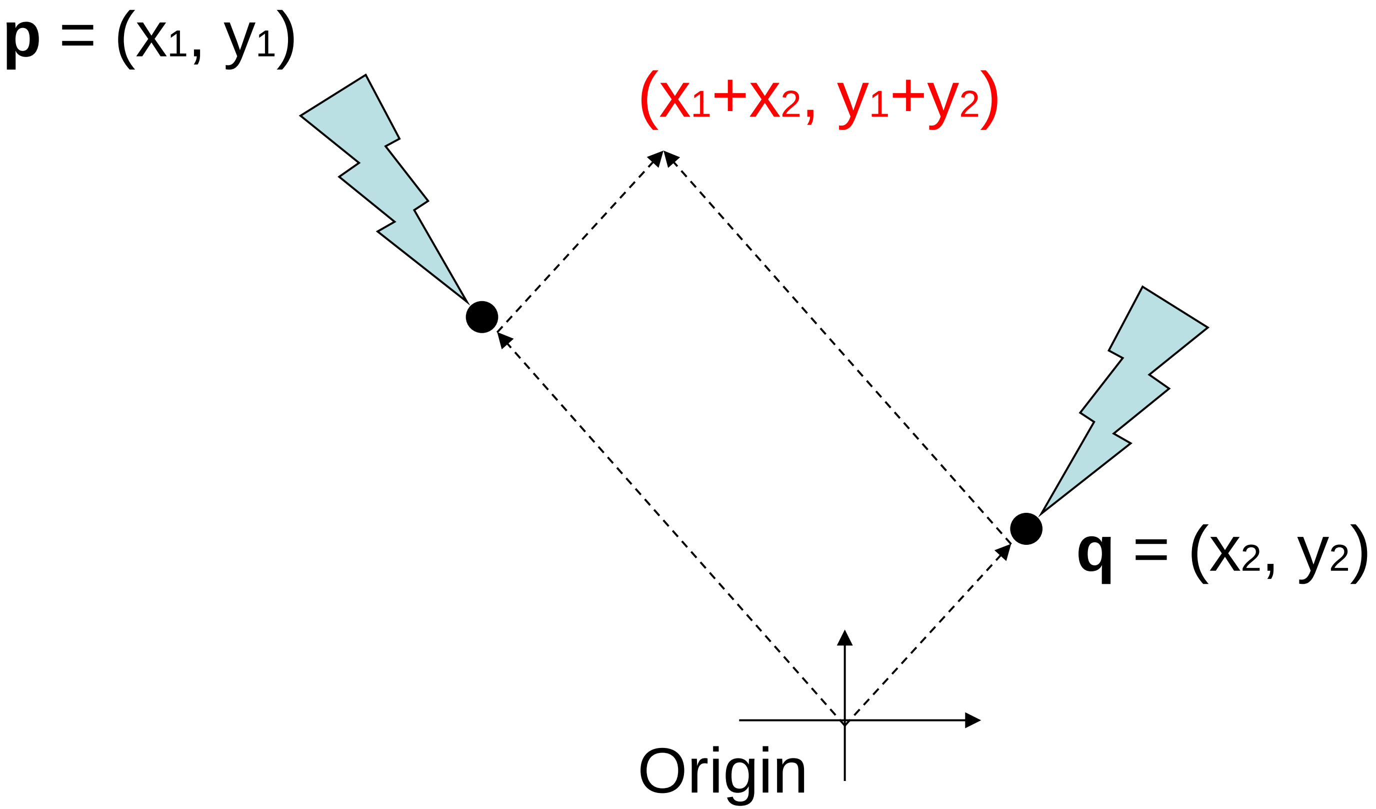
- The sum is (x_1+x_2, y_1+y_2)
  - – Is it correct ?
  - – Is it geometrically meaningful ?

If you assume coordinates, …



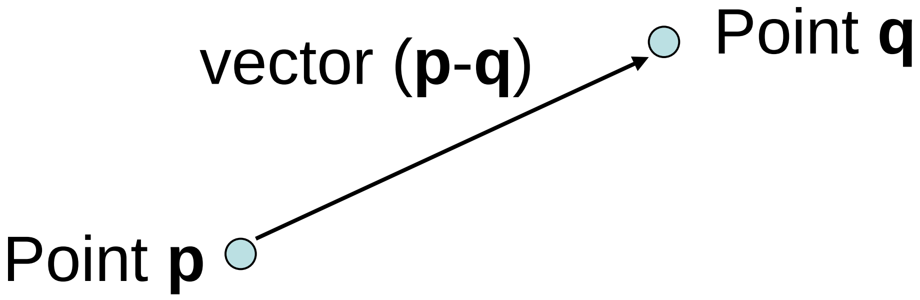
- Vector sum
  - (x_1, y_1) and (x_2, y_2) are considered as vectors from the origin to **p** and **q**, respectively.

If you select a different origin, …



- If you choose a different coordinate frame, you will get a different result

Points and Vectors



- A ***point*** is a position specified with coordinate values.
- A ***vector*** is specified as the difference between two points.
- If an ***origin*** is specified, then a **point** can be represented by a **vector from the origin.**
- But, a point is still not a vector in ***coordinate-free*** concepts.

Points & Vectors are Different!

- Mathematically (and physically),
- A *point* is a **location in space**.
- A *vector* is a **displacement in space**.

- An analogy with time:
- A *datetime* is a **location in time**.
- A *duration* is a **displacement in time**.

Vector and Affine Spaces

- ***Vector space***
  - Includes vectors and related operations
  - No points
- ***Affine space***
  - Superset of vector space
  - Includes vectors, points, and related operations

Vector spaces

- A ***vector space*** consists of
  - Set of vectors, together with
  - Two operations: addition of vectors and multiplication of vectors by scalar numbers
- A ***linear combination*** of vectors is also a vector

$$\mathbf{u}_0, \mathbf{u}_1, \cdots, \mathbf{u}_N \in V \quad \Rightarrow \quad c_0\mathbf{u}_0 + c_1\mathbf{u}_1 + \cdots + c_N\mathbf{u}_N \in V$$

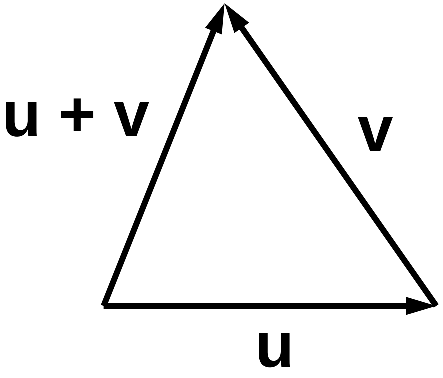
Affine Spaces

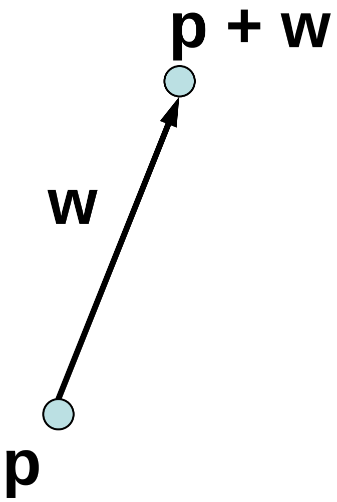
- An ***affine space*** consists of
  - Set of points, an associated vector space, and
  - Two operations: the difference between two points and the addition of a vector to a point

Coordinate-Free Geometric Operations

- Addition
- Subtraction
- Scalar multiplication

Addition



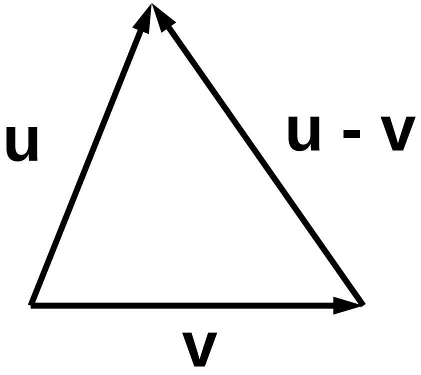


u + **v** is a vector

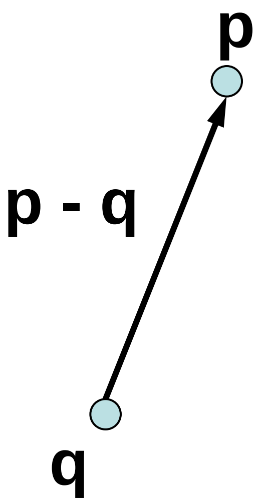
p + **w** is a point

u, v, w : vectors
p, q : points

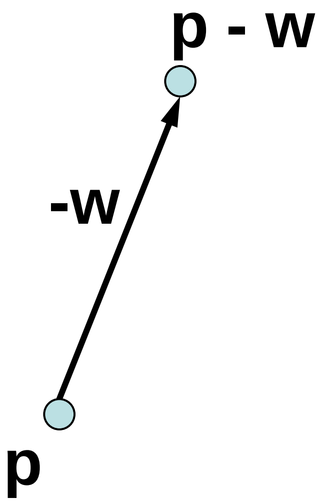
Subtraction



u - **v** is a vector



p - **q** is a vector



p - **w** is a point

u, v, w : vectors
p, q : points

Scalar Multiplication

scalar • vector = vector

1 • point = point

0 • point = vector

c • point = (undefined)   if (c≠0,1)

