Transformations

Lecture 4-2

Yoonsang Lee and Taesoo Kwon
Spring 2019

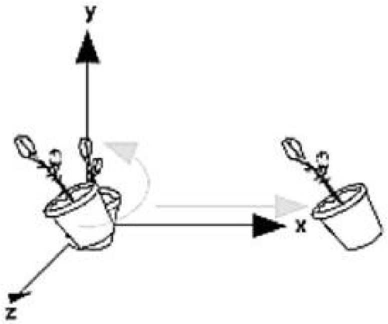
Topics Covered

- Reference Frame & Composite Transformations
  - Coordinate System & Reference Frame
  - Global & Local Coordinate System
  - Interpretation of Composite Transformations
- Affine Geometry: Vectors & Points
- Affine Basis

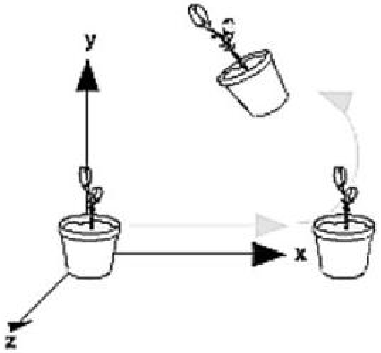
Reference Frame & Composite Transformations

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**
- Note that these transformations are done **w.r.t. global coordinate system**



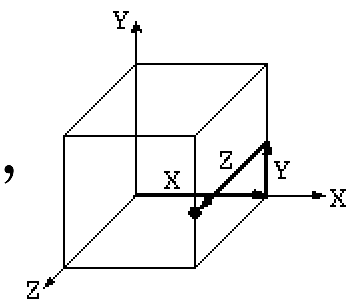
Rotate then Translate



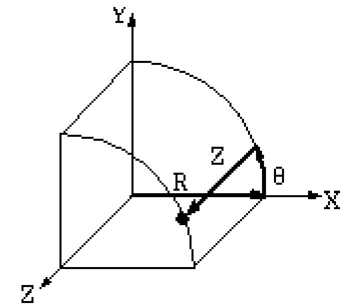
Translate then Rotate

Coordinate System & Reference Frame

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

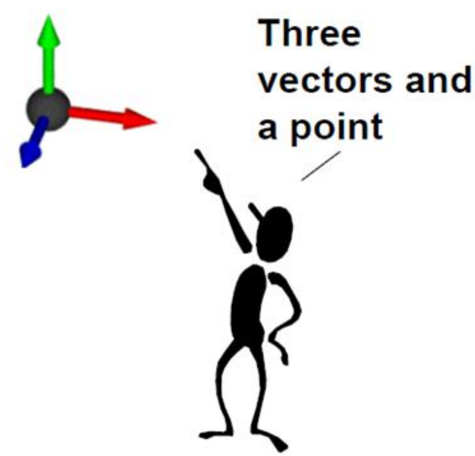


Cartesian (X,Y,Z components) coordinate system 0 (C.S. 0)



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).



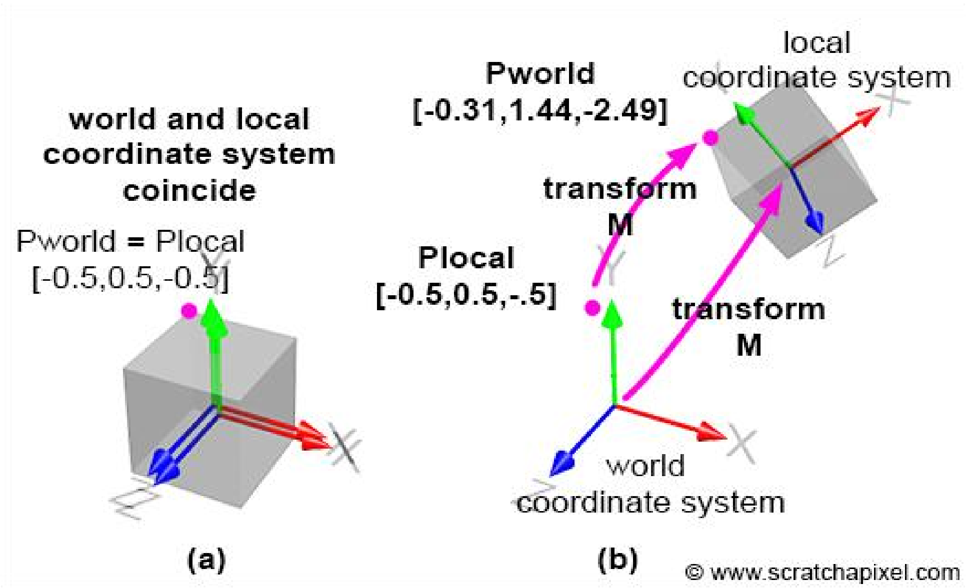
Coordinate System & Reference Frame

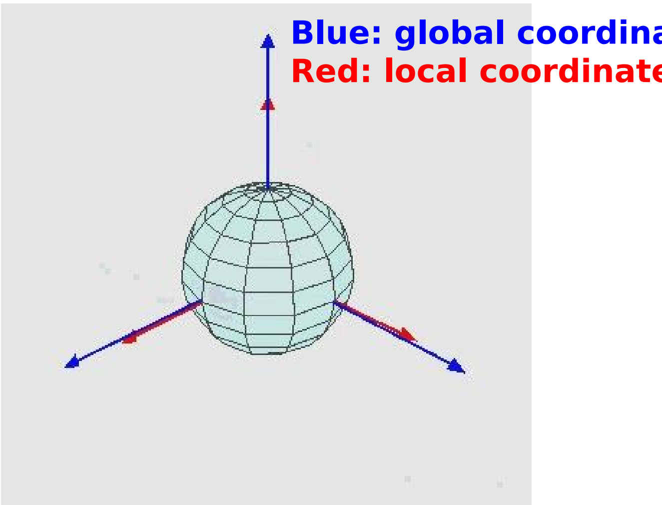
- Two terms are slightly different:
  - **Coordinate system** is a mathematical concept, about a choice of “language” used to describe observations.
  - **Reference frame** is a physical concept related to state of motion.
  - You can think the coordinate system determines the way one describes/observes the motion in each reference frame.

