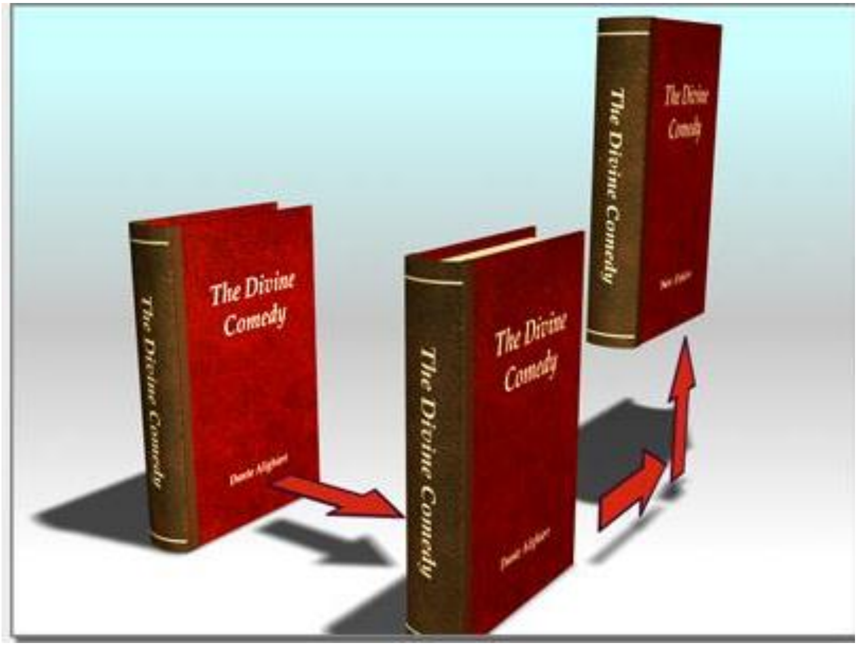
Advanced Computer Graphics

Lecture-04 Transformation

Tzung-Han Lin
National Taiwan University of Science and Technology
Graduate Institute of Color and Illumination Technology
e-mail: thl@mail.ntust.edu.tw

Transformation

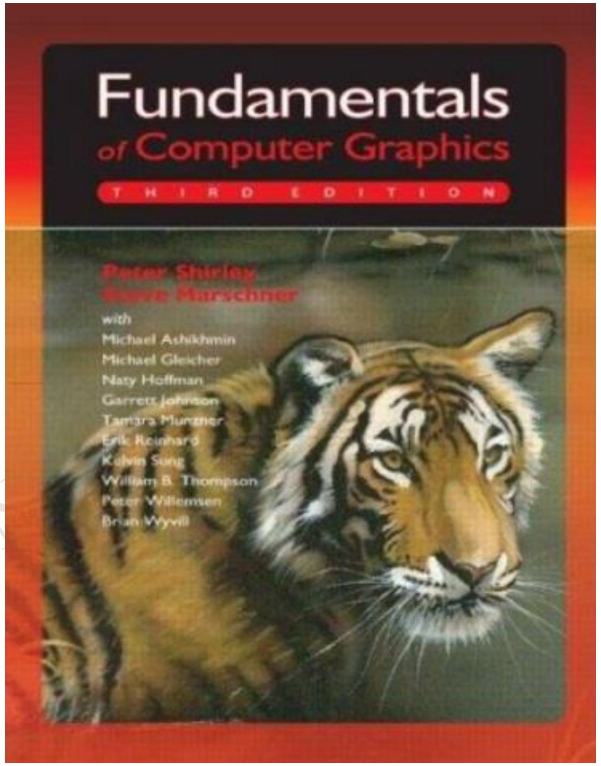
- 2D / 3D Object transformation
- 2D / 3D Coordinate transformation
- Performance computing

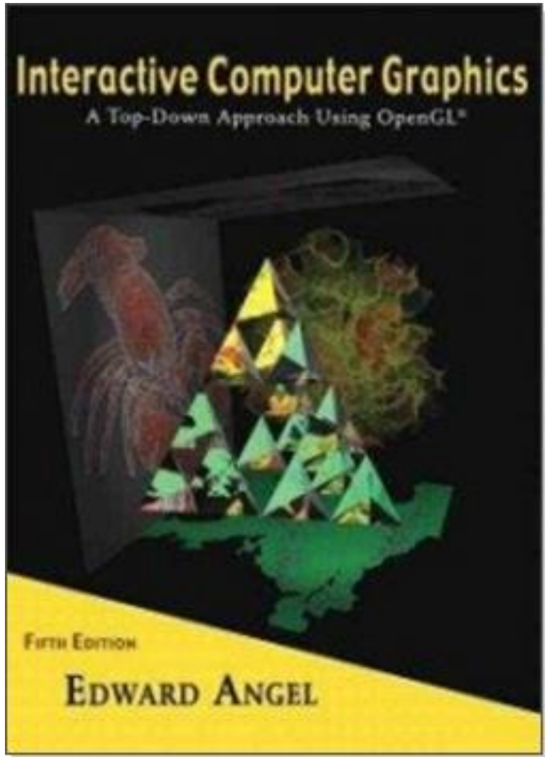


Picture from 3DS Max9 Reference Document

Content from textbook

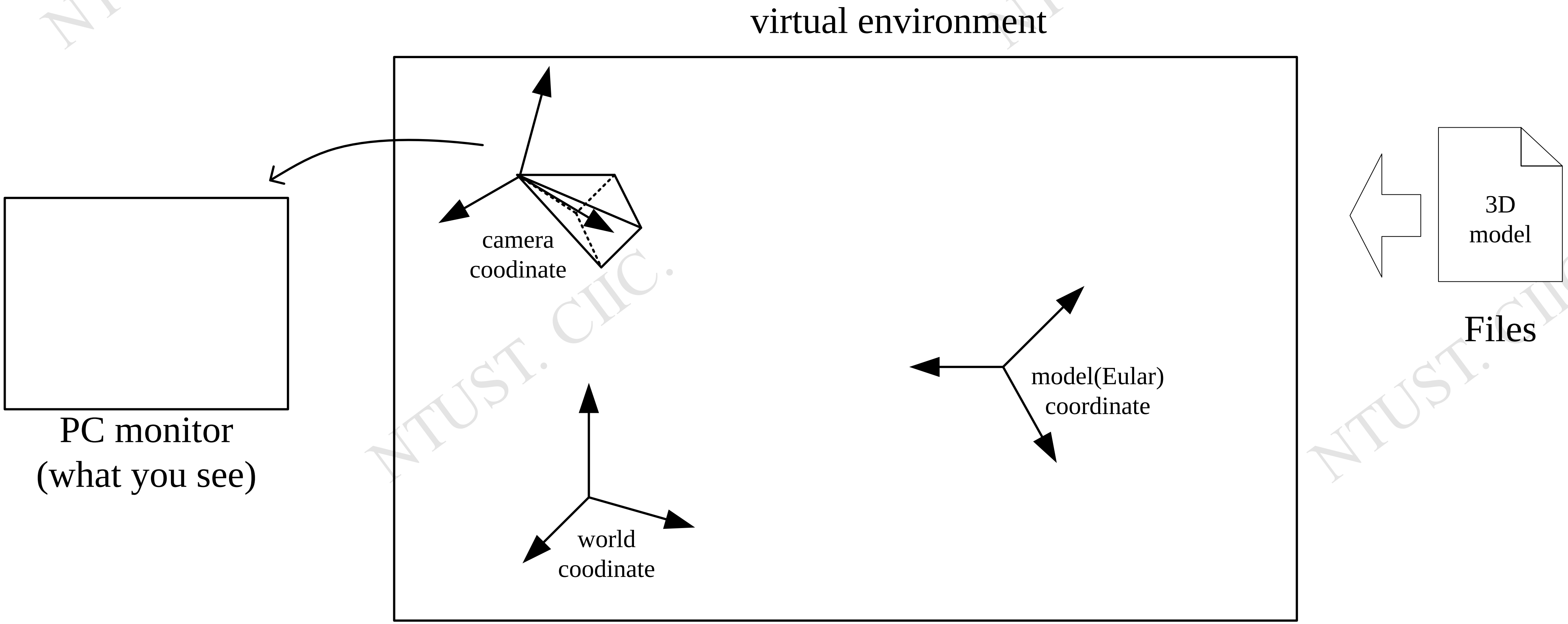
- Fundamentals of Computer Graphics, Chapter 6.
- Interactive Computer Graphics, 5th edition, Chapter 4.7.
- Online resource.





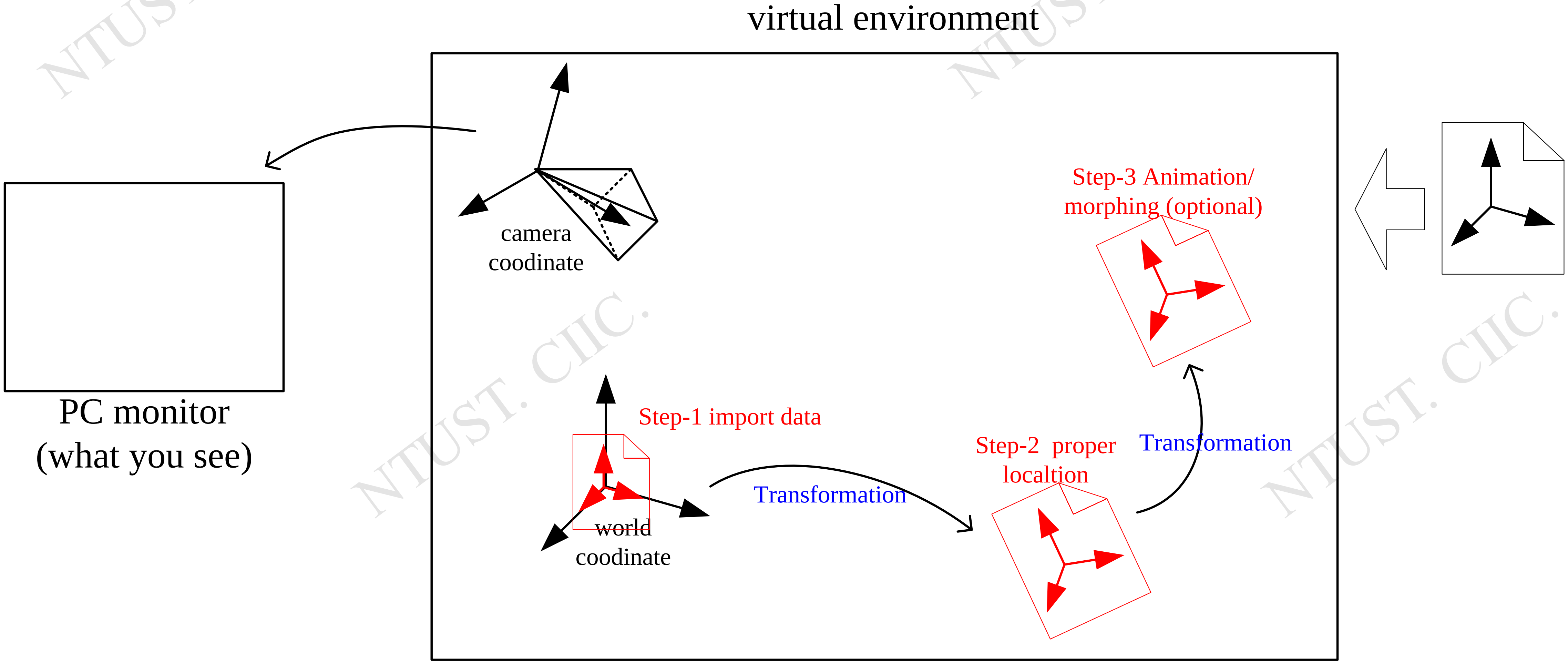
Transformation

- What you see in the monitor comes from "the result of transformation"