Affine Frame

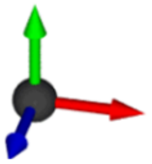
- A ***frame*** is defined as a set of vectors $\{\mathbf{v}_i \mid i=1, \ldots, N\}$ and a point **o**
  - Set of vectors $\{\mathbf{v}_i\}$ are bases of the associate vector space
  - **o** is an origin of the frame
  - *N* is the dimension of the affine space
  - Any point **p** can be written as

$$\mathbf{p} = \mathbf{o} + c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_N\mathbf{v}_N$$

  - Any vector **v** can be written as

$$\mathbf{v} = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_N\mathbf{v}_N$$

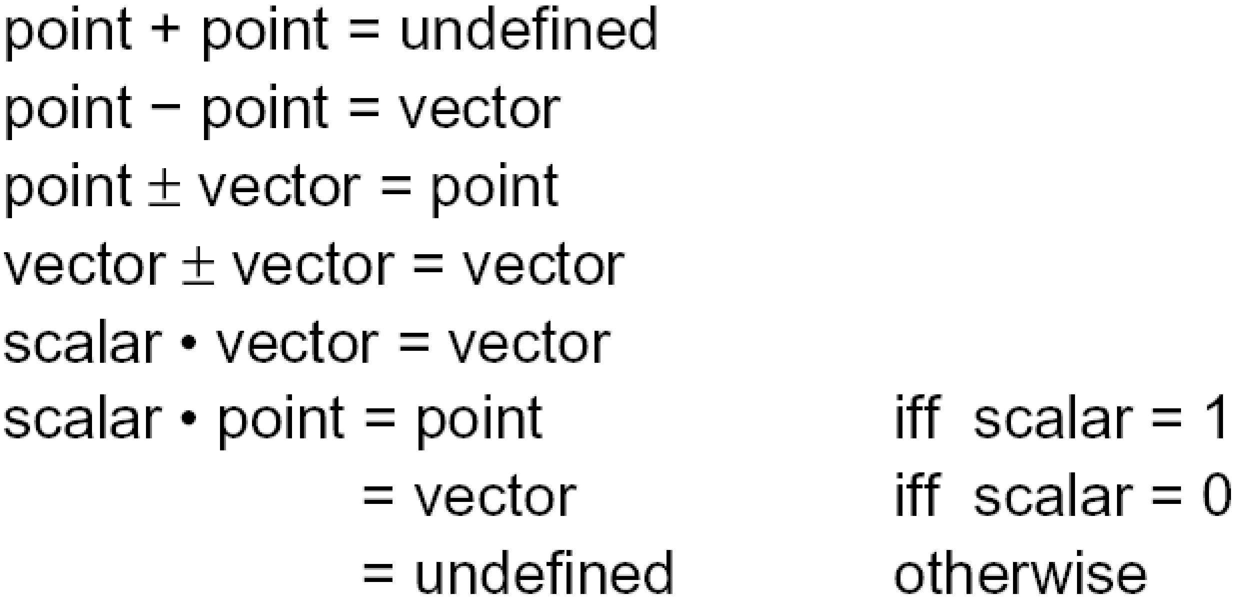
in 3D space



: Three vectors and a point

Summary

- In an affine space,



point + point = undefined

point − point = vector

point ± vector = point

vector ± vector = vector

scalar • vector = vector

scalar • point = point	iff scalar = 1
= vector	iff scalar = 0
= undefined	otherwise

Points & Vectors in Homogeneous Coordinates

- In homogeneous coordinates,
- A 3D **point** is represented: (x, y, z, **1**)
- A 3D **vector** is represented: (x, y, z, **0**)

- → This representation gives a completely consistent model with the concept of points and vectors in coordinate-free geometric programming!

Points & Vectors in Homogeneous Coordinates

$$(x_1, y_1, z_1, 1) + (x_2, y_2, z_2, 1) = (x_1+x_2, y_1+y_2, z_1+z_2, 2)$$

point ***point*** ***undefined***

$$(x_1, y_1, z_1, 1) - (x_2, y_2, z_2, 1) = (x_1-x_2, y_1-y_2, z_1-z_2, 0)$$

point ***point*** ***vector***

$$(x_1, y_1, z_1, 1) + (x_2, y_2, z_2, 0) = (x_1+x_2, y_1+y_2, z_1+z_2, 1)$$

point ***vector*** ***point***

$$(x_1, y_1, z_1, 0) + (x_2, y_2, z_2, 0) = (x_1+x_2, y_1+y_2, z_1+z_2, 0)$$

vector ***vector*** ***vector***

$$c * (x_1, y_1, z_1, 0) = (cx_1, cy_1, cz_1, 0)$$

vector ***vector***

$$c * (x_1, y_1, z_1, 1) = (cx_1, cy_1, cz_1, c)$$

point ***undefined*** ***(if $c \neq 0,1$)***

Points & Vectors in Homogeneous Coordinates

- Multiplying affine transformation matrix to a point and a vector:

$$\begin{bmatrix} M & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \boxed{1} \end{bmatrix} = \begin{bmatrix} M\mathbf{p} + \mathbf{t} \\ \boxed{1} \end{bmatrix} \qquad \begin{bmatrix} M & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \boxed{0} \end{bmatrix} = \begin{bmatrix} \boxed{M\mathbf{v}} \\ \boxed{0} \end{bmatrix}$$

point → point   vector → vector

- Note that translation is not applied to a vector!

Quiz 1

- Go to https://www.slido.com/
- Join #**cg-ys**
- Click "Polls"

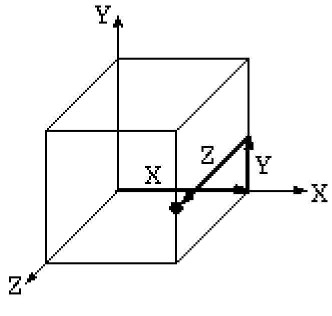
- Submit your answer in the following format:
  - **Student ID: Your answer**
  - **e.g. 2021123456: 4.0**

- Note that your quiz answer must be submitted **in the above format** to receive a quiz score!

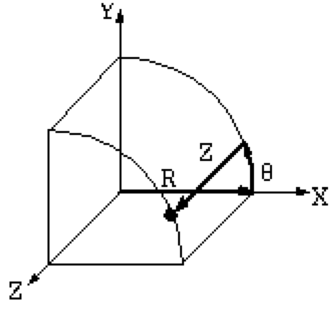
Coordinate System & Reference Frame

Coordinate System & Reference Frame

- Coordinate system
  - A system which uses one or more numbers, or coordinates, to uniquely determine the position of points.

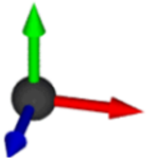


Cartesian (X,Y,Z components)
coordinate system 0 (C.S. 0)



Cylindrical (R,q,Z components)
coordinate system 1 (C.S. 1)

- Reference frame
  - Abstract coordinate system + physical reference points (to uniquely fix the coordinate system).



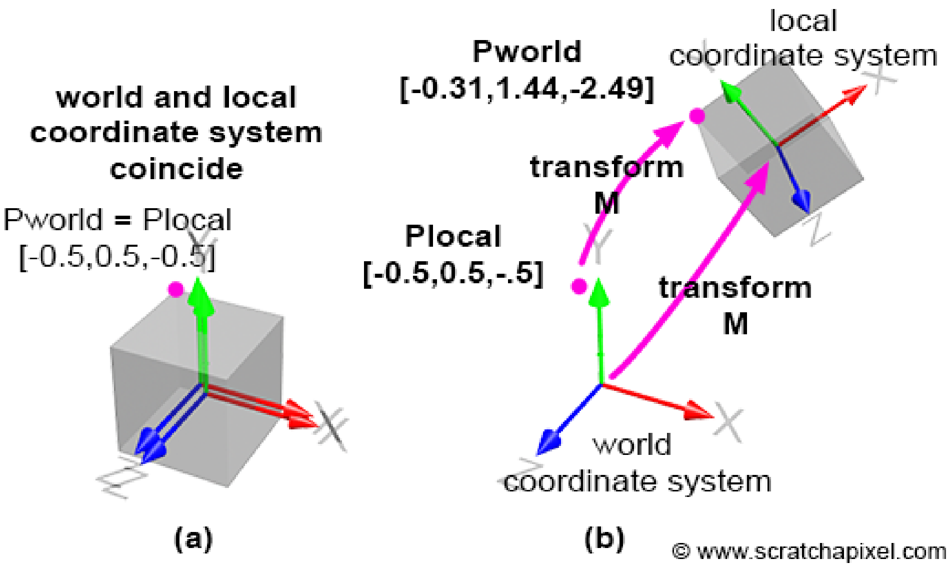
: Three vectors and a point

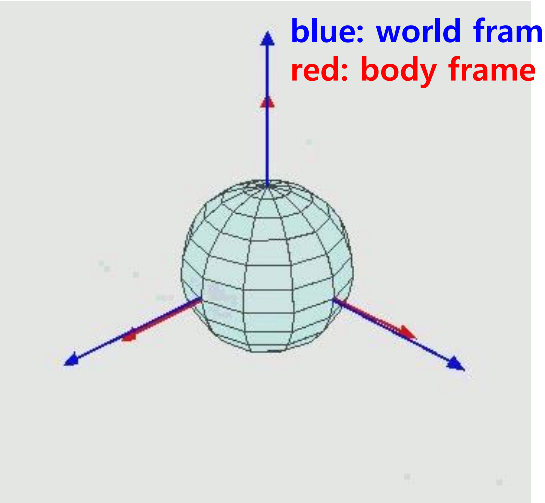
- These two terms are often mixed although they have slightly different meanings.