- But these two terms are often mixed.

Global & Local Coordinate System(or Frame)

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





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

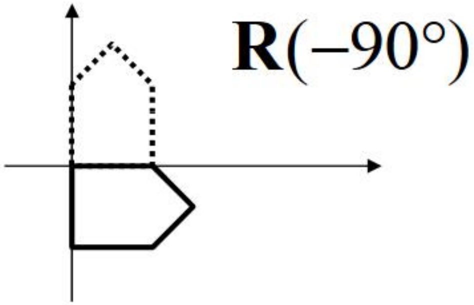
Interpretation of Composite Transformations #1

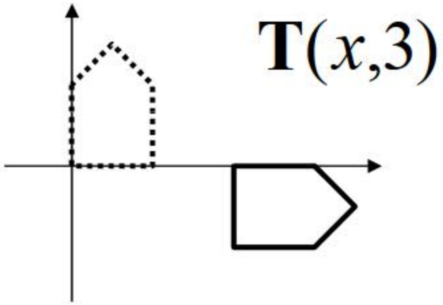
- An example transformation:

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

- how we've interpreted so far:
  - R-to-L : interpret operations w.r.t. fixed coordinates

=global coordinate
=world coordinate





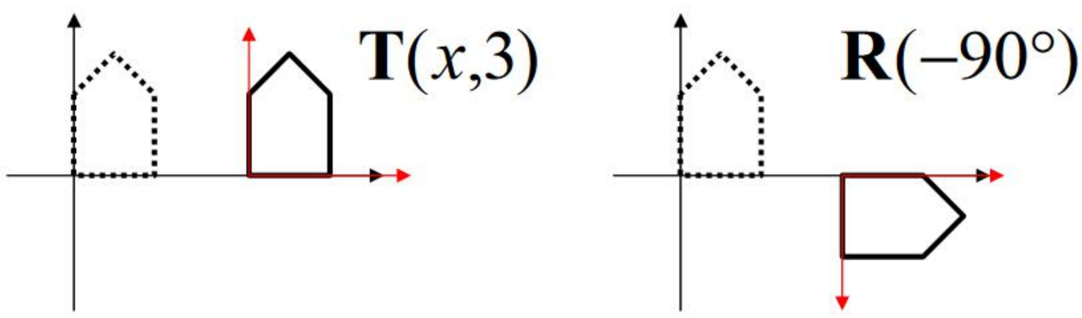
Interpretation of Composite Transformations #2

- An example transformation:

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

- **Another way of interpretation:**

  – L-to-R : interpret operations w.r.t local coordinates



Left & Right Multiplication

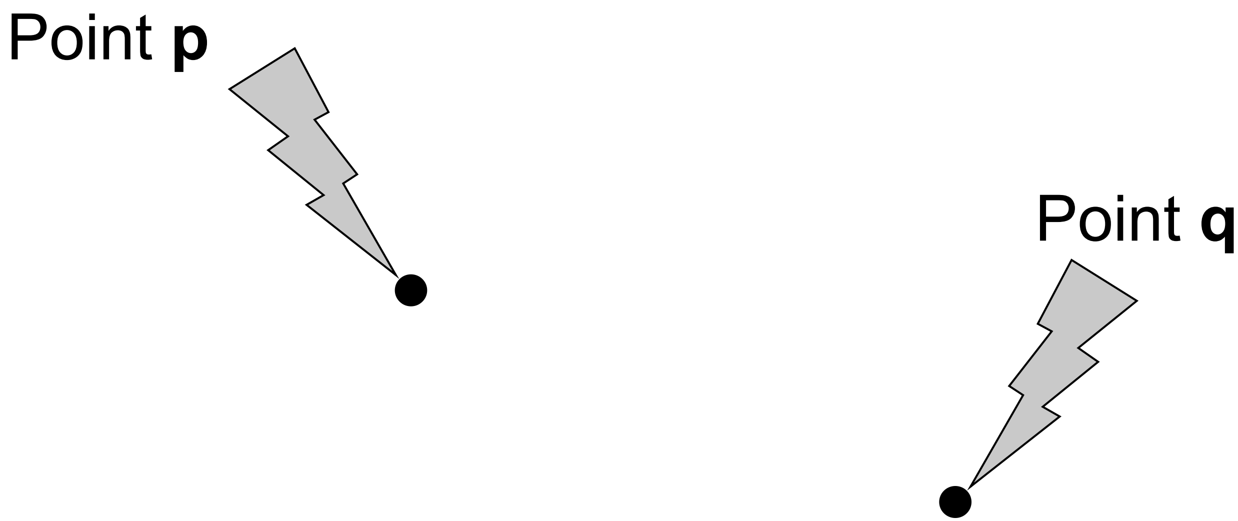
- In summary, after some practice, we will learn that:

- p' = **R**Tp (left-multiplication by **R**)
  - Apply transformation **R** to point Tp w.r.t. global coordinates

- p' = T**R**p (right-multiplication by **R**)
  - Apply transformation **R** to point Tp w.r.t. local coordinates
  - Also moves the local frame itself