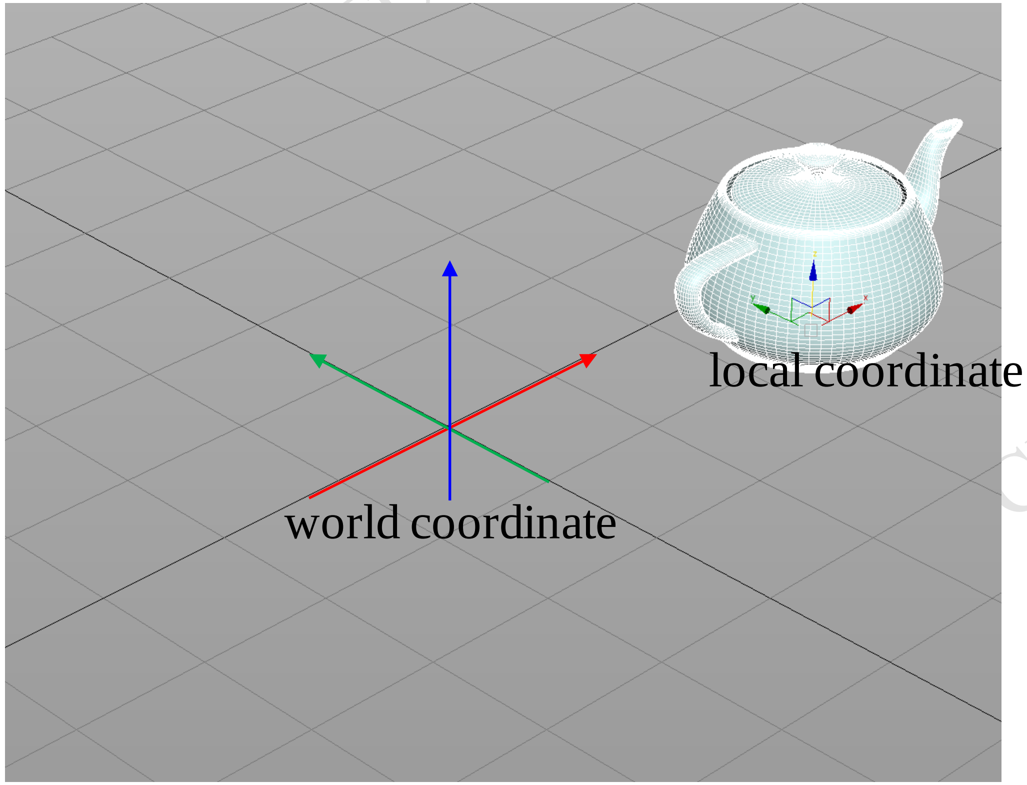
Transformation

- What you see in the monitor comes from “the result of transformation”



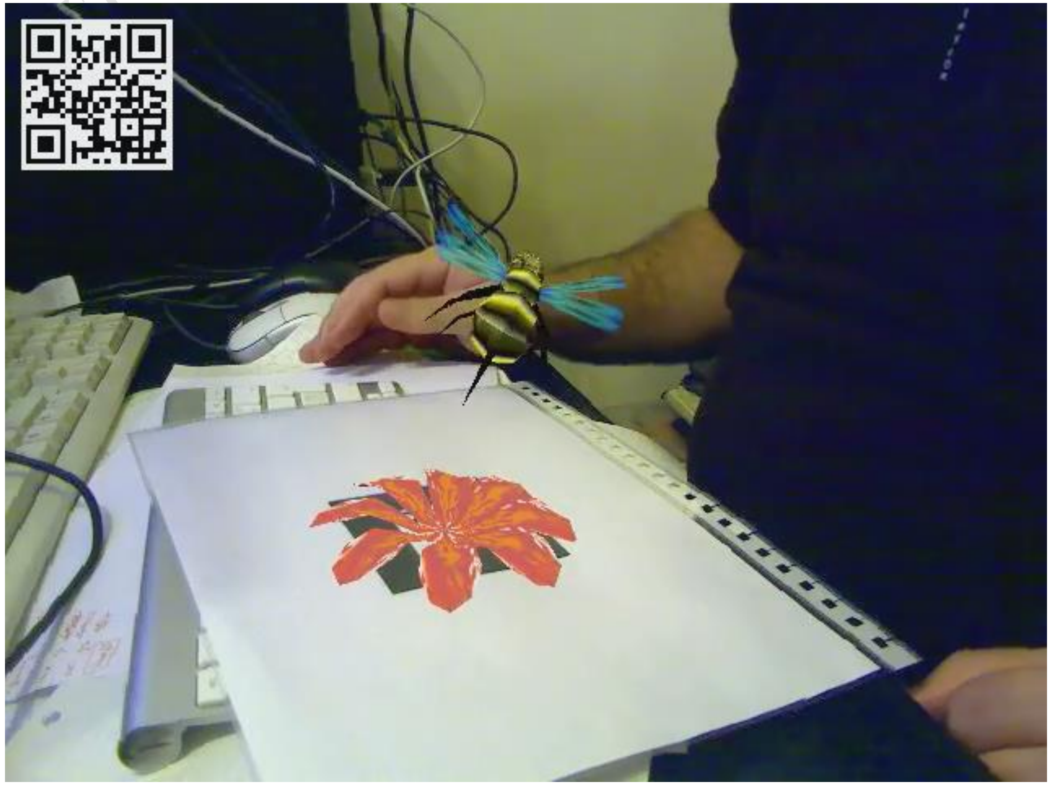
Transformation—example

- A teapot in either a “world coordinate” or “local coordinate”



Transformation—example

- Files include of motion scripts: dae, vrml



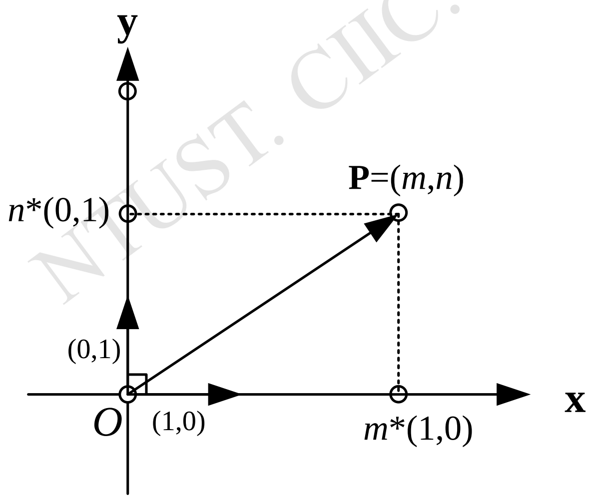
VRML file format in ARToolkit (library)

Rigid body motion

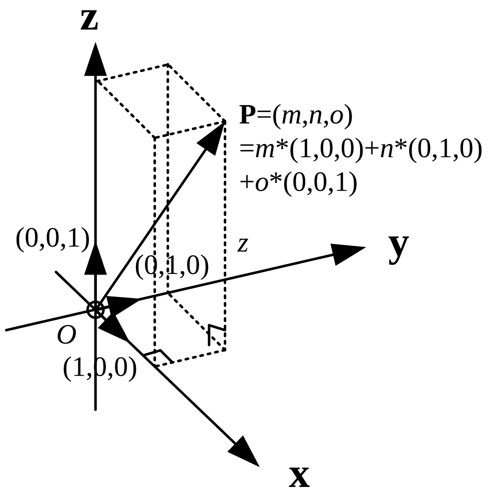
Coordinate definition in 2D / 3D

- Basis vector: The vectors are linearly independent.
- Every vector in this vector space (or called coordinate) is a linear combination of basis vectors.

$$\mathbf{i} = (1,0)$$
$$\mathbf{j} = (0,1)$$
$$\mathbf{P} = (m,n) = m(1,0) + n(0,1) = m\mathbf{i} + n\mathbf{j}$$



$$\mathbf{i} = (1,0,0)$$
$$\mathbf{j} = (0,1,0)$$
$$\mathbf{k} = (0,0,1)$$
$$\mathbf{P} = (m,n,o) = m(1,0,0) + n(0,1,0) + o(0,0,1) = m\mathbf{i} + n\mathbf{j} + o\mathbf{k}$$



Coordinate definition in 2D / 3D

- Representation of "vector": we usually use bold text to denote vector.
  - In 2D, three components (with a dummy) are used…
  - In 3D, four components are used to represent "homogenous" coordinate.

$$\mathbf{p} = \begin{bmatrix} x_p \\ y_p \end{bmatrix} \qquad \mathbf{p} = \begin{bmatrix} x_q \\ y_q \\ 1 \end{bmatrix}$$

$$\mathbf{q} = \begin{bmatrix} x_q \\ y_q \\ z_q \end{bmatrix} \qquad \mathbf{q} = \begin{bmatrix} x_q \\ y_q \\ z_q \\ 1 \end{bmatrix}$$