World / Body Frame (or Coordinate System)

- **World frame** (or coordinate system)
  - A frame (or coordinate system) attached to the **world.**
  - a.k.a. **global** frame, **fixed** frame

- **Body frame** (or coordinate system)
  - A frame (or coordinate system) attached to a **moving object.**
  - a.k.a. **local** frame





blue: world frame
red: body frame

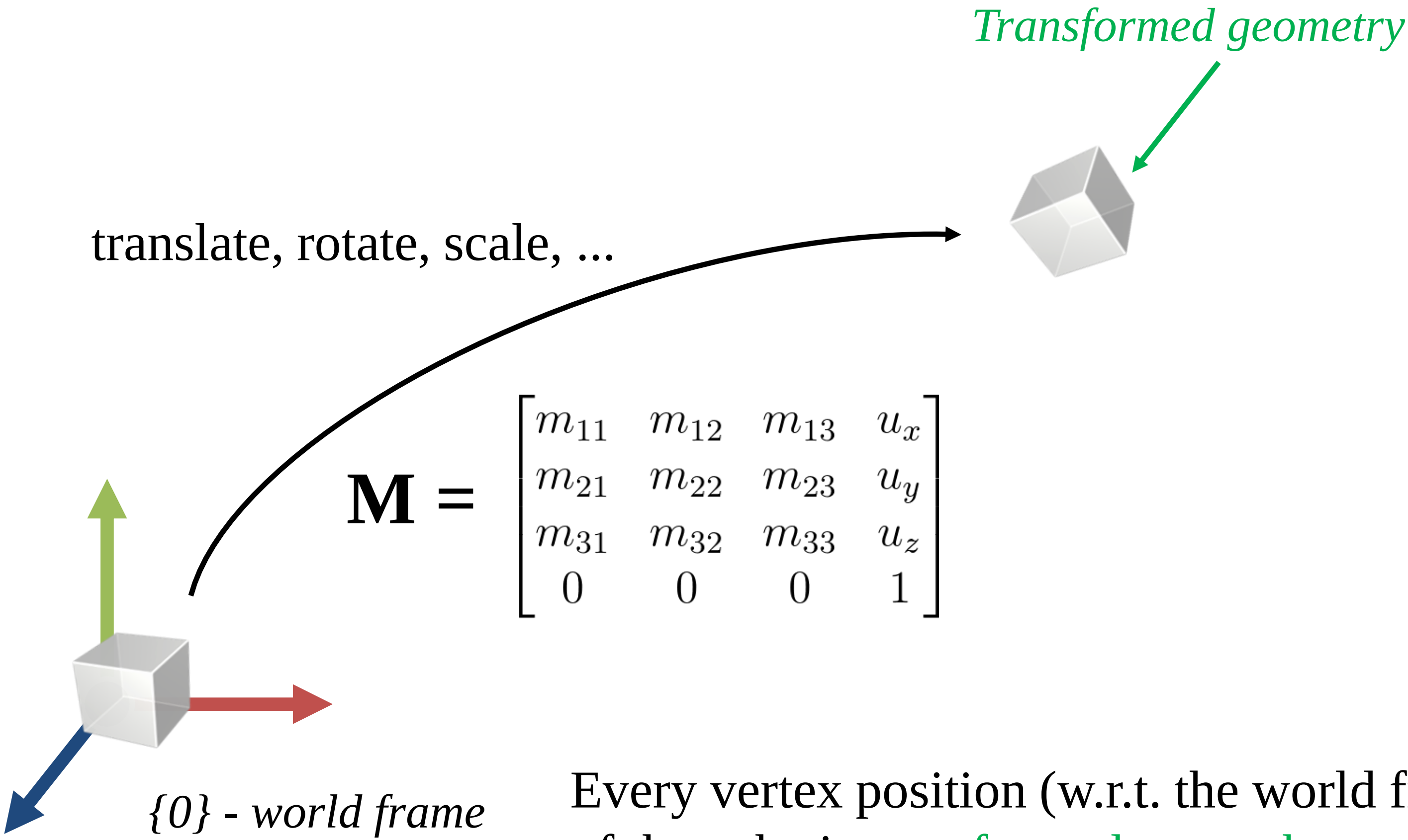
https://commons.wikimedia.org/wiki/File:Euler2a.gif

Affine Transformation Matrix

Meanings of Affine Transformation Matrix

- The meaning of the same affine transformation matrix can be described from several different perspectives.

1) Affine Transformation Matrix Transforms a Geometry w.r.t. World Frame

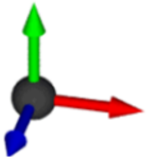


$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & u_x \\ m_{21} & m_{22} & m_{23} & u_y \\ m_{31} & m_{32} & m_{33} & u_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Every vertex position (w.r.t. the world frame) of the cube is transformed to another position (w.r.t. the world frame)

Review: Affine Frame

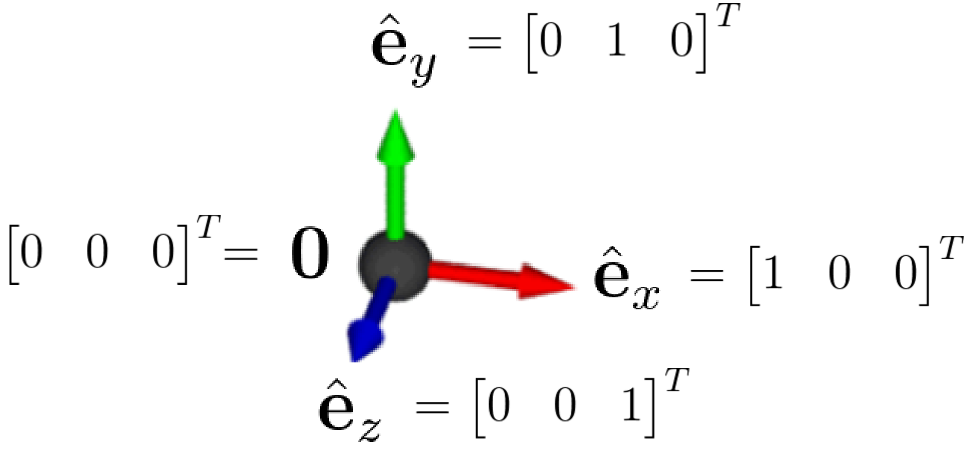
- An **affine frame** (for a 3D space) is defined by three vectors and one point
  - Three vectors for x, y, z axes
  - One point for origin



: Three vectors and a point

World Frame

- The **world frame** is usually represented by
  - Standard basis vectors for axes : $\hat{\mathbf{e}}_x, \hat{\mathbf{e}}_y, \hat{\mathbf{e}}_z$
  - Origin point : $\mathbf{0}$


$\hat{\mathbf{e}}_y = [0 \;\; 1 \;\; 0]^T$
$[0 \;\; 0 \;\; 0]^T = \mathbf{0}$
$\hat{\mathbf{e}}_x = [1 \;\; 0 \;\; 0]^T$
$\hat{\mathbf{e}}_z = [0 \;\; 0 \;\; 1]^T$

Let’s transform a "world frame"

- Apply M to this "world frame", that is,
  - Multiply M with the x, y, z axis *vectors* and the origin *point* of the world frame:

x axis *vector*

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & u_x \\ m_{21} & m_{22} & m_{23} & u_y \\ m_{31} & m_{32} & m_{33} & u_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ \boxed{0} \end{bmatrix} = \begin{bmatrix} m_{11} \\ m_{21} \\ m_{31} \\ \boxed{0} \end{bmatrix}$$

y axis *vector*

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & u_x \\ m_{21} & m_{22} & m_{23} & u_y \\ m_{31} & m_{32} & m_{33} & u_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ \boxed{0} \end{bmatrix} = \begin{bmatrix} m_{12} \\ m_{22} \\ m_{32} \\ \boxed{0} \end{bmatrix}$$