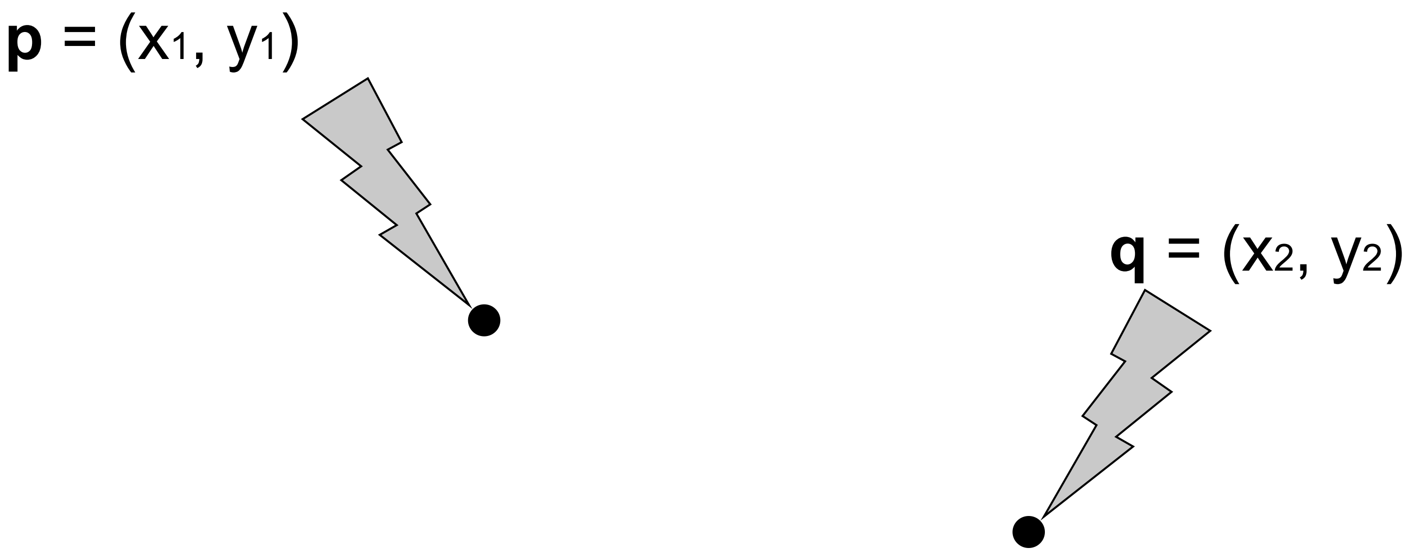
Affine Geometry: Vectors & Points

Points

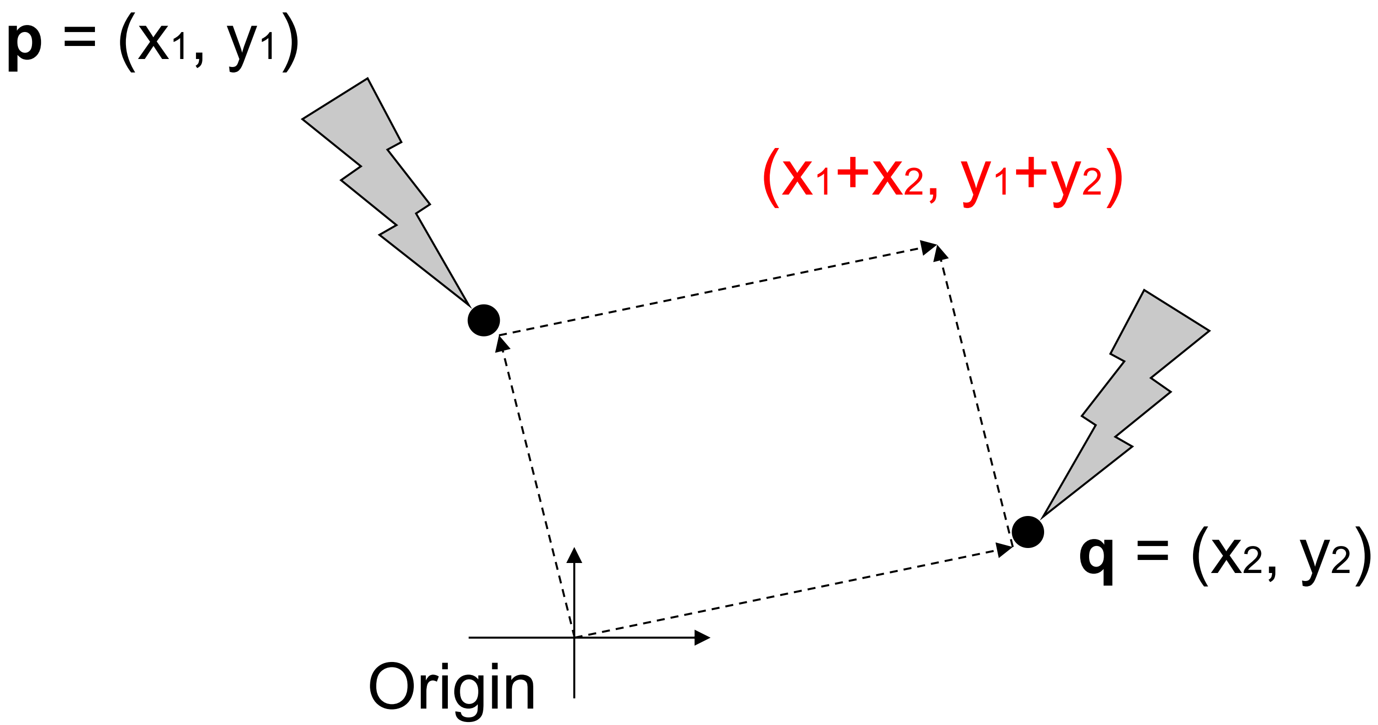


- Can we add these two positions？

Coordinates can be added but...