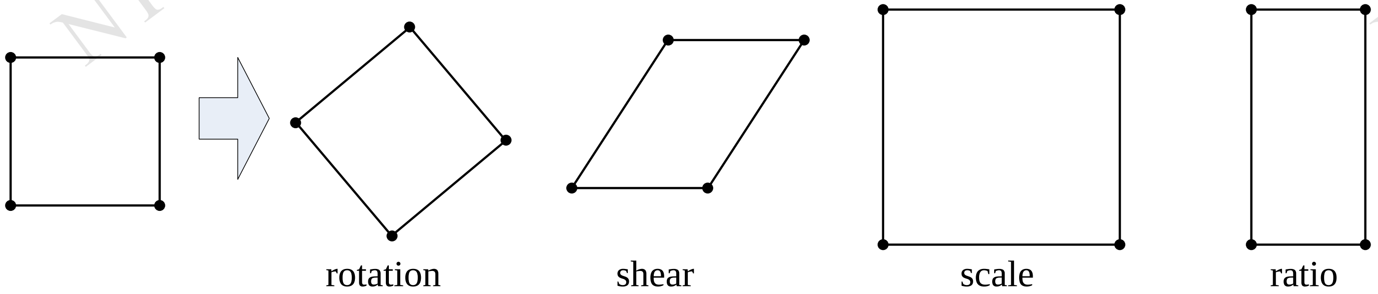
Mapping and affine transformation in 2D

- 2D transformation is one kind of affine transformation (or mapping). If a 2D vector **p** was transformed by an affine transformation **A** into **p***, it is shown as

$$\mathbf{p}^* = \mathbf{A}\mathbf{p}$$

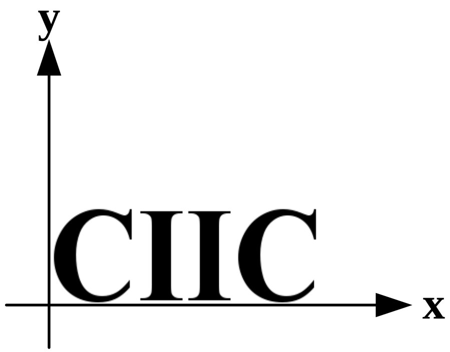
$$\mathbf{p}^* = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} a_{11}x_p + a_{12}y_p \\ a_{21}x_p + a_{22}y_p \end{bmatrix} = \begin{bmatrix} x_p^* \\ y_p^* \end{bmatrix}$$

- From the equation, the new vector $[x_p^*, y_p^*]$ is a linear combination of $[x_p, y_p]$ and A. As a result, all conditions are: rotation, shear, scale, and ratio.

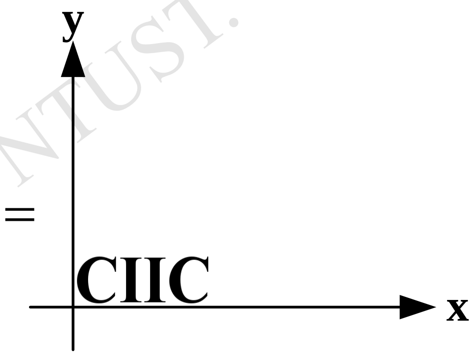


Mapping and affine transformation in 2D

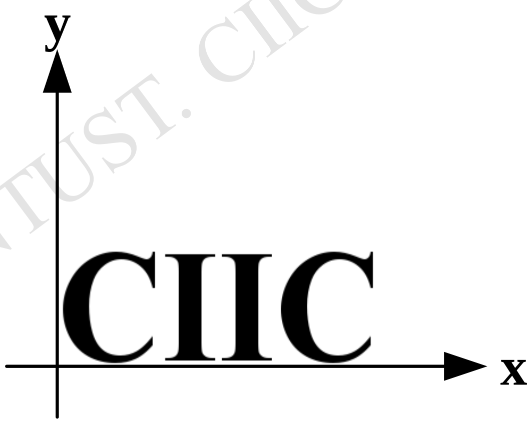
- Scaling examples



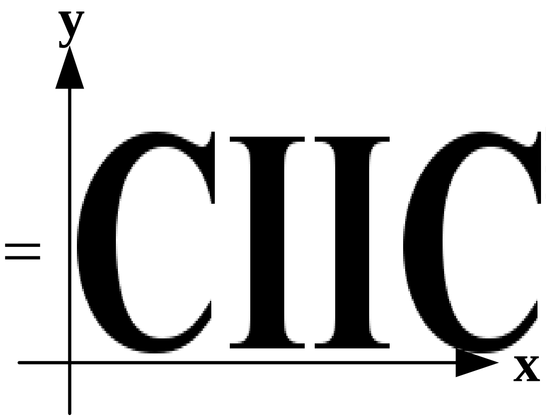
$$\times \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} =$$



Uniform scaling



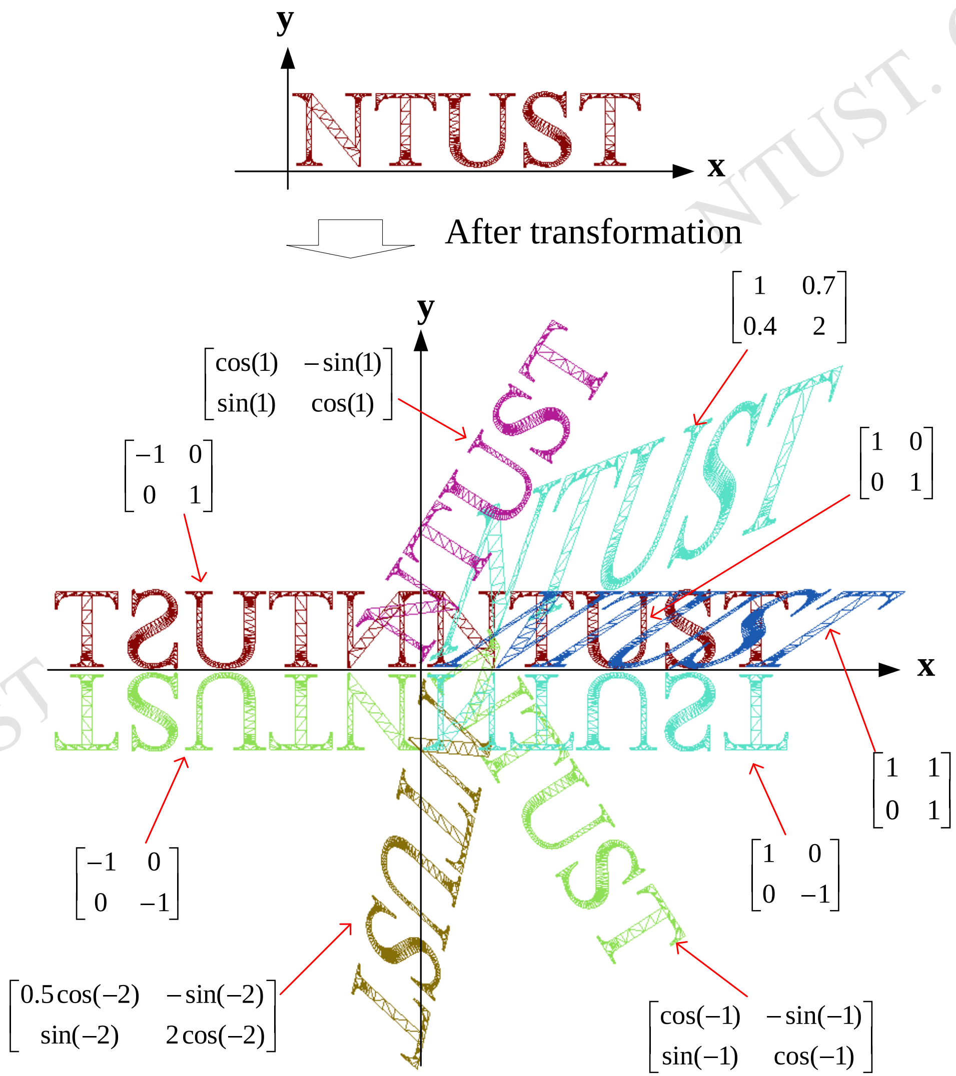
$$\times \begin{bmatrix} 1.5 & 0 \\ 0 & 2 \end{bmatrix} =$$



Non-uniform scaling

Mapping and affine transformation in 2D

- Examples:



Mapping and affine transformation in 2D

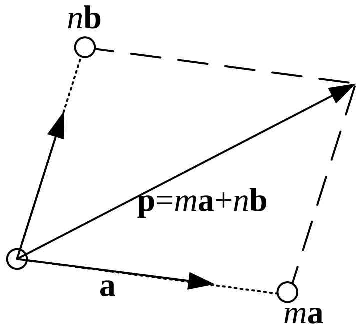
- The new coordinate p* could be a linear combination of two vectors (says $\mathbf{a}=[a_{11}, a_{21}]^T$ and $\mathbf{b}=[a_{12}, a_{22}]^T$) and the original components (says x_p and y_p).