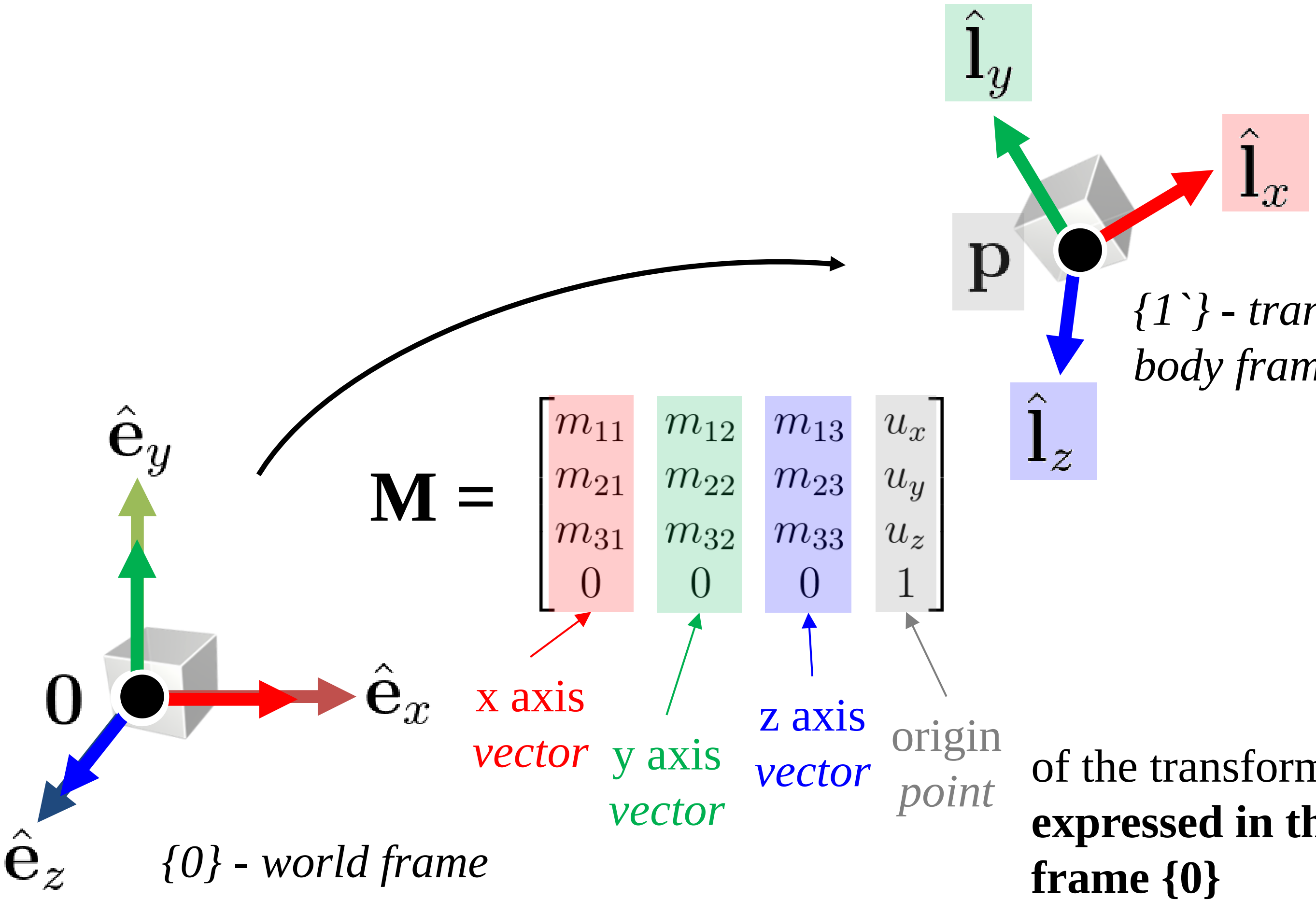
z axis *vector*

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & u_x \\ m_{21} & m_{22} & m_{23} & u_y \\ m_{31} & m_{32} & m_{33} & u_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ \boxed{0} \end{bmatrix} = \begin{bmatrix} m_{13} \\ m_{23} \\ m_{33} \\ \boxed{0} \end{bmatrix}$$

origin *point*

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & u_x \\ m_{21} & m_{22} & m_{23} & u_y \\ m_{31} & m_{32} & m_{33} & u_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \boxed{1} \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \\ u_z \\ \boxed{1} \end{bmatrix}$$

2) Affine Transformation Matrix **Defines an Affine Frame** w.r.t. World Frame



{1`} - transformed body frame

$$\mathbf{M} = \begin{array}{c} \begin{bmatrix} m_{11} & m_{12} & m_{13} & u_x \\ m_{21} & m_{22} & m_{23} & u_y \\ m_{31} & m_{32} & m_{33} & u_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

x axis *vector*

y axis *vector*

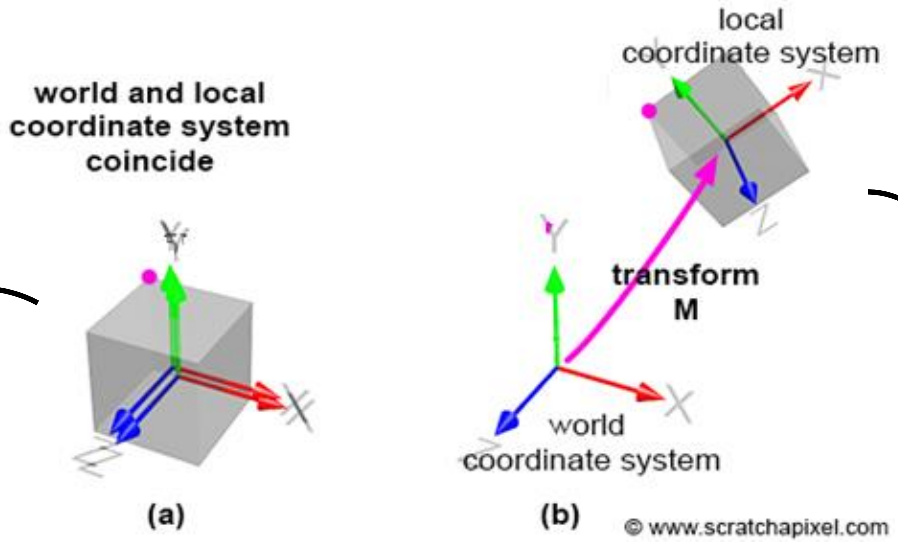
z axis *vector*

origin *point*

of the transformed frame {1'} **expressed in the world frame {0}**

{0} - world frame
{1} - initial body frame
{0} and {1} coincide

Examples



© www.scratchapixel.com

The object's body frame is defined by:

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

x axis *vector*

y axis *vector*

z axis *vector*

origin *point* *of the body frame* ***represented in the world frame***

The object's body frame is defined by:

origin *point*

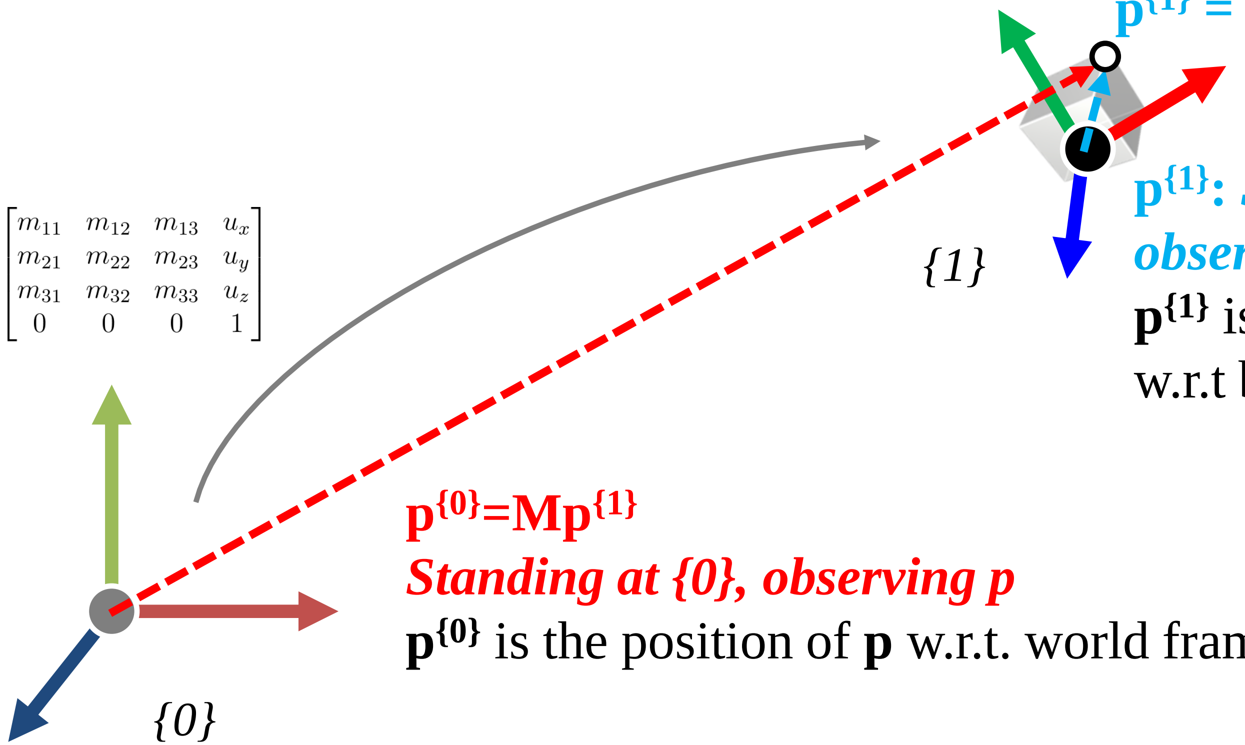
$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & u_1 \\ m_{21} & m_{22} & m_{23} & u_2 \\ m_{31} & m_{32} & m_{33} & u_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