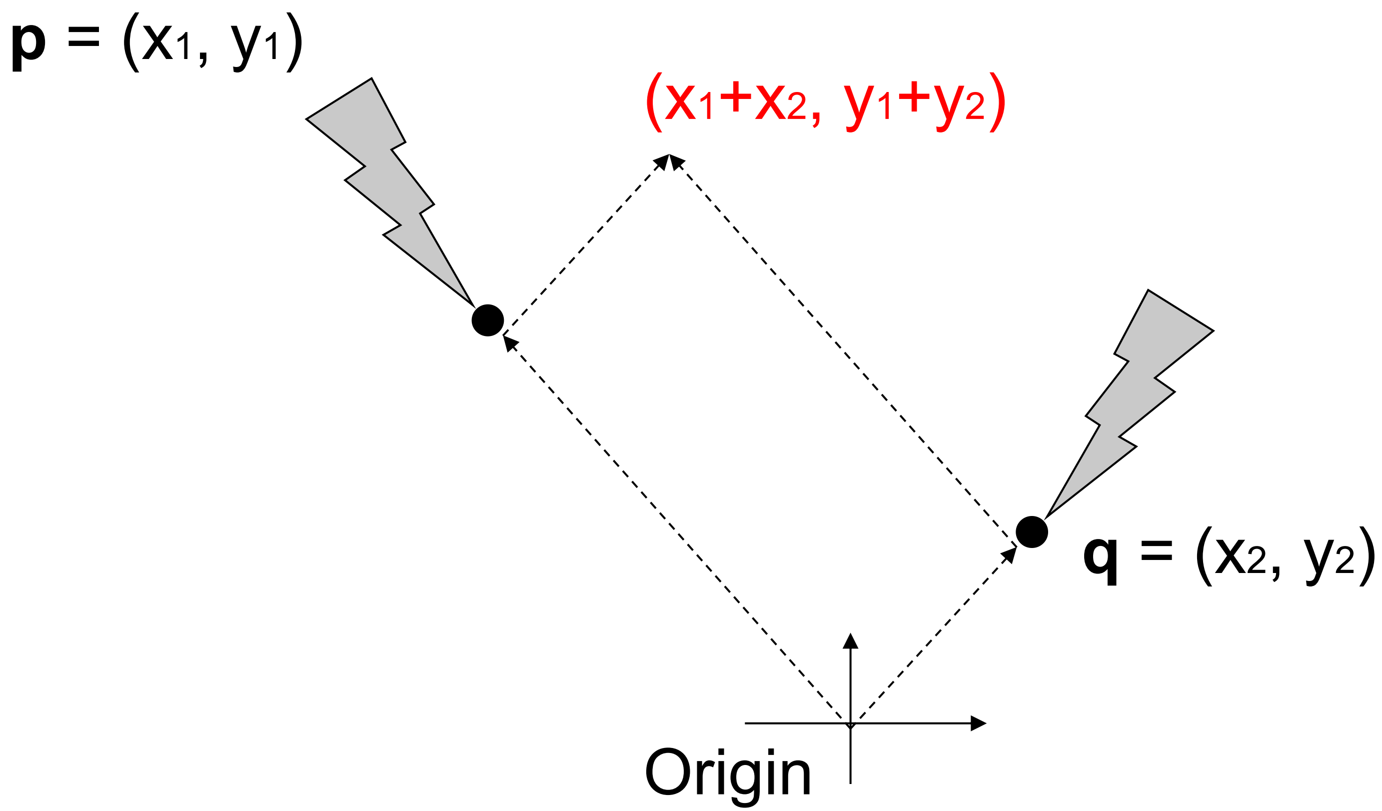


- The sum is (x_1+x_2, y_1+y_2)
  - Is it correct ?
  - Is it geometrically meaningful ?



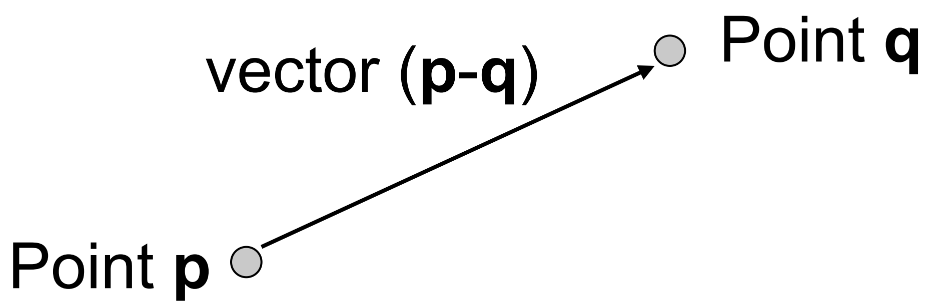
- Vector sum
  - (x_1, y_1) and (x_2, y_2) are considered as vectors from the origin to **p** and **q**, respectively.

If you choose 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 a ***coordinate-independent*** manner.

Points & Vectors are Different!

- Mathematically (and physically),
- *Points* are **locations in space**.
- *Vectors* are **displacements in space**.

- An analogy with time:
- *Times*, (or datetimes) are **locations in time**.
- *Durations* are **displacements in time**.

Vector and Affine Spaces

- ***Vector space***
  - Includes vectors and related operations
  - No points

- ***Affine space***
  - Includes vectors, points, and related operations

Vector spaces

- A ***vector space*** consists of
  - Set of vectors
  - Two operations: addition of vectors and multiplication of vectors by scalar numbers

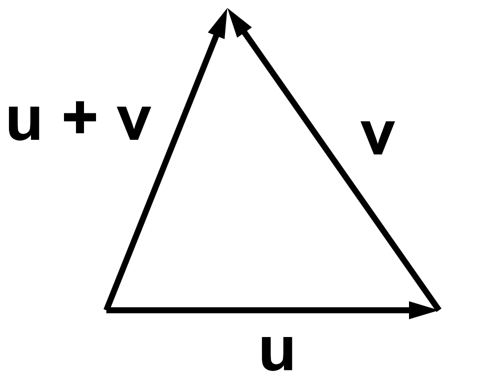
- A ***linear combination*** of vectors is also a vector

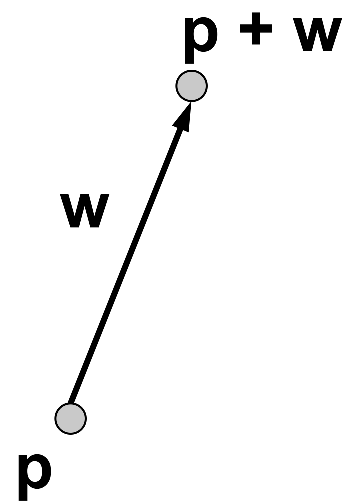
$$\mathbf{u}_0, \mathbf{u}_1, \cdots, \mathbf{u}_N \in V \quad \Rightarrow \quad c_0 \mathbf{u}_0 + c_1 \mathbf{u}_1 + \cdots + c_N \mathbf{u}_N \in V$$