$$\mathbf{p}^* = \begin{bmatrix} x_p^* \\ y_p^* \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} a_{11}x_p + a_{12}y_p \\ a_{21}x_p + a_{22}y_p \end{bmatrix} = x_p \begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix} + y_p \begin{bmatrix} a_{12} \\ a_{22} \end{bmatrix}$$

$$\mathbf{p}^* = x_p\mathbf{a} + y_p\mathbf{b}$$

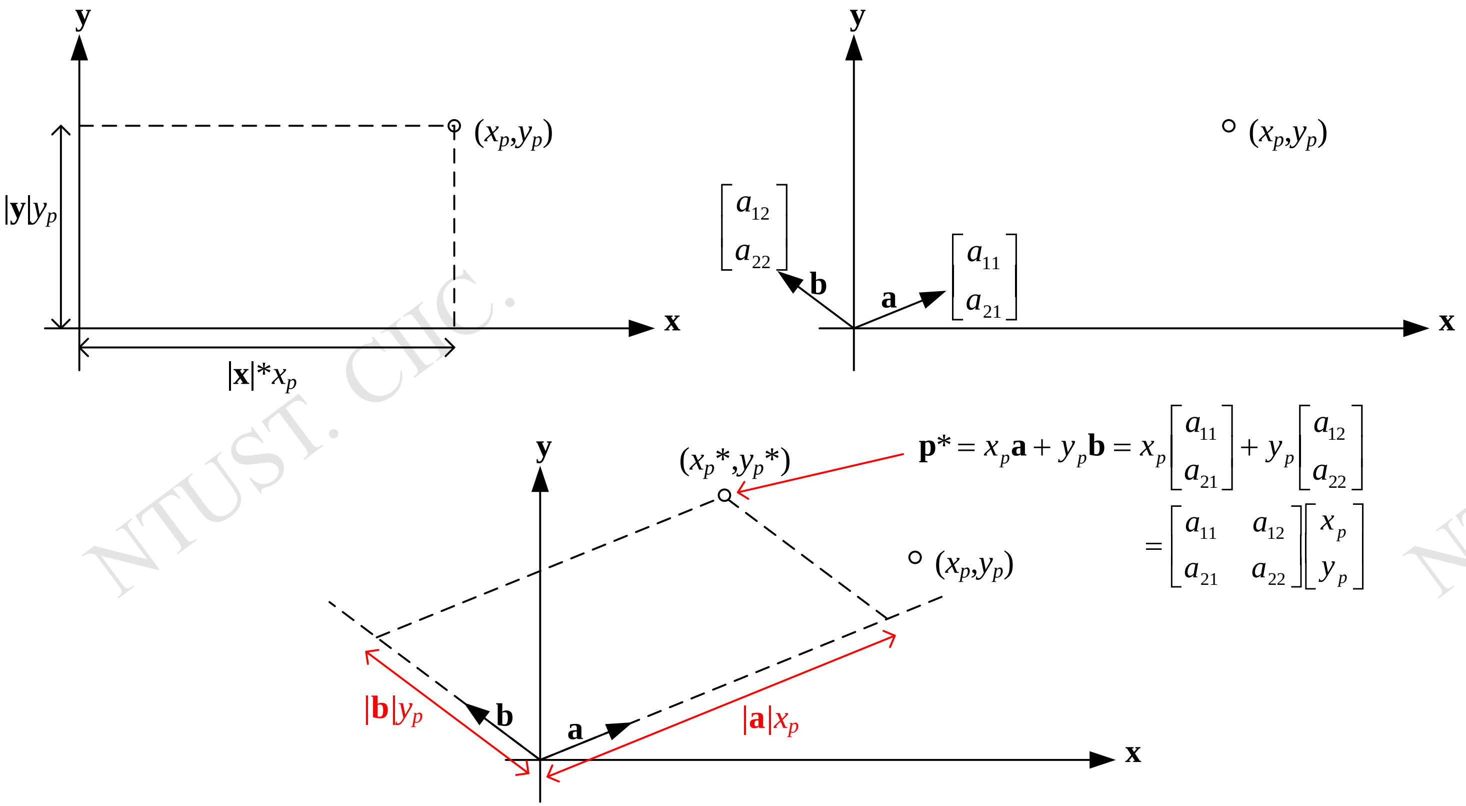
$$\mathbf{a} = \begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix}, \text{and } \mathbf{b} = \begin{bmatrix} a_{12} \\ a_{22} \end{bmatrix}$$

Recall the concept in previous slides:



Mapping and affine transformation in 2D

- In another words, the new coordinate p* can be determined by "Parallelogram rule". Thus, $\mathbf{p}^* = x_p\,\mathbf{a} + y_p\,\mathbf{b}$.

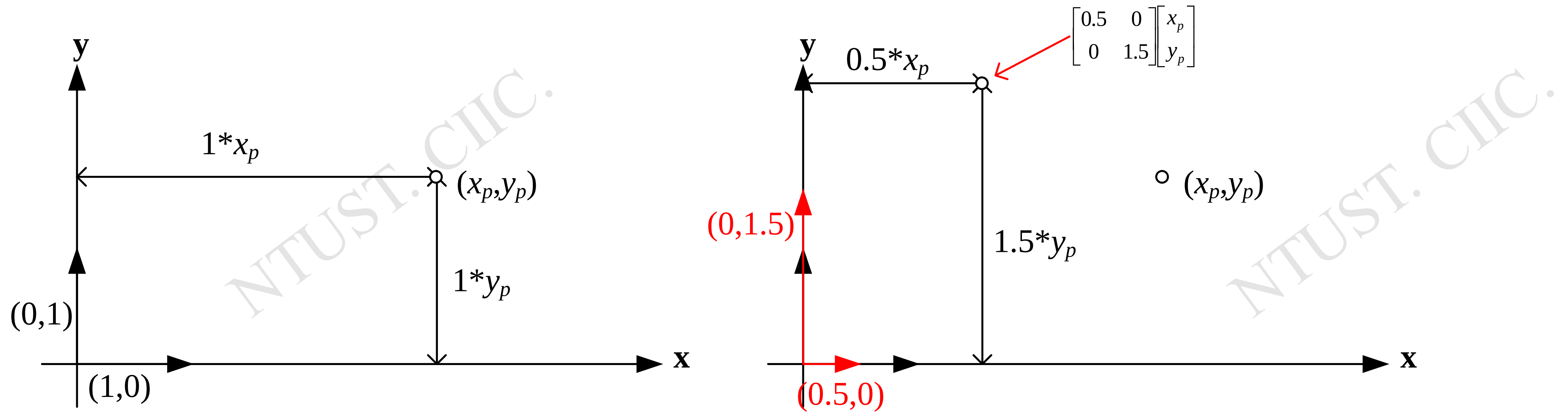


$$\mathbf{p}^* = x_p\mathbf{a} + y_p\mathbf{b} = x_p\begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix} + y_p\begin{bmatrix} a_{12} \\ a_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} x_p \\ y_p \end{bmatrix}$$

Mapping and affine transformation in 2D

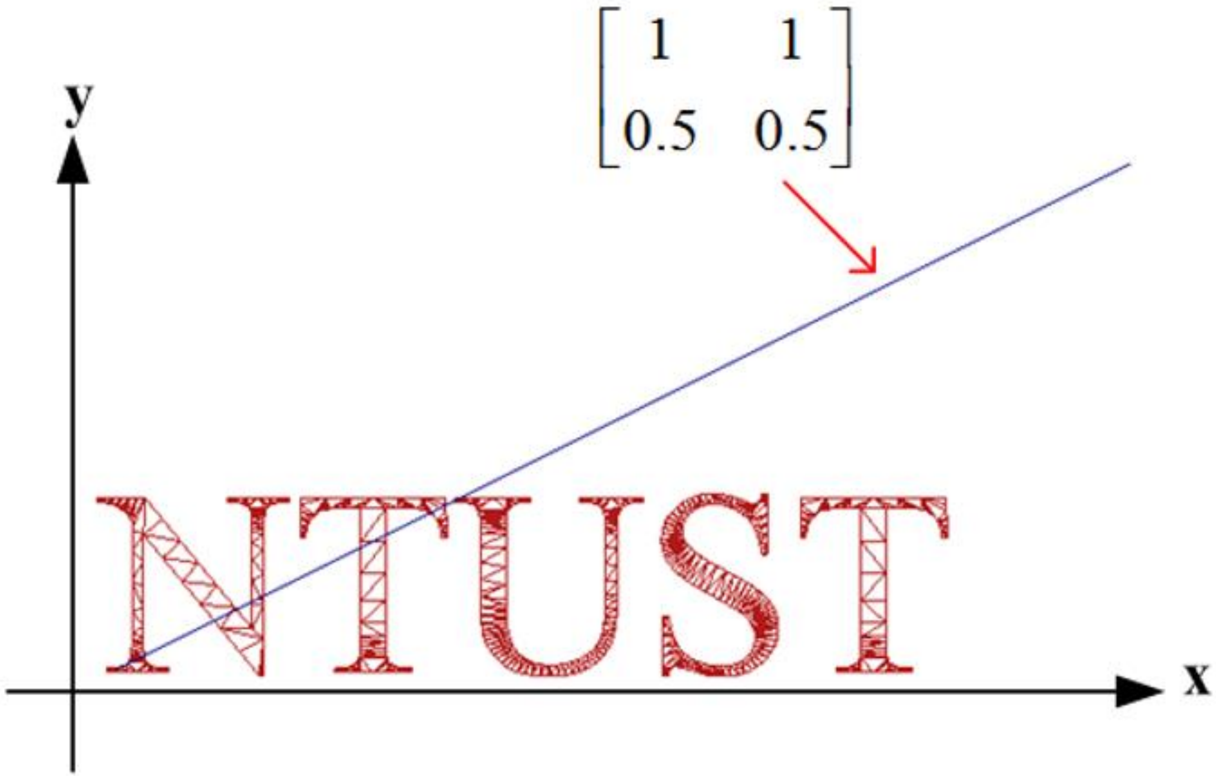
- An example:

$$\begin{bmatrix} 0.5 & 0 \\ 0 & 1.5 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \end{bmatrix} = x_p \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} + y_p \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}$$



Mapping and affine transformation in 2D

- A degenerate case:
  - If the transformation matrix rank is 1



Mapping and affine transformation in 2D

- A "scaling" operation in 2D case

$$\mathbf{p}^* = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \mathbf{p}$$

Matrix	Effect
$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	Do nothing
$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	Mirror along **x** axis
$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	Mirror along **y** axis
$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$	Rotation with 180 degrees

Mapping and affine transformation in 2D

- A "rotation" transformation
  - consider two orthogonal unit vectors as the new bases:

$$\mathbf{p}^* = \begin{bmatrix} \cos\theta & \cos(\theta+\frac{\pi}{2}) \\ \sin\theta & \sin(\theta+\frac{\pi}{2}) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} x + \begin{bmatrix} \cos(\theta+\frac{\pi}{2}) \\ \sin(\theta+\frac{\pi}{2}) \end{bmatrix} y$$

$$\mathbf{p}^* = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} (\cos\theta)x - (\sin\theta)y \\ (\sin\theta)x + (\cos\theta)y \end{bmatrix}$$

Mapping and affine transformation in 2D

- A “rotation” transformation
  - Proof: the distance to original is equal

$$\sqrt{((\cos\theta)x-(\sin\theta)y)^2+((\sin\theta)x+(\cos\theta)y)^2}$$

$$=\sqrt{x^2+y^2}\sqrt{((\cos\theta)\frac{x}{\sqrt{x^2+y^2}}-(\sin\theta)\frac{y}{\sqrt{x^2+y^2}})^2+((\sin\theta)\frac{x}{\sqrt{x^2+y^2}}+(\cos\theta)\frac{y}{\sqrt{x^2+y^2}})^2}$$