x axis *vector*

y axis *vector*

z axis *vector*

3) Affine Transformation Matrix Transforms a Point Represented in an Affine Frame to (the same) Point (but) Represented in World Frame



$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & u_x \\ m_{21} & m_{22} & m_{23} & u_y \\ m_{31} & m_{32} & m_{33} & u_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\mathbf{p}^{\{1\}} = (1, 1, 0)$

$\mathbf{p}^{\{1\}}$: ***Standing at {1}, observing p***

$\mathbf{p}^{\{1\}}$ is the position of **p** w.r.t body frame {1}

$\mathbf{p}^{\{0\}}=\mathbf{M}\mathbf{p}^{\{1\}}$

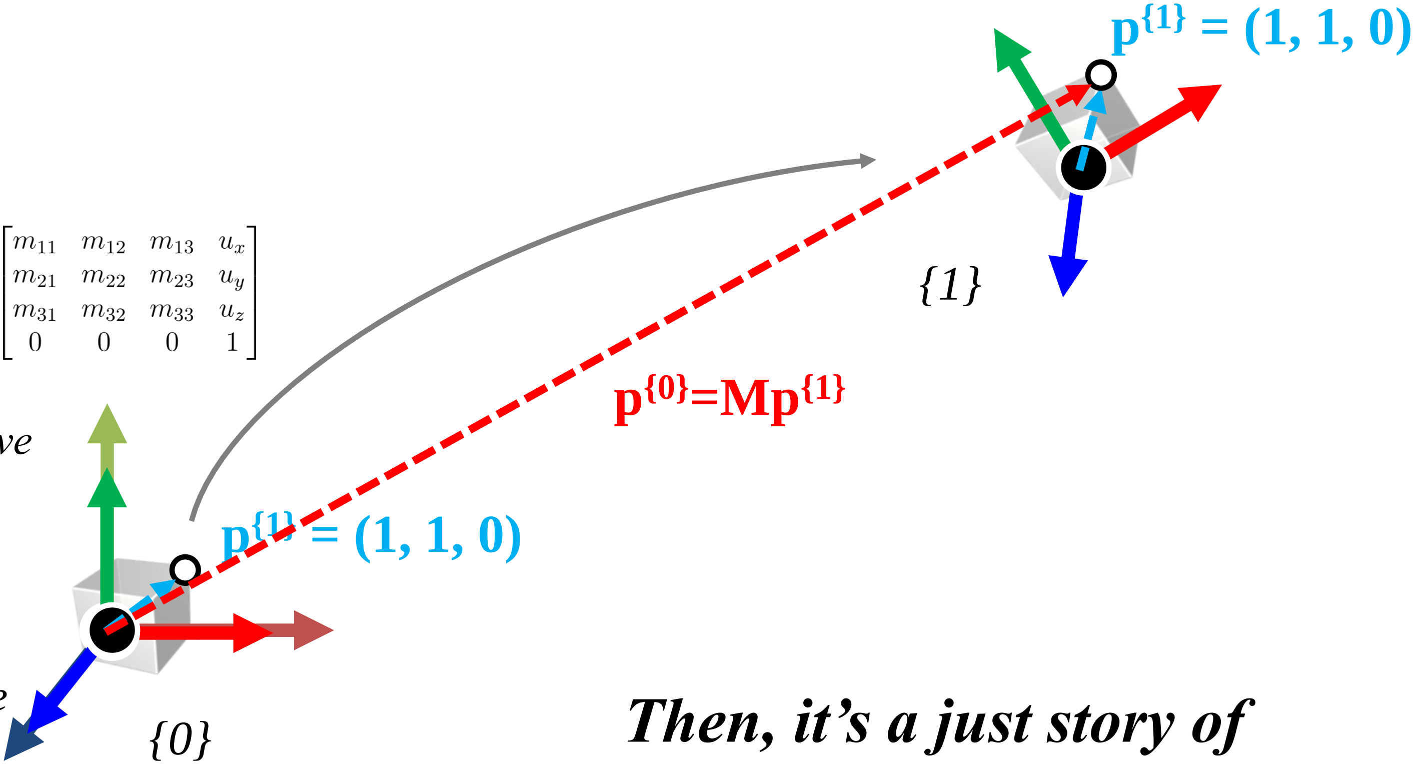
Standing at {0}, observing p

$\mathbf{p}^{\{0\}}$ is the position of **p** w.r.t. world frame {0}

3) Affine Transformation Matrix Transforms a Point Represented in an Affine Frame to (the same) Point (but) Represented in World Frame Because...

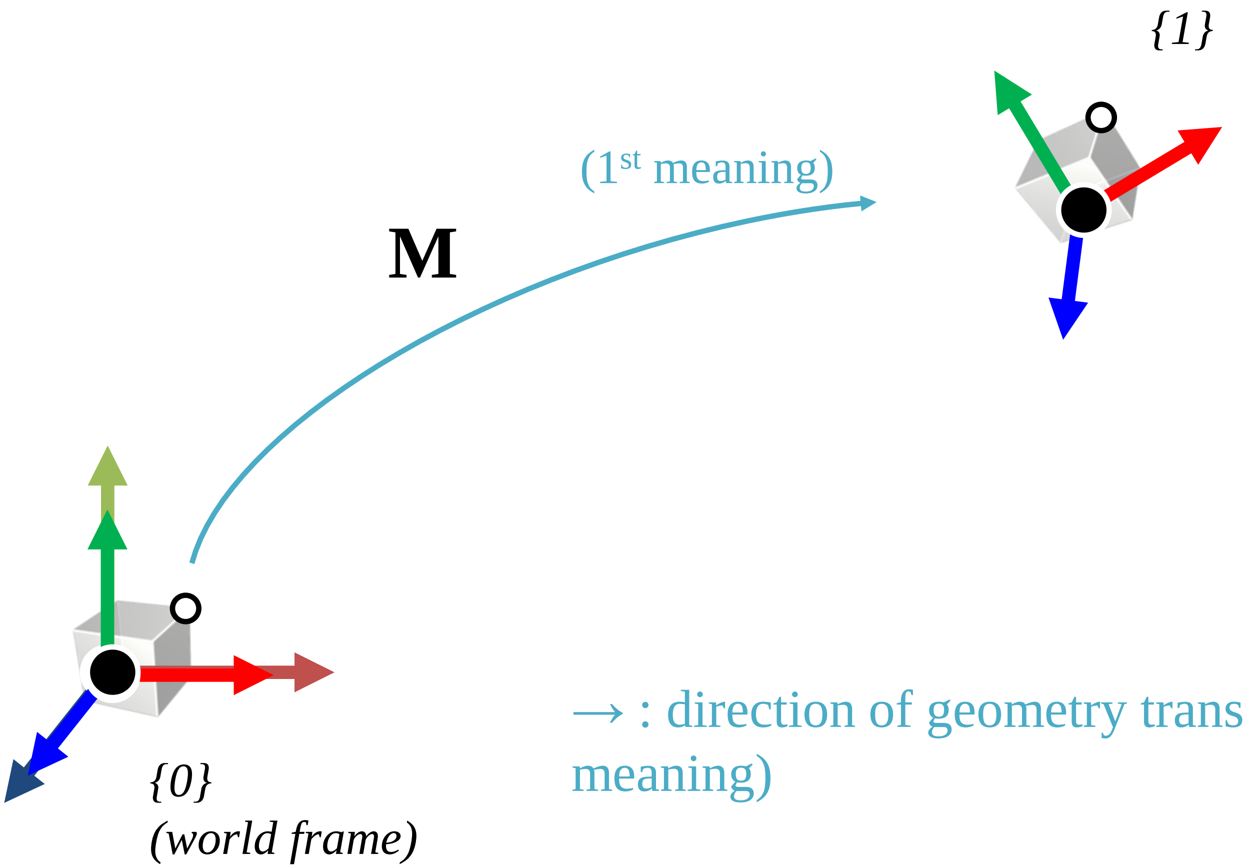
$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & u_x \\ m_{21} & m_{22} & m_{23} & u_y \\ m_{31} & m_{32} & m_{33} & u_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Let's say we have the same cube object whose body frame coincides with the world frame