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

Addition



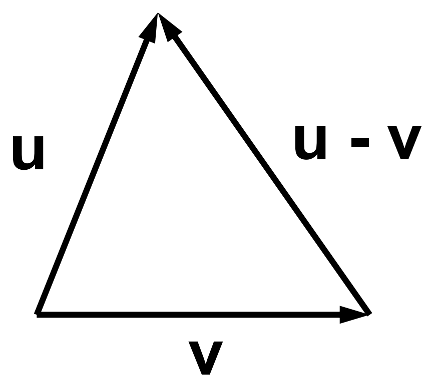


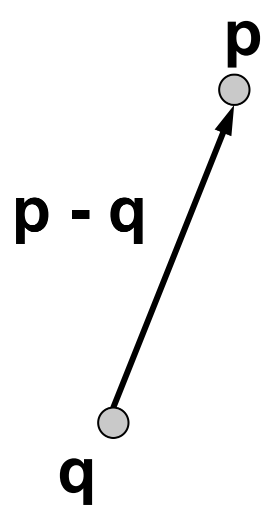
u + **v** is a vector

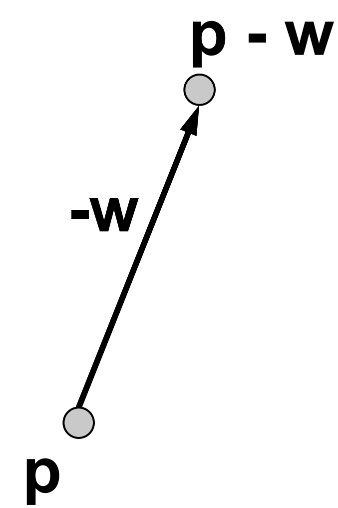
p + **w** is a point

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

$$\text{scalar} \bullet \text{vector} = \text{vector}$$

$$1 \bullet \text{point} = \text{point}$$

$$0 \bullet \text{point} = \text{vector}$$

$$c \bullet \text{point} = (\text{undefined}) \quad \text{if } (c \neq 0, 1)$$

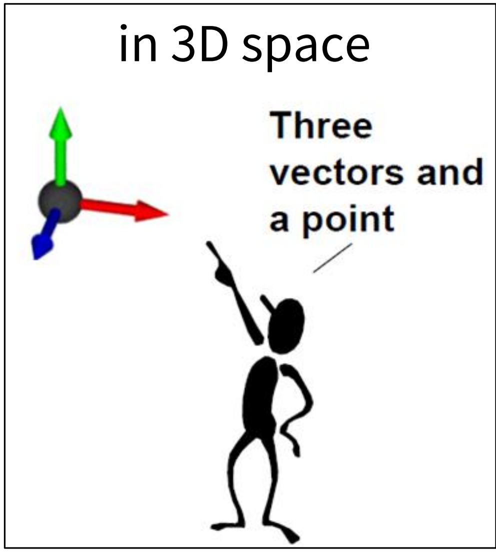
Affine Frame

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

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



Summary

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 3D spaces,
- A **point** is represented: $(x, y, z, 1)$
- A **vector** can be represented: $(x, y, z, 0)$

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

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*

A Consistent Model

- **Behavior of affine frame coordinates is completely consistent with our intuition**
  - **Subtracting two points yields a vector**
  - **Adding a vector to a point produces a point**
  - **If you multiply a vector by a scalar you still get a vector**
  - **Scaling points gives a nonsense 4th coordinate element in most cases**

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ 1 \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 - b_1 \\ a_2 - b_2 \\ a_3 - b_3 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ 1 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ 0 \end{bmatrix} = \begin{bmatrix} a_1 + v_1 \\ a_2 + v_2 \\ a_3 + v_3 \\ 1 \end{bmatrix}$$

KAIST

Transforming points and vectors

- Recall distinction points vs. vectors
  - vectors are just offsets (differences between points)
  - points have a location
    - represented by vector offset from a fixed origin
- Points and vectors transform differently
  - points respond to translation; vectors do not

$$\mathbf{v} = \mathbf{p} - \mathbf{q}$$

$$T(\mathbf{x}) = M\mathbf{x} + \mathbf{t}$$

$$\begin{aligned} T(\mathbf{p} - \mathbf{q}) &= M\mathbf{p} + \mathbf{t} - (M\mathbf{q} + \mathbf{t}) \\ &= M(\mathbf{p} - \mathbf{q}) + (\mathbf{t} - \mathbf{t}) = M\mathbf{v} \end{aligned}$$

Transforming points and vectors

- Homogeneous coords. let us exclude translation
  - just put 0 rather than 1 in the last place

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

point → point vector → vector

Note that translation is not applied to a vector!

Rigid motions

- A transform made up of only translation and rotation is a *rigid motion* or a *rigid body transformation*
- The linear part is an orthonormal matrix