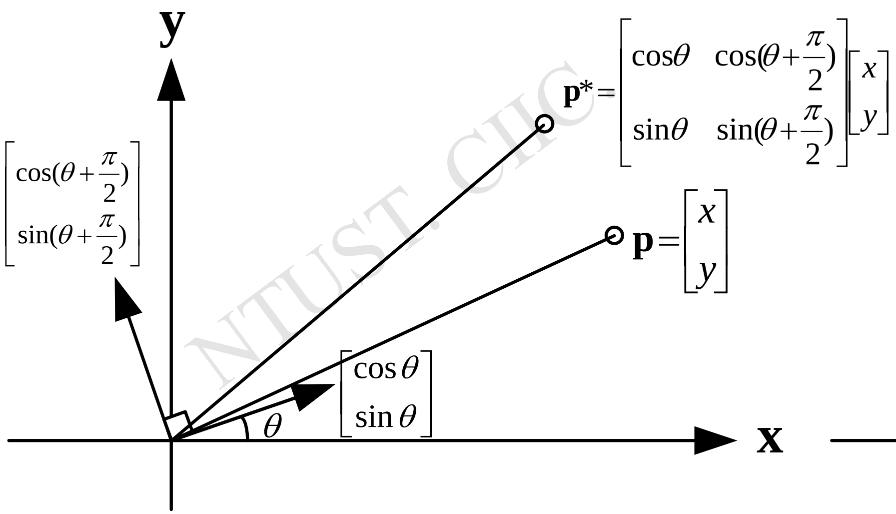
$$=\sqrt{x^2+y^2}\sqrt{(\cos\theta\cos\varpi-\sin\theta\sin\varpi)^2+(\sin\theta\cos\varpi+\cos\theta\sin\varpi)^2}$$

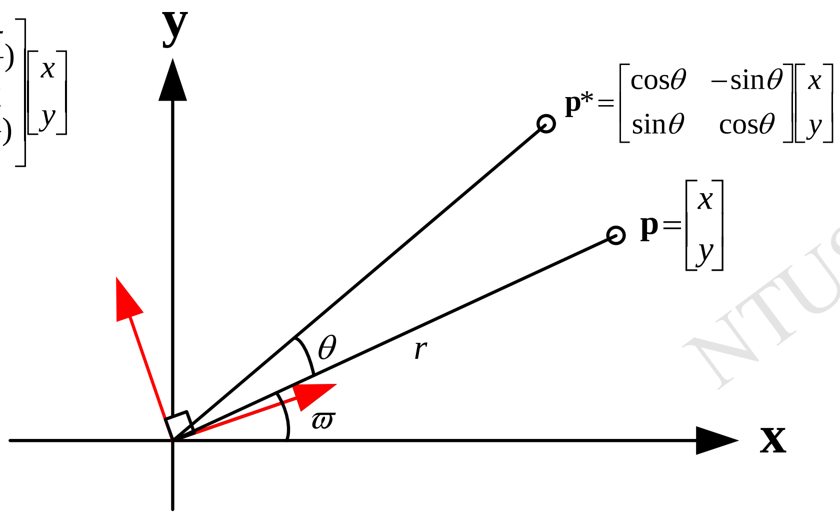
$$=\sqrt{x^2+y^2}\sqrt{\cos^2(\theta+\varpi)+\sin^2(\theta+\varpi)}$$

$$=\sqrt{x^2+y^2}=r$$

Rotation in 2D

- Summary:
  - 2D rotation: $$\mathbf{p}^* = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}\mathbf{p}$$
  - Inverse rotation: $$\mathbf{p} = \begin{bmatrix} \cos-\theta & -\sin-\theta \\ \sin-\theta & \cos-\theta \end{bmatrix}\mathbf{p}^* = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}^{-1}\mathbf{p}^*$$





Rotation in 2D

- Combination of rotations

$$\mathbf{p}_1 = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \mathbf{p}_0$$

$$\mathbf{p}_2 = \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix} \mathbf{p}_1$$

Then

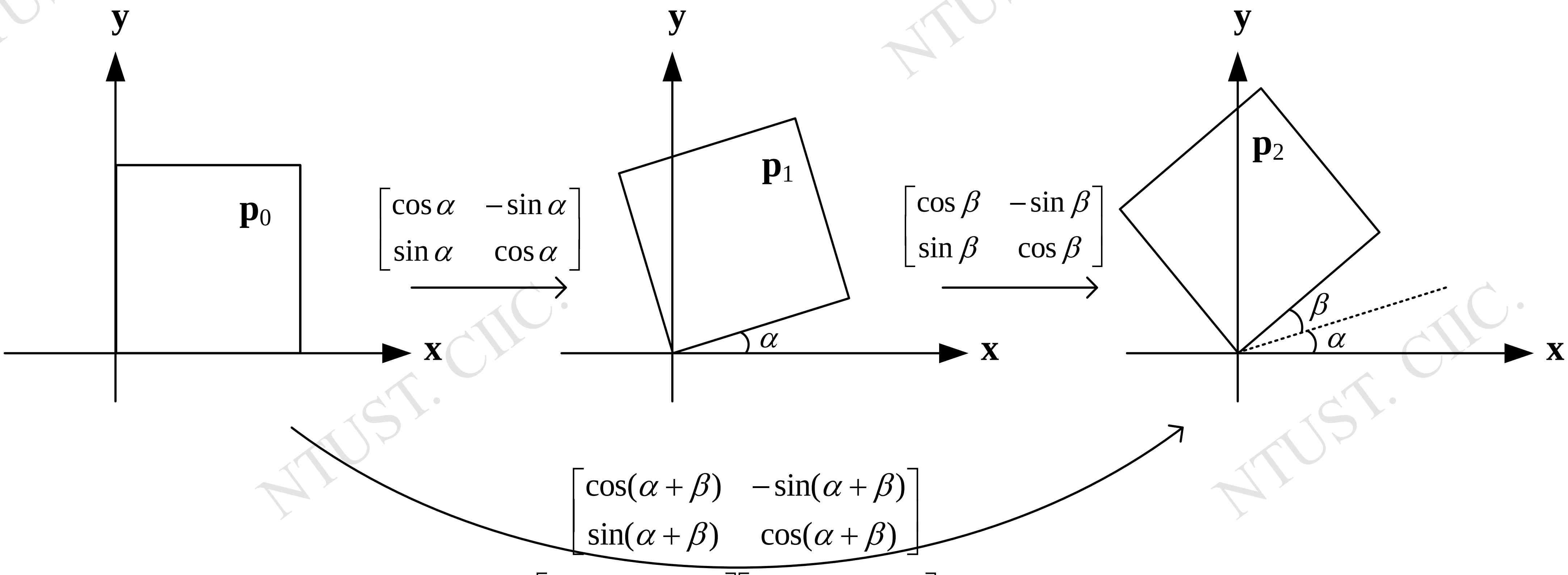
$$\mathbf{p}_2 = \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \mathbf{p}_0$$

Finally,

$$\mathbf{p}_2 = \begin{bmatrix} \cos(\alpha+\beta) & -\sin(\alpha+\beta) \\ \sin(\alpha+\beta) & \cos(\alpha+\beta) \end{bmatrix} \mathbf{p}_0$$

Rotation in 2D

- Rotation with 2 steps



$$\begin{bmatrix} \cos(\alpha+\beta) & -\sin(\alpha+\beta) \\ \sin(\alpha+\beta) & \cos(\alpha+\beta) \end{bmatrix}$$

$$\text{OR} \quad \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix}\begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}$$

Rotation in 2D

- Rotation in "Matrix form"
  - A dummy vector (homogenous coordinate) is used for matrix-vector operator. It will be much convenient and programmable.

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{n\theta} = (R_\theta)^n = \begin{bmatrix} \cos n\theta & -\sin n\theta & 0 \\ \sin n\theta & \cos n\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{2\theta} = R_\theta R_\theta = (R_\theta)^2 = \begin{bmatrix} \cos 2\theta & -\sin 2\theta & 0 \\ \sin 2\theta & \cos 2\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

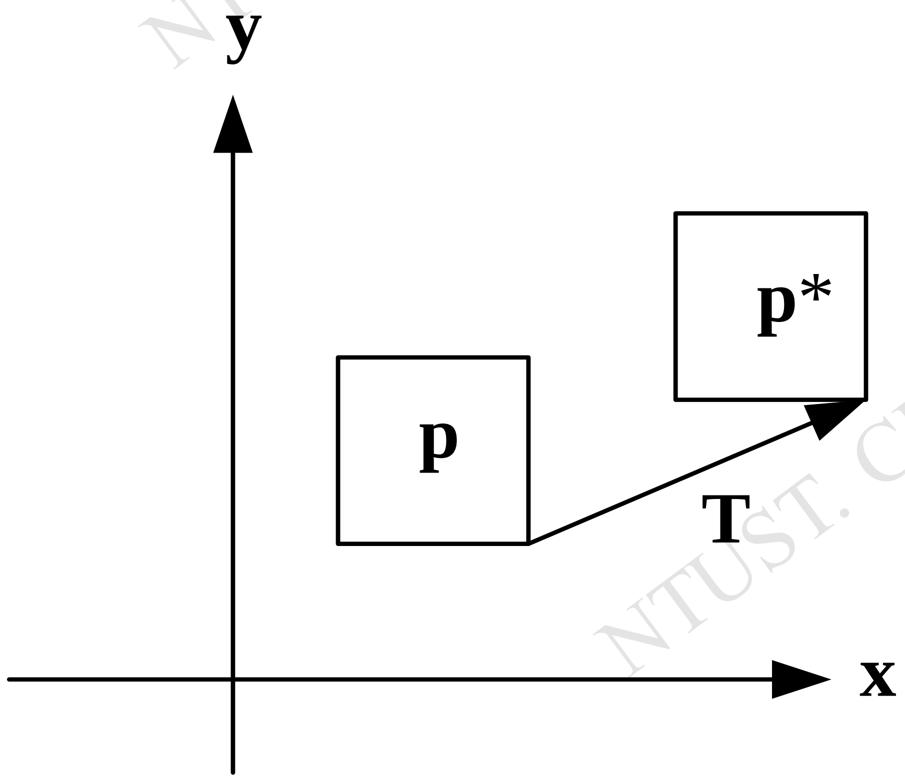
$$R_{\alpha+\beta} = R_\alpha R_\beta = \begin{bmatrix} \cos(\alpha+\beta) & -\sin(\alpha+\beta) & 0 \\ \sin(\alpha+\beta) & \cos(\alpha+\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Translation in 2D