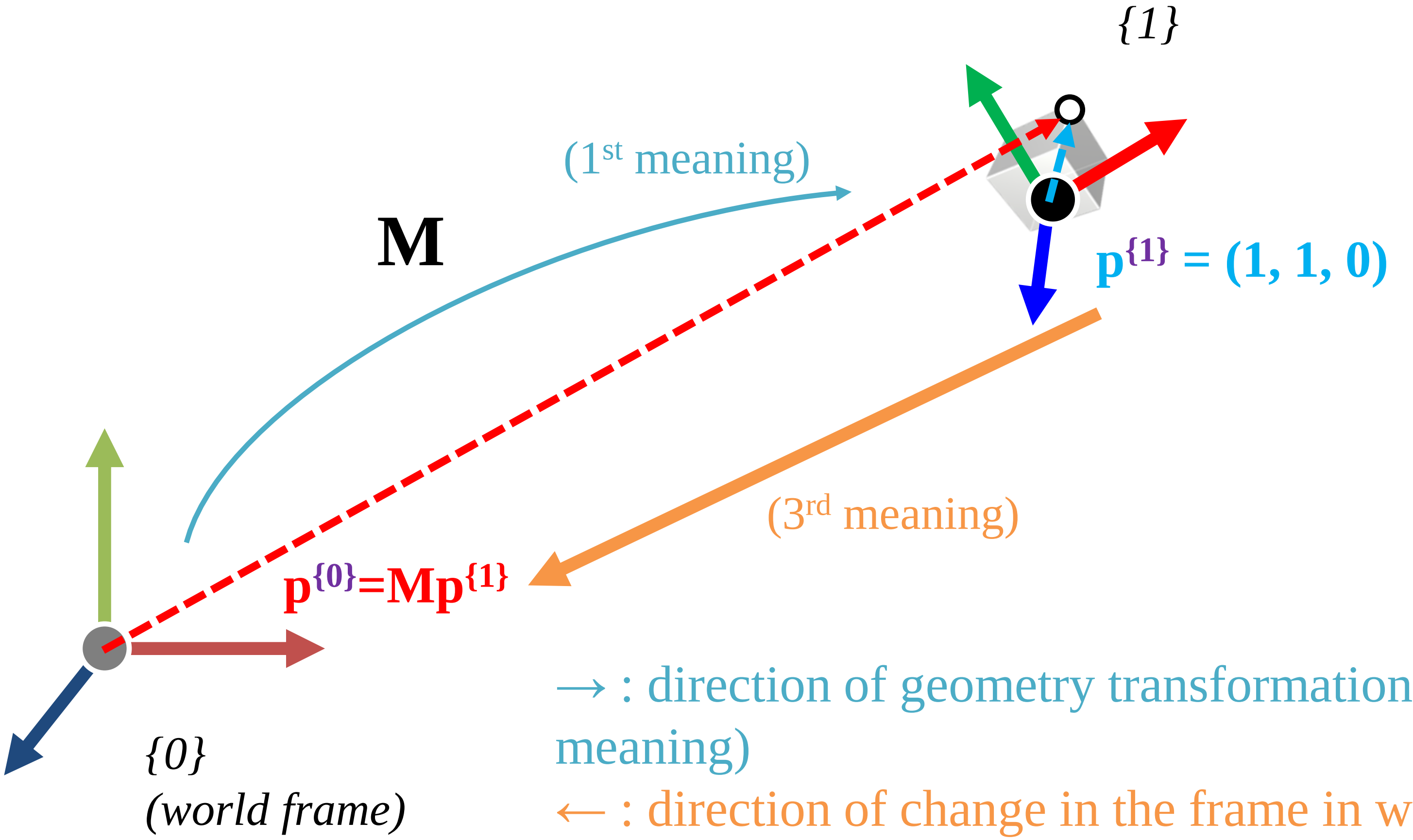
Then, it's a just story of transforming a geometry!

Directions of the "arrow"



→ : direction of geometry transformation (1st meaning)

Directions of the "arrow"



→ : direction of geometry transformation (1st meaning)

← : direction of change in the frame in which the point is described (3rd meaning) - {1}→{0}

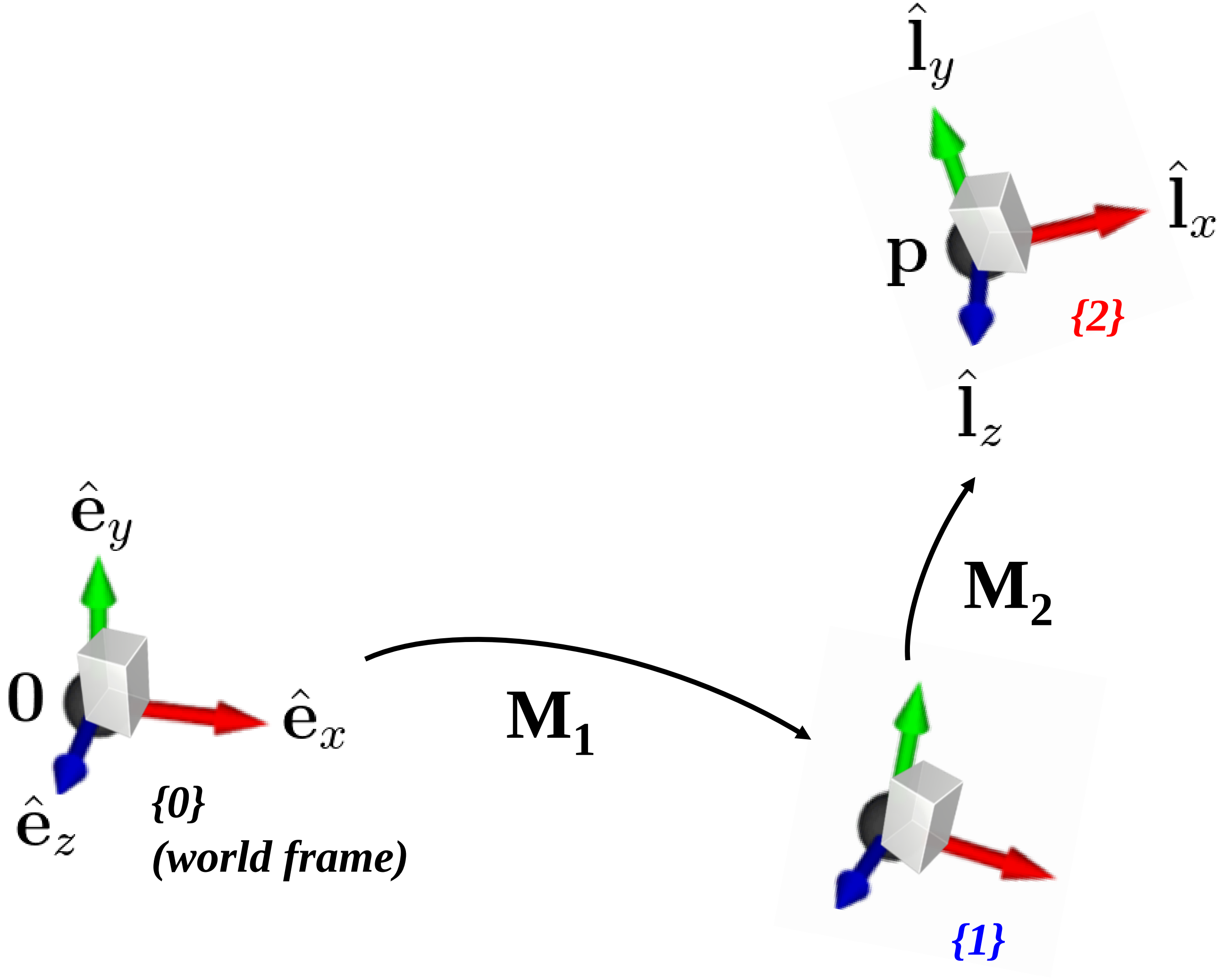
Quiz 2

- Go to https://www.slido.com/
- Join #**cg-ys**
- Click "Polls"