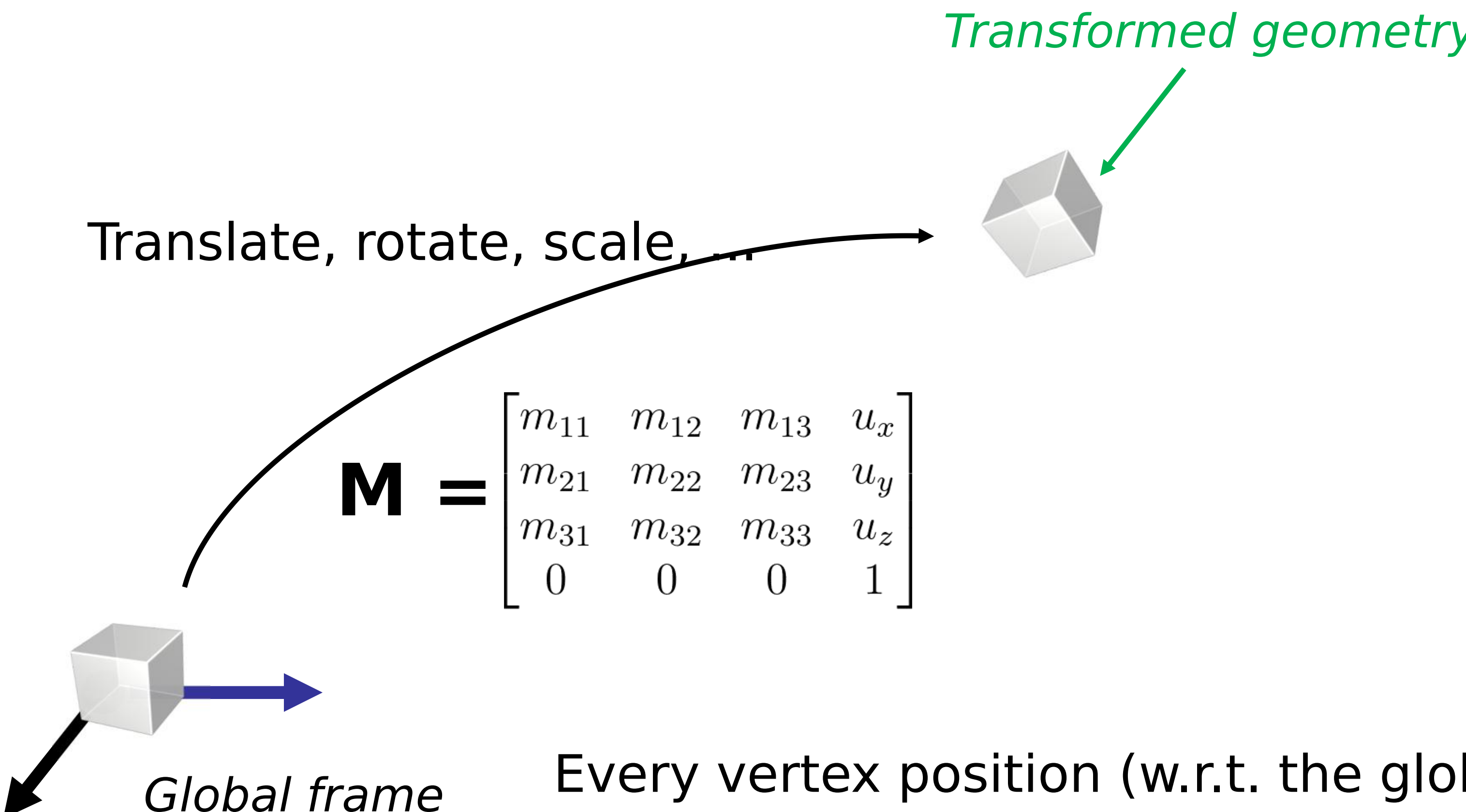
$$R = \begin{bmatrix} Q & \mathbf{u} \\ 0 & 1 \end{bmatrix}$$

- Inverse of orthonormal matrix is transpose
  – so inverse of rigid motion is easy:

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

Meanings of an Affine Matrix

1) A 4x4 Affine Transformation Matrix transforms a Geometry

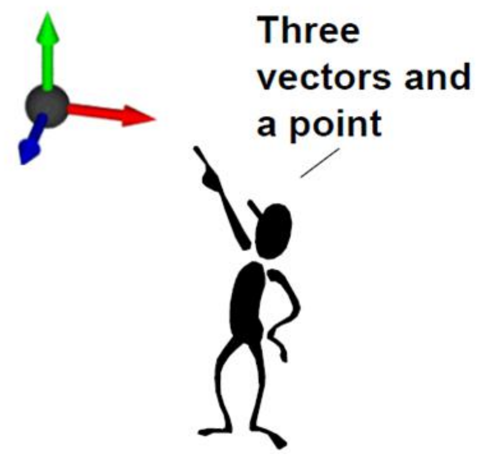


$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & u_x \\ m_{21} & m_{22} & m_{23} & u_y \\ m_{31} & m_{32} & m_{33} & u_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

Review: Affine Frame

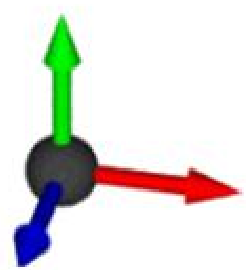
- An **affine frame** in 3D space is defined by three vectors and one point
  - Three vectors for x, y, z axes
  - One point for origin



Global Frame

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

$$\hat{\mathbf{e}}_y = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$$



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

$$\hat{\mathbf{e}}_x = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$$

$$\hat{\mathbf{e}}_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

Let's transform a global frame

- Apply M to a global frame, that is,
  - Multiply M with the x, y, z axis *vectors* and the origin *point* of the global 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*