- To translate an object, just simply add a vector to all vertex of the object.

$$\mathbf{p}^* = \mathbf{p} + \mathbf{T}$$

$$\begin{bmatrix} x_p^* \\ y_p^* \end{bmatrix} = \begin{bmatrix} x_p \\ y_p \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$



2D transformation

- Summary

Scaling

$$\begin{bmatrix} x_p^* \\ y_p^* \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \quad \text{or} \quad \mathbf{p}^* = \mathbf{S} \cdot \mathbf{p}$$

Rotation

$$\begin{bmatrix} x_p^* \\ y_p^* \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \quad \text{or} \quad \mathbf{p}^* = \mathbf{R} \cdot \mathbf{p}$$

Translation

$$\begin{bmatrix} x_p^* \\ y_p^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \quad \text{or} \quad \mathbf{p}^* = \mathbf{T} \cdot \mathbf{p}$$

2D transformation

- Inverse transformation

Inverse scaling $\begin{bmatrix} \frac{1}{s_x} & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix}$ or $\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1}$

Inverse rotation $\begin{bmatrix} \cos(-\theta) & -\sin(-\theta) & 0 \\ \sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$ or $\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1}$

Inverse translation $\begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix}$ or $\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}^{-1}$

Linear algebra (remind)

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

$$(\mathbf{A}\mathbf{B})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$$

$$(\mathbf{A}\mathbf{B})^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}$$

$$k(\mathbf{A}\mathbf{B}) = (k\mathbf{A})\mathbf{B} = \mathbf{A}(k\mathbf{B})$$

$$\mathbf{A}(\mathbf{B}\mathbf{C}) = (\mathbf{A}\mathbf{B})\mathbf{C}$$

$$(\mathbf{A}+\mathbf{B})\mathbf{C} = \mathbf{A}\mathbf{C} + \mathbf{B}\mathbf{C}$$

$$\mathbf{C}(\mathbf{A}+\mathbf{B}) = \mathbf{C}\mathbf{A} + \mathbf{C}\mathbf{B}$$

Linear algebra (remind)

- For square matrix

Symmetric: $\mathbf{A}^T = \mathbf{A}$

Skew-symmetric: $\mathbf{A}^T = -\mathbf{A}$

Orthogonal: $\mathbf{A}^T = \mathbf{A}^{-1}$

$$\mathbf{R} = \frac{1}{2}(\mathbf{A} + \mathbf{A}^T)$$ →Symmetric

$$\mathbf{R} = \frac{1}{2}(\mathbf{A} - \mathbf{A}^T)$$ →Skew-symmetric

2D transformation (example)

- Example: An object, p, is initially scaled by S, then translated by T, and finally rotated by R. Thus, the position of this object will be

$$\mathbf{p}^* = \mathbf{R} \cdot \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{p}$$

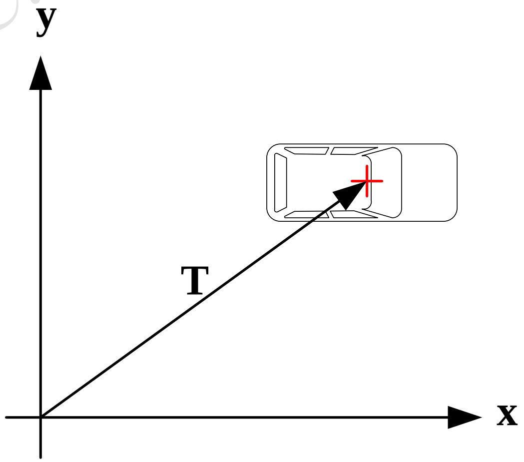
- Example: An object, p, is translate by T1, then scaled by S, then rotated by R, and translate back by $(\mathbf{T}_1)^{-1}$. The new position of this object will be

$$\mathbf{p}^* = \mathbf{T}_1^{-1} \cdot \mathbf{R} \cdot \mathbf{S} \cdot \mathbf{T}_1 \cdot \mathbf{p}$$

2D transformation (example)

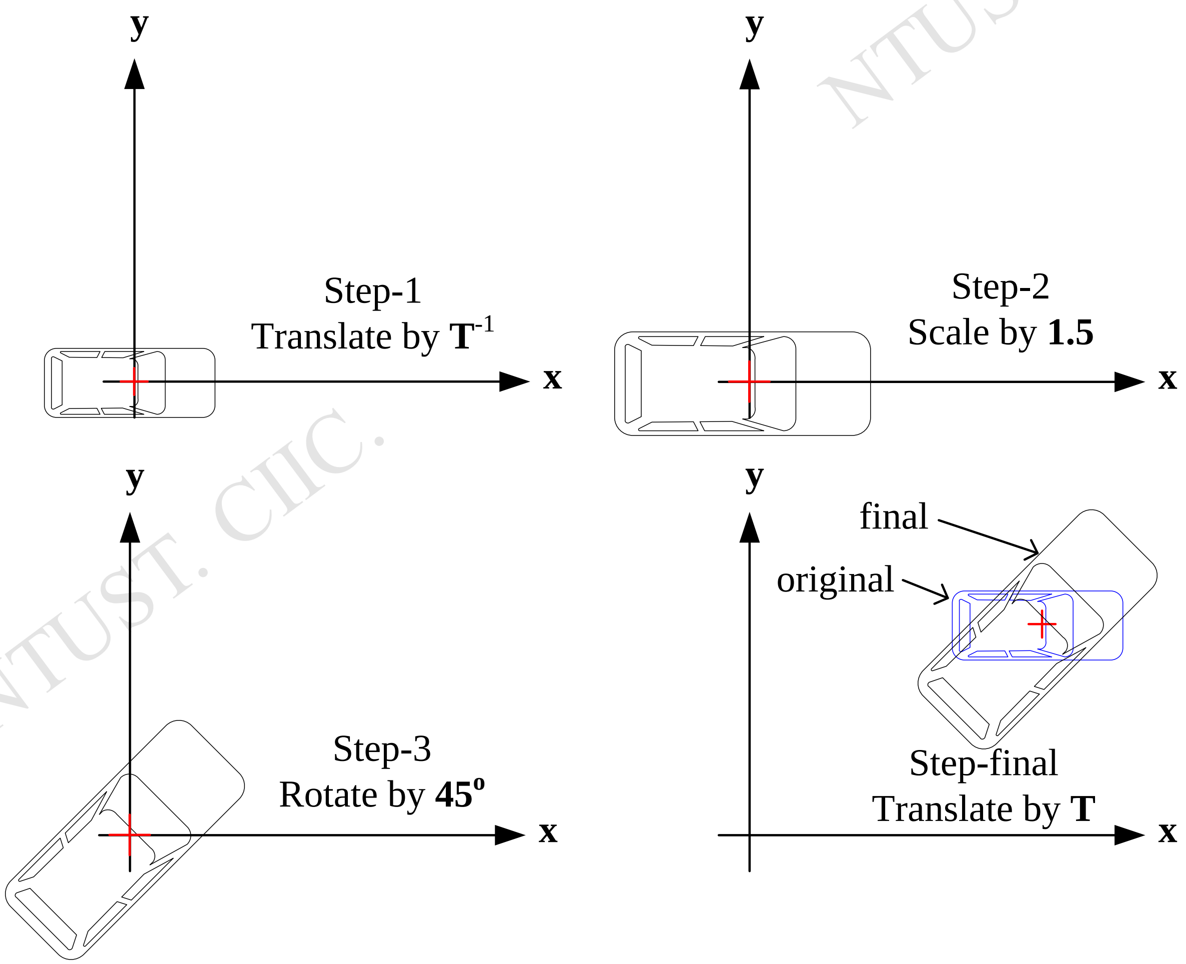
- Rotation or Scaling according to self
  - Example

$$\mathbf{p}^* = \mathbf{T}\cdot\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}\cdot\begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}\cdot\mathbf{T}^{-1}\cdot\mathbf{p} = \mathbf{T}\cdot\begin{bmatrix} \frac{1.5}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1.5}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}\cdot\mathbf{T}^{-1}\cdot\mathbf{p}$$

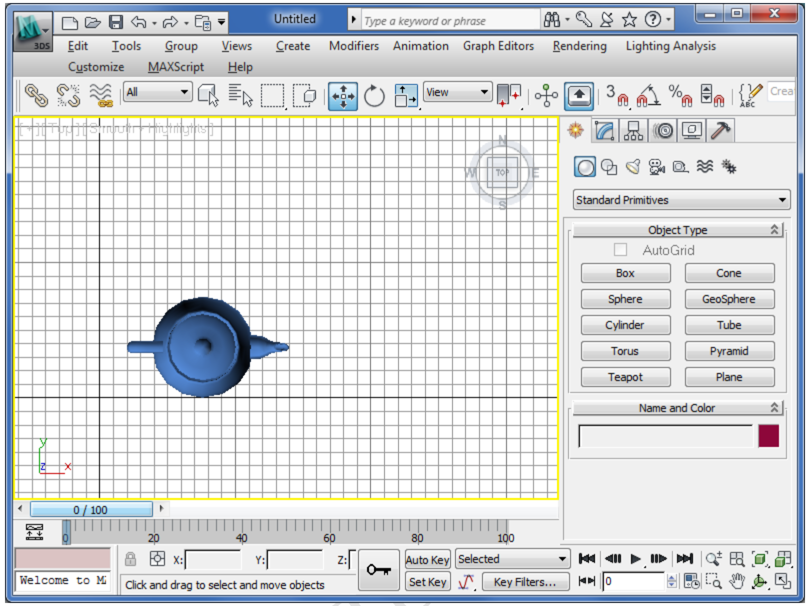


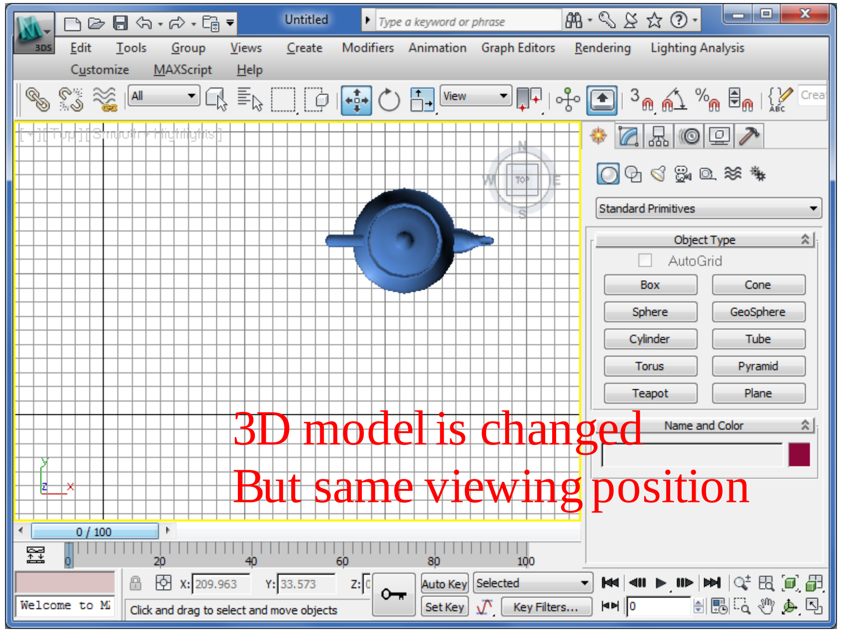
2D transformation (example)