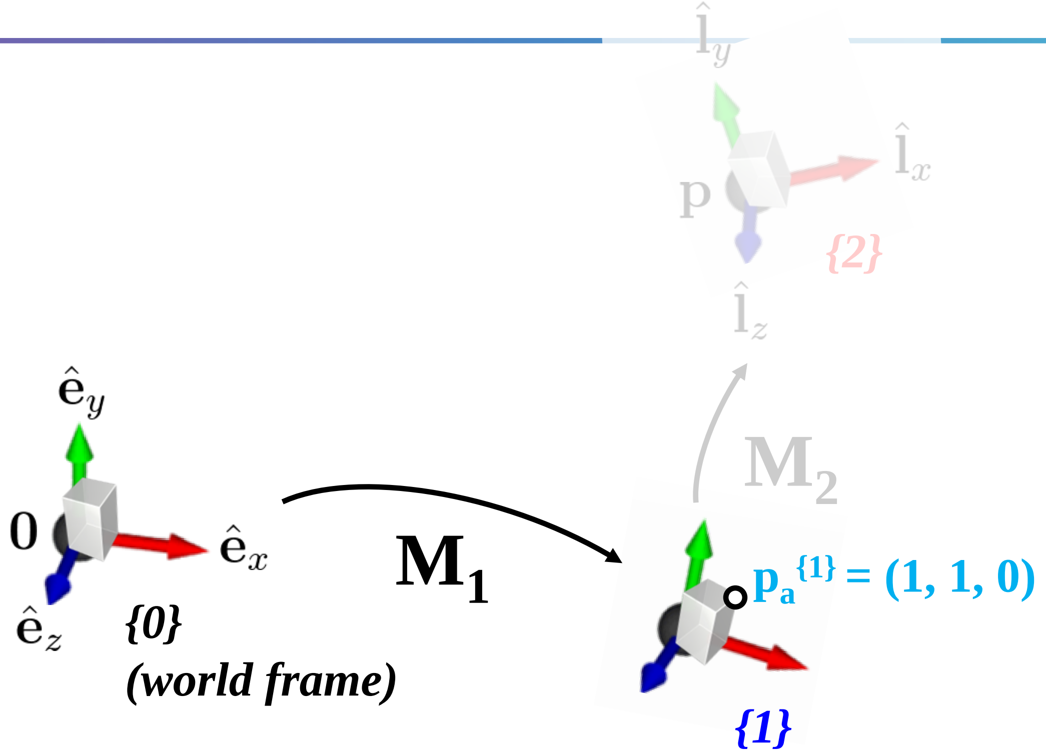
- Submit your answer in the following format:
  - **Student ID: Your answer**
  - **e.g. 2021123456: 4.0**

- Note that your quiz answer must be submitted **in the above format** to receive a quiz score!

All these concepts works even if the starting frame is not world frame!

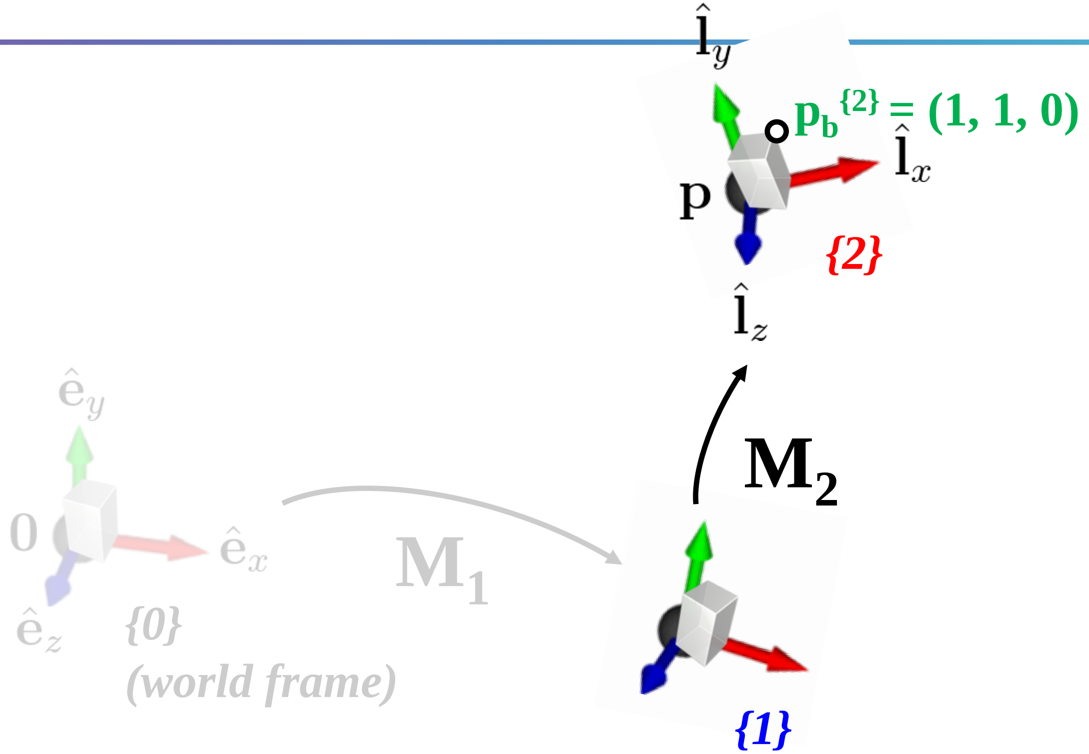


{0} to {1}



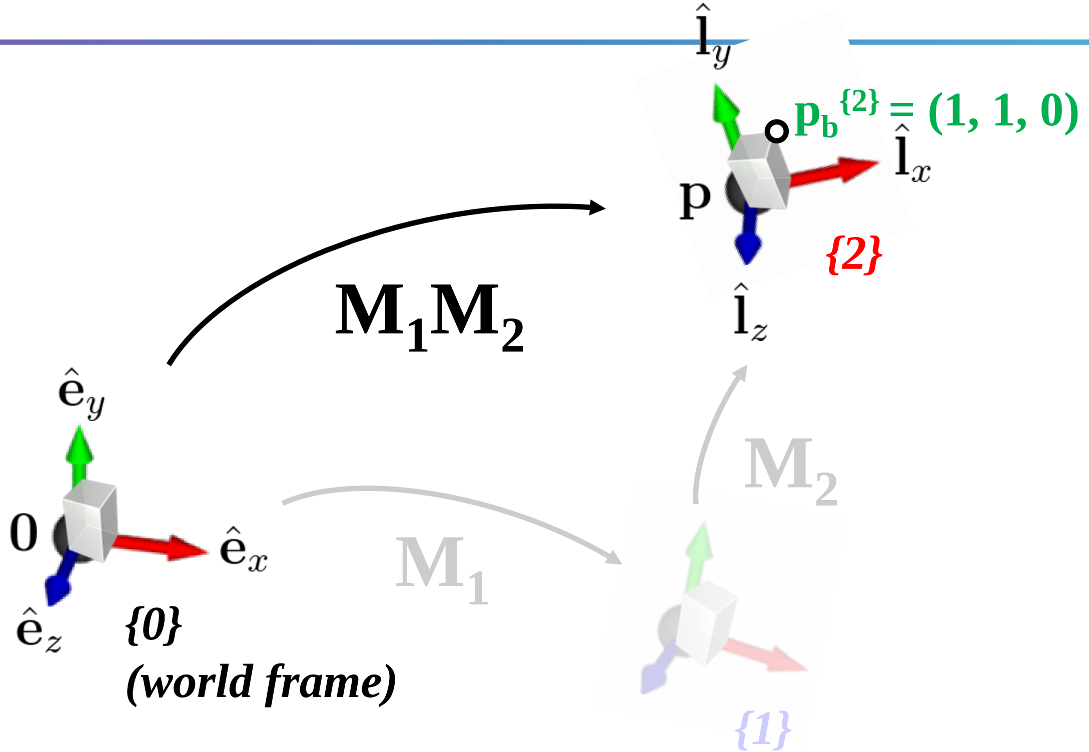
- 1) $\mathbf{M}_1$ transforms w.r.t. ***{0}*** a geometry represented in ***{0}***
- 2) $\mathbf{M}_1$ defines an ***{1}*** w.r.t. ***{0}***
- 3) $\mathbf{M}_1$ transforms a point represented in ***{1}*** to the same point but represented in ***{0}***
  - $\mathbf{p}_a^{\{0\}} = \mathbf{M}_1 \mathbf{p}_a^{\{1\}}$

{1} to {2}



- 1) $\mathbf{M_2}$ transforms w.r.t. ***{1}*** a geometry represented in ***{1}***
- 2) $\mathbf{M_2}$ defines an ***{2}*** w.r.t. ***{1}***
- 3) $\mathbf{M_2}$ transforms a point represented in ***{2}*** to the same point but represented in ***{1}***
  - $\mathbf{p_b^{\{1\}}=M_2p_b^{\{2\}}}$

{0} to {2}



- 1) $\mathbf{M_1M_2}$ transforms w.r.t. ***{0}*** a geometry represented in ***{0}***
- 2) $\mathbf{M_1M_2}$ defines an ***{2}*** w.r.t. ***{0}***
- 3) $\mathbf{M_1M_2}$ transforms a point represented in ***{2}*** to the same point but represented in ***{0}***
  - $\mathbf{p_b}^{\{1\}}=\mathbf{M_2p_b}^{\{2\}}$, $\mathbf{p_b}^{\{0\}}=\mathbf{M_1p_b}^{\{1\}}=\mathbf{M_1M_2p_b}^{\{2\}}$

Interpretation of Composite Transformations

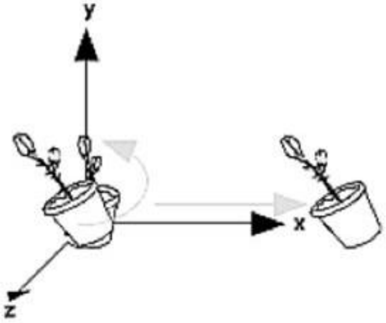
Revisit: Order Matters!