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

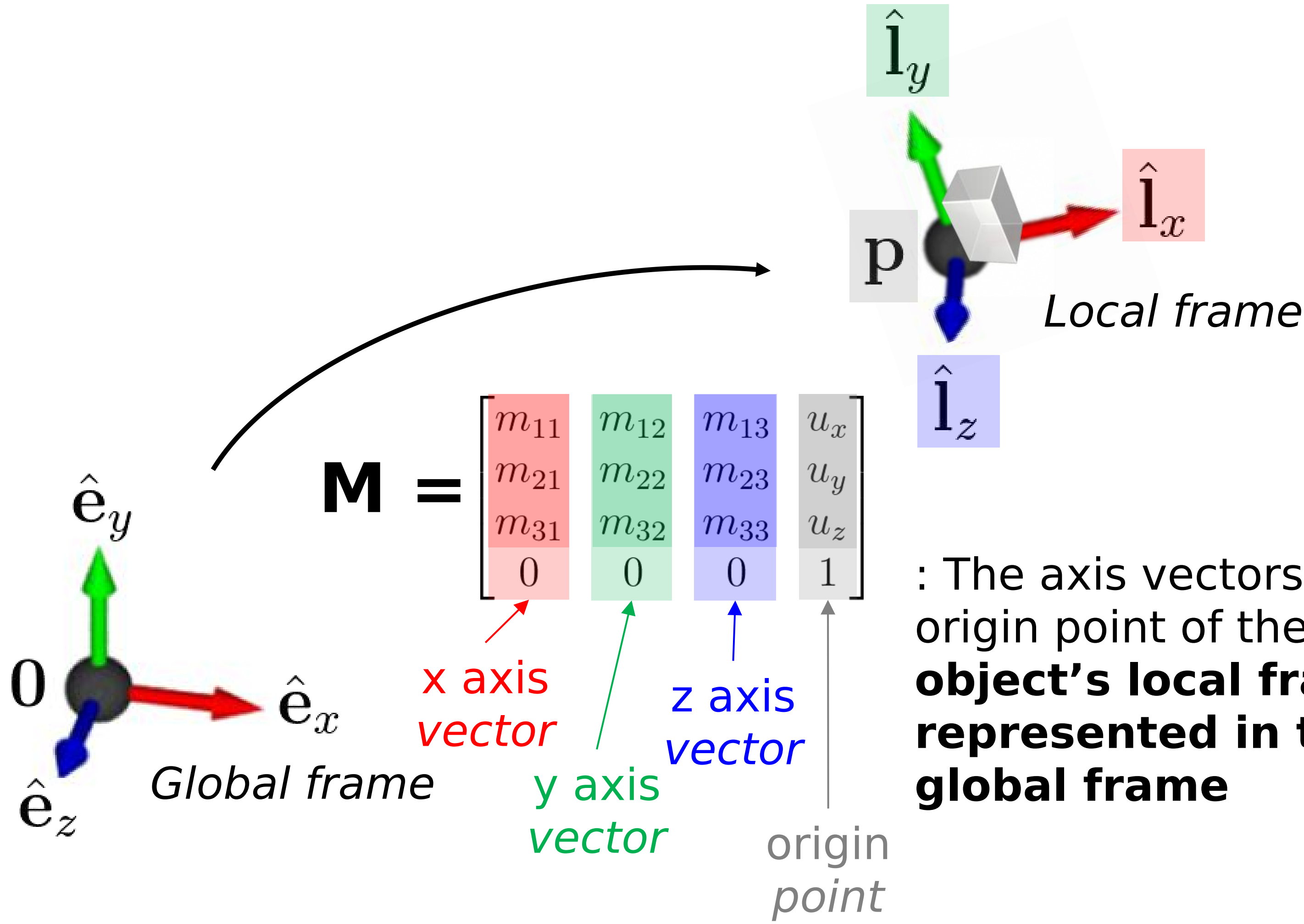
z axis *vector*

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

origin *point*

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

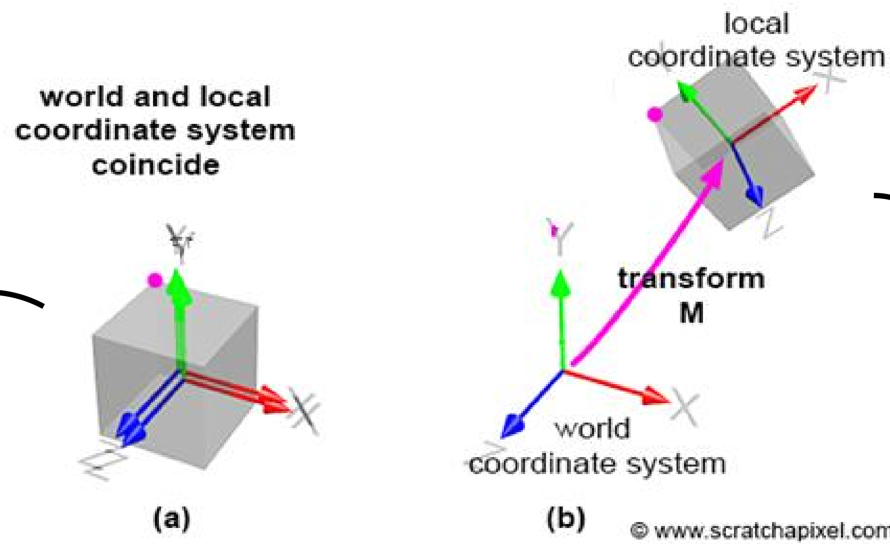
2) A 4x4 Affine Transformation Matrix defines an Affine Frame w.r.t. Global Frame



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

: The axis vectors and origin point of the **object's local frame represented in the global frame**

Examples



This local 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 local frame* ***represented in the global frame***

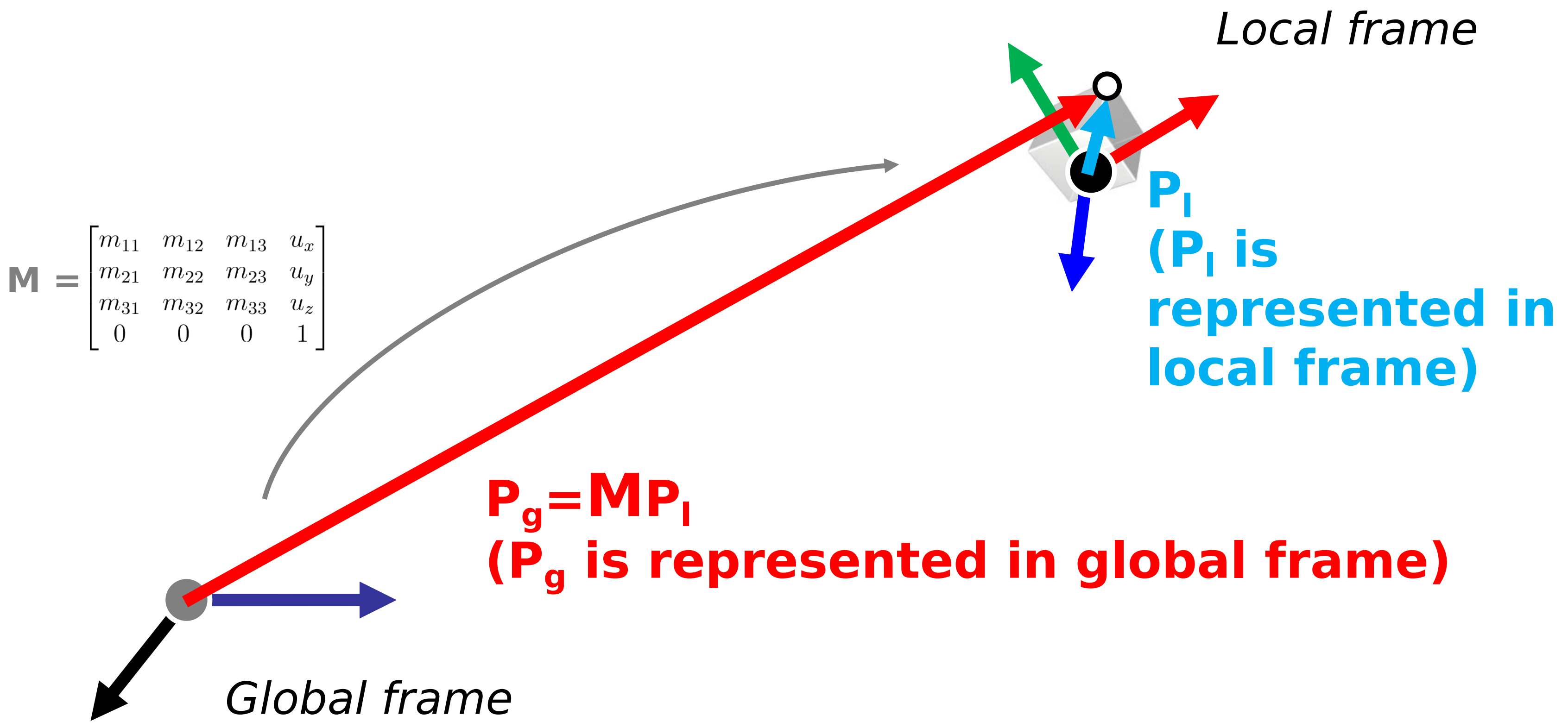
This local frame is defined by:

origin *point*

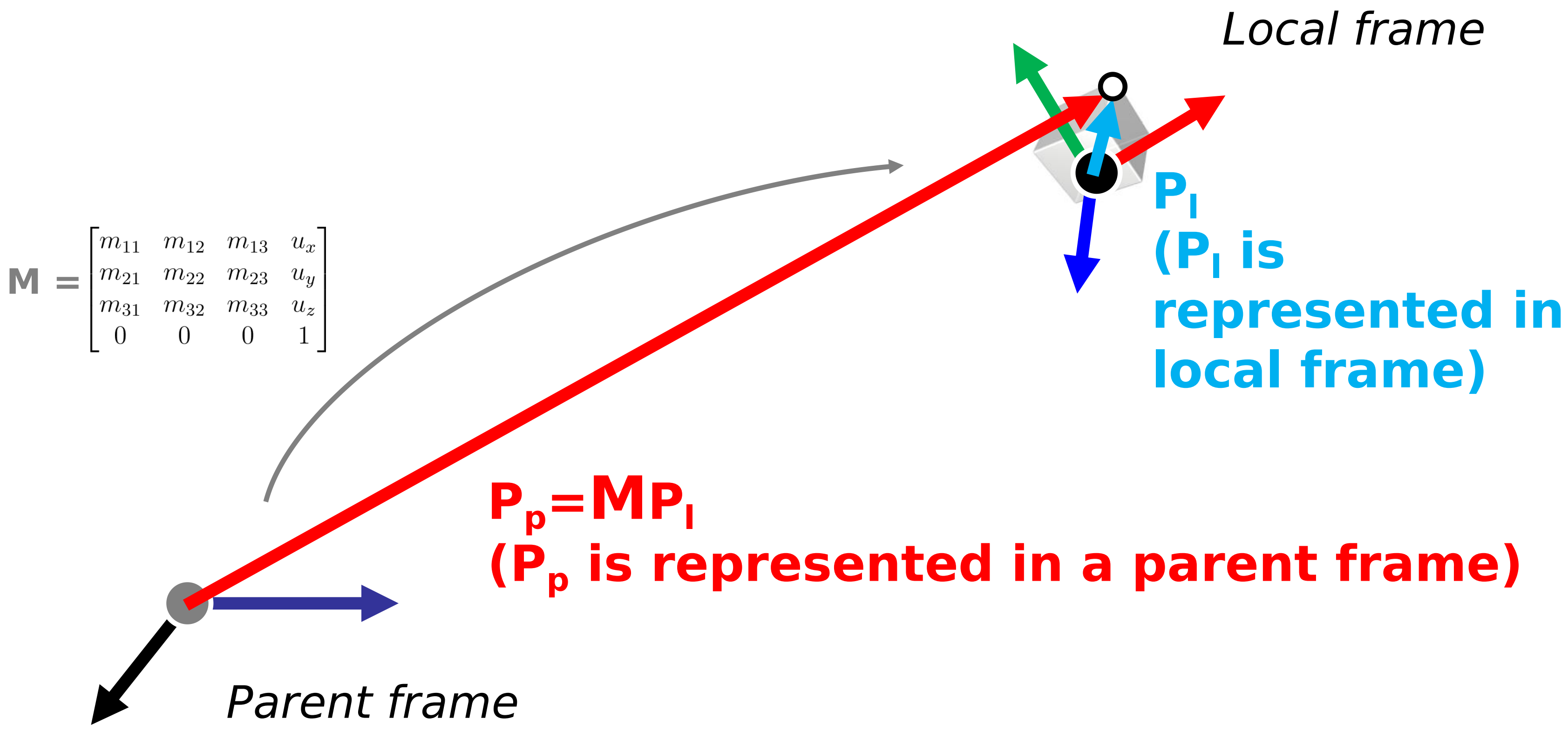
$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & u_1 \\ m_{21} & m_{22} & m_{23} & u_2 \\ m_{31} & m_{32} & m_{33} & u_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

x axis *vector* y axis *vector* z axis *vector*

3) A 4x4 affine transformation matrix transforms a point in one frame to a point represented in another frame



3) A 4x4 affine transformation matrix transforms a point in one frame to a point represented in another frame



Coordinate frame summary

- Frame = point plus basis
- Frame matrix (frame-to-canonical) is

$$F = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{p} \\ 0 & 0 & 1 \end{bmatrix}$$

- Move points to and from frame by multiplying with *F*

$$p_e = F p_F \quad p_F = F^{-1} p_e$$

- Move transformations using similarity transforms

$$T_e = F T_F F^{-1} \quad T_F = F^{-1} T_e F$$

Next Time

- Lab in this week:
  - Lab assignment 3

- Next lecture:
  - 5 - Rendering Pipeline

- Acknowledgement: Some materials come from the lecture slides of
  - Prof. Jehee Lee, SNU, http://mrl.snu.ac.kr/courses/CourseGraphics/index_2017spring.html
  - Prof. Sung-eui Yoon, KAIST, https://sglab.kaist.ac.kr/~sungeui/CG/