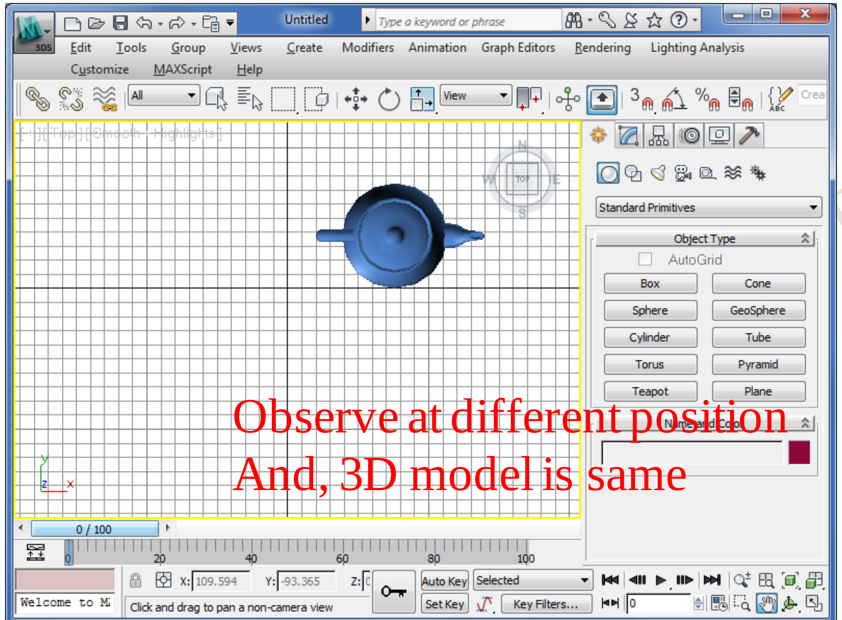
- Rotation or Scaling according to self—cont.



Recall: correct concept-1

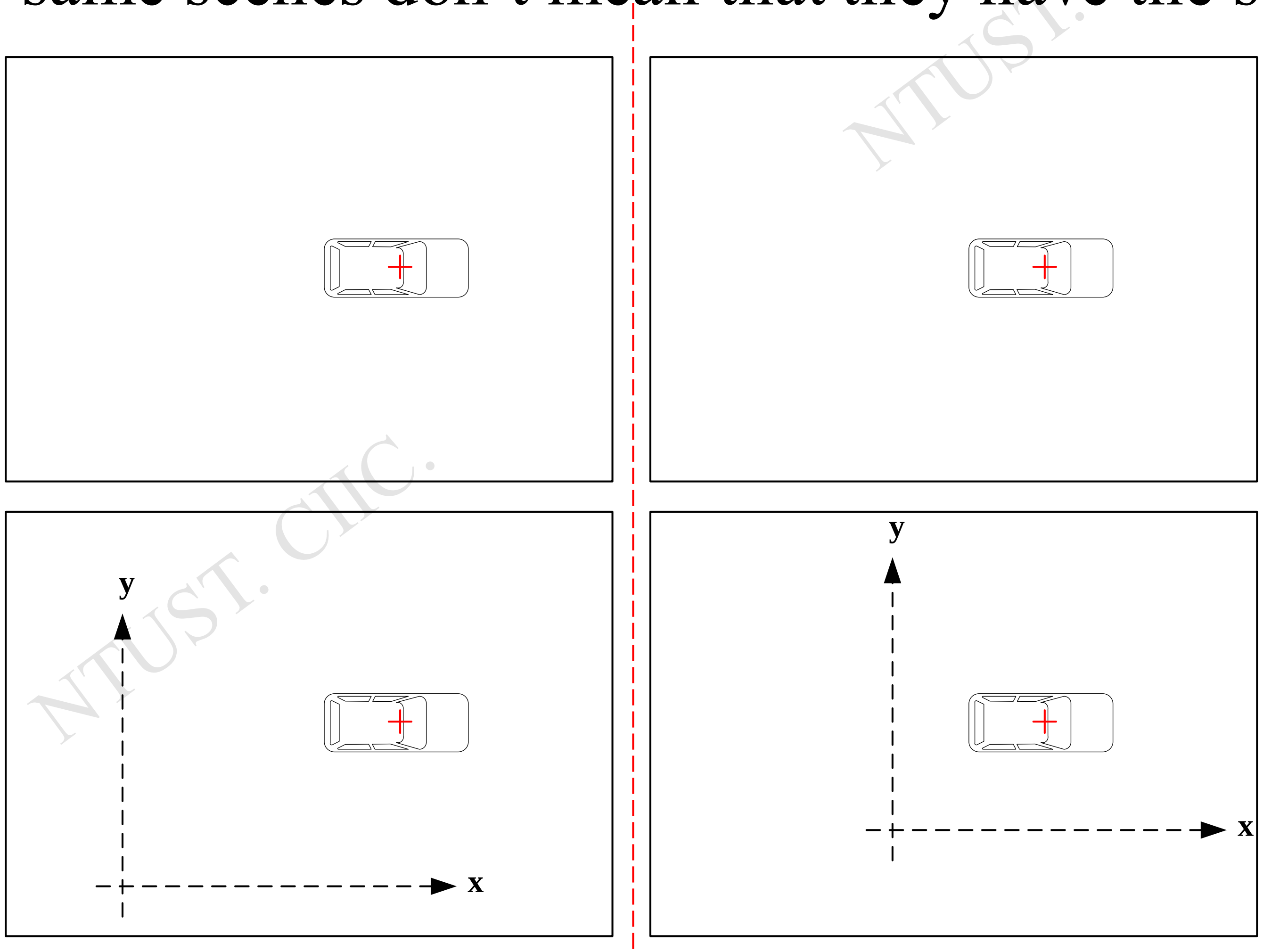






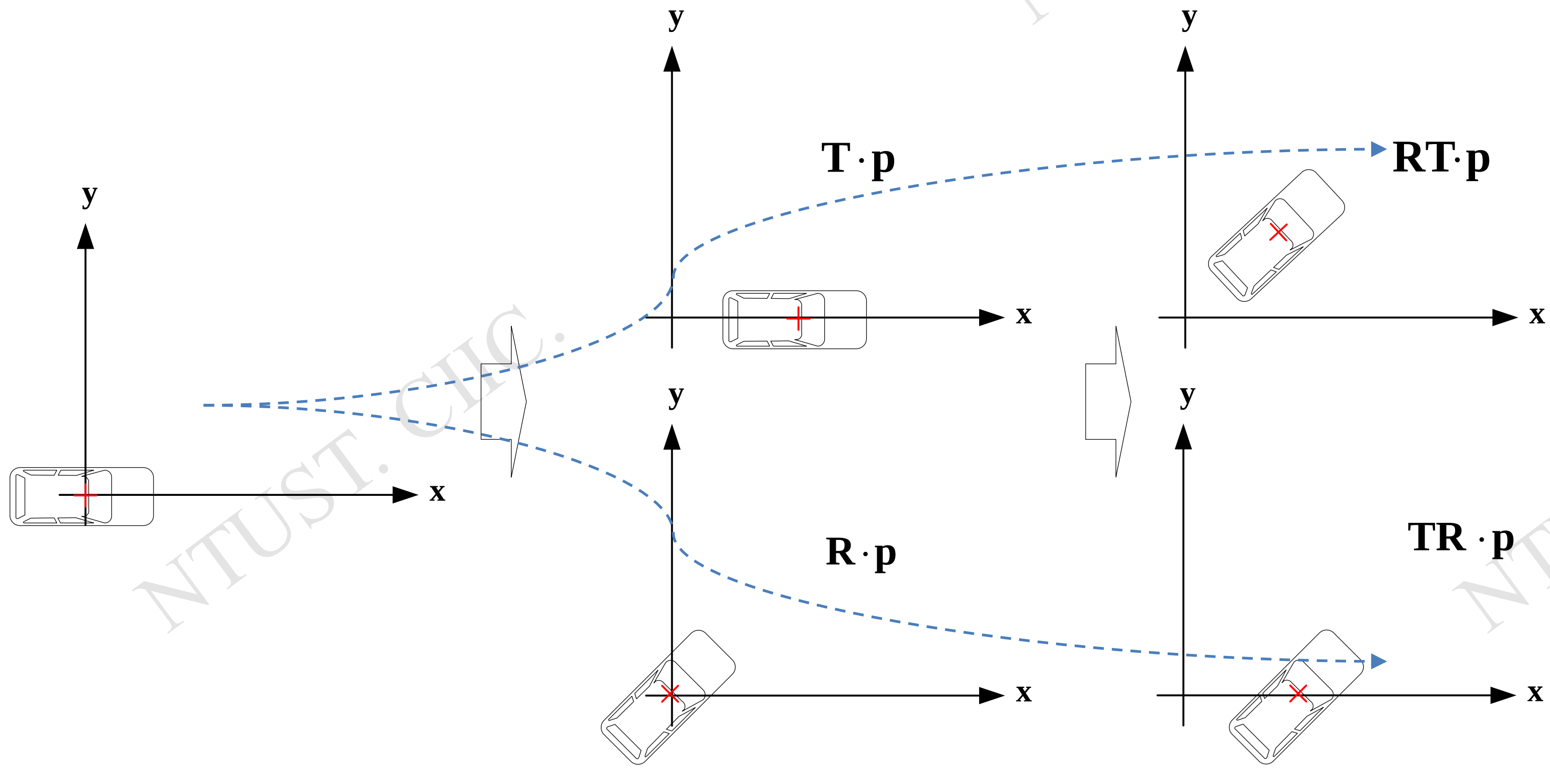
Recall: correct concept-2

- Two exactly same scenes don't mean that they have the same 3D data.