- If T and R are matrices representing affine transformations,

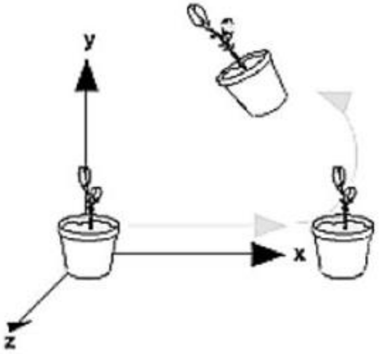
- **p'** = TR**p**
  - First apply transformation R to point **p**, then apply transformation T to transformed point R**p**

- **p'** = RT**p**
  - First apply transformation T to point **p**, then apply transformation R to transformed point T**p**

- ***w.r.t. world frame!***



Rotate then Translate



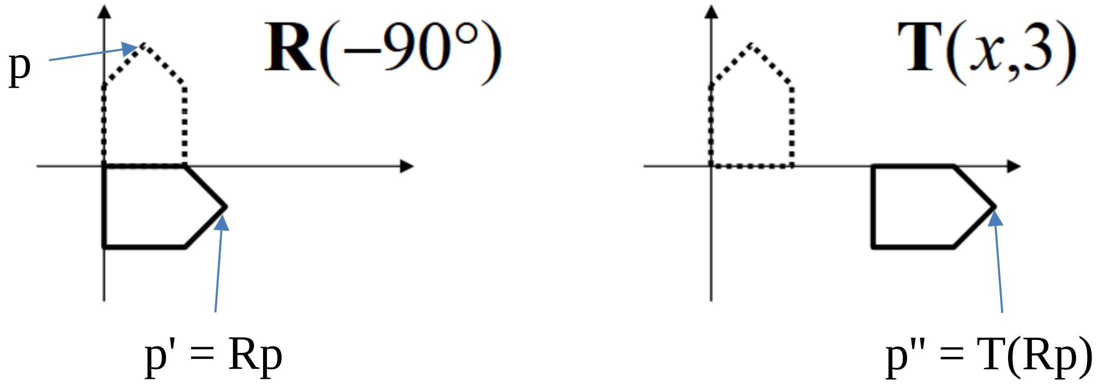
Translate then Rotate

Interpretation of Composite Transformations #1

- An example transformation:

$$M = \mathbf{T}(x,3) \cdot \mathbf{R}(-90°)$$

- This is how we've interpreted so far:
  - R-to-L: Transforms ***w.r.t. world frame***

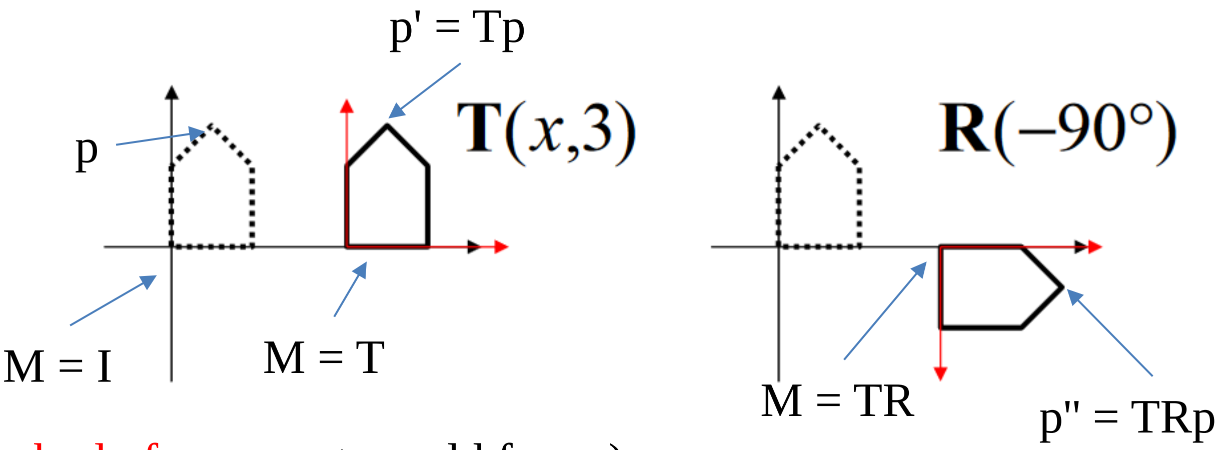


Interpretation of Composite Transformations #2

- An example transformation:

$$M = \mathbf{T}(x,3)\cdot\mathbf{R}(-90^\circ)$$

- **Another way of interpretation:**
  - L-to-R: Transforms ***w.r.t. body frame***



(M defines the body frame w.r.t. world frame)

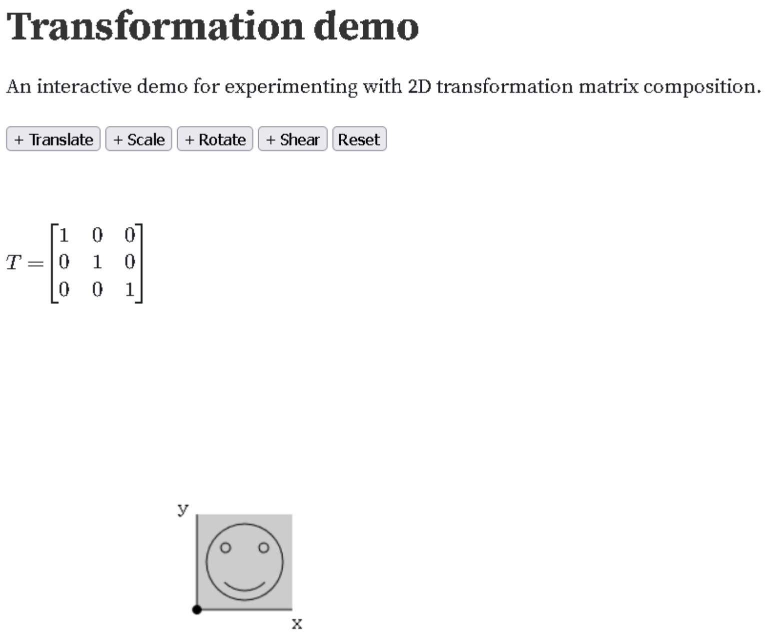
Pre-(left) & Post-(right) Multiplication

- **p' =** $\mathbf{M_1 M_2 p}$ **(pre-multiplication** by $\mathbf{M_1}$**)**
  - (R-to-L)
  - 1) Apply $\mathbf{M_2}$ w.r.t. world frame to a point **p** represented in world frame.
  - 2) Apply $\mathbf{M_1}$ w.r.t. world frame to a point $\mathbf{M_2 p}$ represented in world frame.

- **p' =** $\mathbf{M_1 M_2 p}$ **(post-multiplication** by $\mathbf{M_2}$**)**
  - (L-to-R)
  - 1) Apply $\mathbf{M_1}$ w.r.t. initial body frame **I** (world frame) to a point **p** represented in body frame **I** (world frame). Body frame is updated to $\mathbf{M_1}$.
  - 2) Apply $\mathbf{M_2}$ w.r.t. body frame $\mathbf{M_1}$ to a point **p** represented in body frame $\mathbf{M_1}$. Body frame is updated to $\mathbf{M_1 M_2}$.

  - *Another useful interpretation:*
  - 1') Apply $\mathbf{M_1}$ w.r.t. body frame **I** (world frame) to update body frame to $\mathbf{M_1}$
  - 2') Apply $\mathbf{M_2}$ w.r.t. body frame $\mathbf{M_1}$ to update body frame to $\mathbf{M_1 M_2}$
  - 3') Locate **p** in body frame $\mathbf{M_1 M_2}$

[Demo] L-to-R & R-to-L Interpretation



Transformation demo

An interactive demo for experimenting with 2D transformation matrix composition.

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

https://observablehq.com/@esperanc/transformation-demo

- Add translation and linear transforms in various orders with '+' buttons.
- Drag the slider to see the matrix value change and the shape transform.
- Interpret the meaning of composite transformations in L-to-R and R-to-L order.

Lab Session

- Now, let's start the lab today.