Recall: correct concept-3

- Multiple matrix operations are not exchangeable.
  - Ex: $\mathbf{RT} \cdot \mathbf{p} \neq \mathbf{TR} \cdot \mathbf{p}$



Recall: correct concept-4

- Compare the meaning of two matrix combinations:
  - Rotate, then translate (according to world coordinate (the original))

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

  - Translate, then rotate (according to world coordinate (the original))

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x\cos\theta - t_y\sin\theta \\ \sin\theta & \cos\theta & t_x\sin\theta + t_y\cos\theta \\ 0 & 0 & 1 \end{bmatrix}$$

Short summary of 2D transformation

- Scaling **S**
- Translate **T**
- Rotate **R**
- Combination of several transformations (example)

$$\mathbf{S}=\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{T}=\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R}=\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{STR}=\begin{bmatrix} s_x\cos\theta & -\sin\theta & t_x \\ \sin\theta & s_y\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}=\mathbf{TSR}=\mathbf{TRS}$$

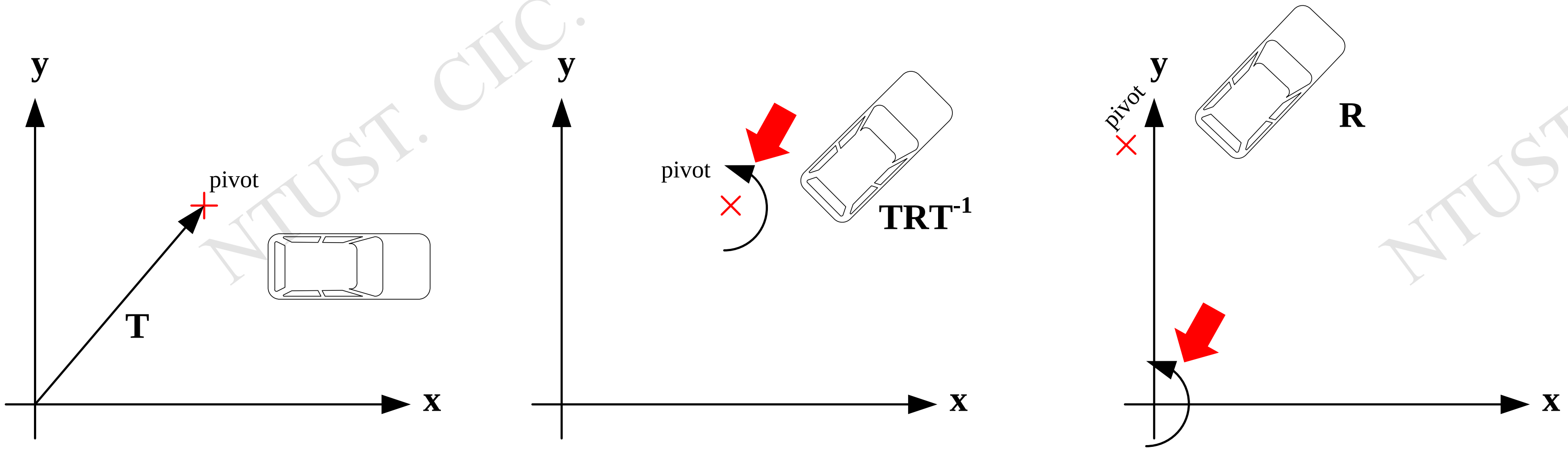
$$\mathbf{RST}=\begin{bmatrix} s_x\cos\theta & -s_y\sin\theta & s_x t_x\cos\theta - s_y t_y\sin\theta \\ s_x\sin\theta & s_y\cos\theta & s_x t_x\sin\theta + s_y t_y\cos\theta) \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{SRT}=\begin{bmatrix} s_x\cos\theta & -s_x\sin\theta & s_x(t_x\cos\theta - t_y\sin\theta) \\ s_y\sin\theta & s_y\cos\theta & s_y(t_x\sin\theta + t_y\cos\theta) \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{RTS}=\begin{bmatrix} s_x\cos\theta & -s_y\sin\theta & t_x\cos\theta - t_y\sin\theta \\ s_x\sin\theta & s_y\cos\theta & t_x\sin\theta + t_y\cos\theta \\ 0 & 0 & 1 \end{bmatrix}$$

2D transformation

- Most 3D software provide the definition of "Pivot" for 2D/3D translation, which is a local coordinate to indicate the shape property, such as centroid, symmetrical shape, center et al.
- For example, a car is rotated by 45° according to the pivot. Thus, **TRT^{-1}** will be performed.



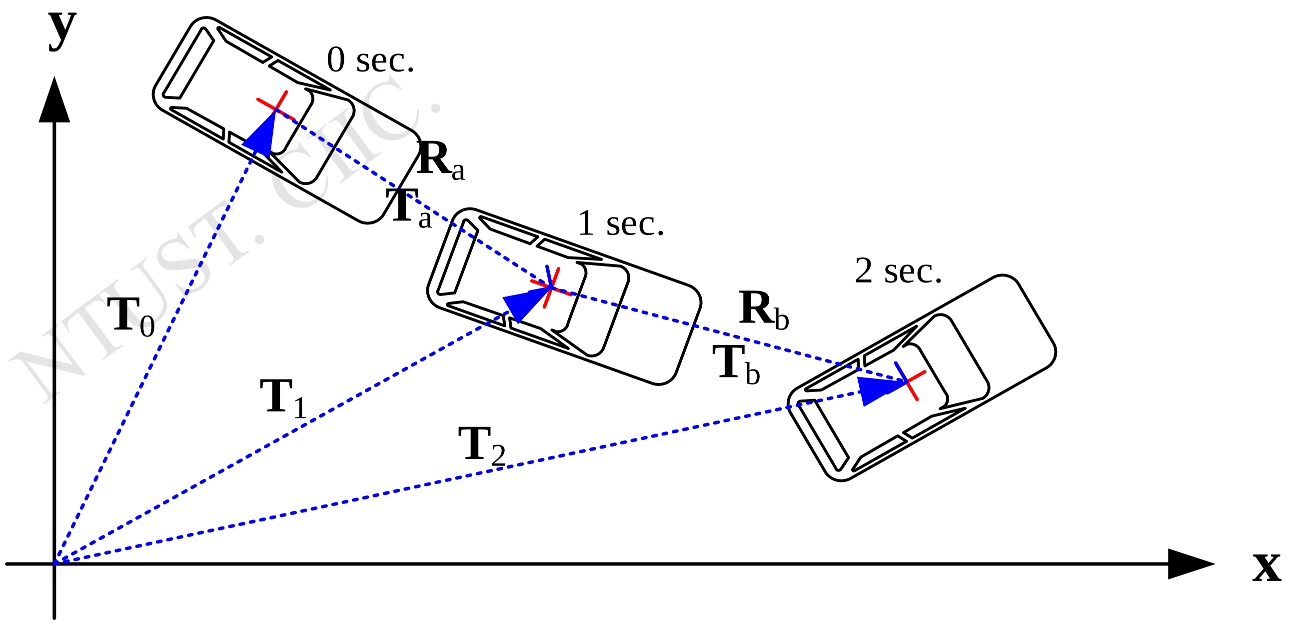
2D transformation (rigid body motion)

- In kinematics, a “rigid body motion” could be the combination of a rotation (**R**) and a translation (**T**). For any transformation, it can be decomposed into **T*R**, as well.
- For example, an object, **p**, rotates with **R** at its centroid, $\mathbf{T}_0$, then translate with $\mathbf{T}_1$, it becomes

$$\mathbf{p}^* = \mathbf{T}_1(\mathbf{T}_0\mathbf{R}\mathbf{T}_0^{-1})\mathbf{p}$$

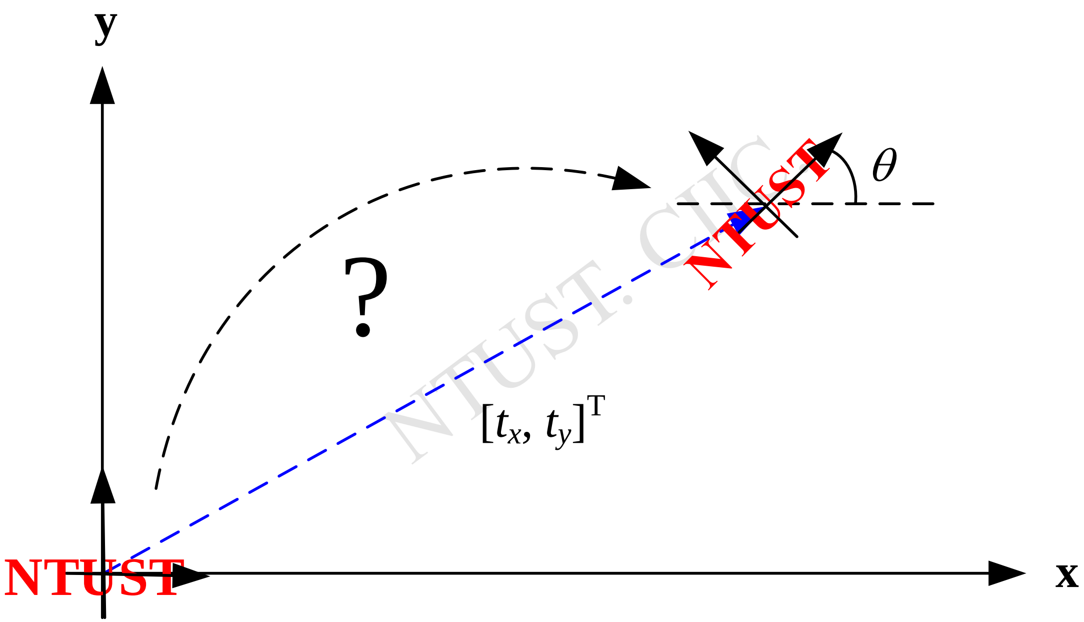
2D transformation (continuous motion)

- For continuing motions,
- The first step will be $\mathbf{p}_1 = \mathbf{T}_a(\mathbf{T}_0\mathbf{R}_a\mathbf{T}_0^{-1})\mathbf{p}_0$
- And the next step is $\mathbf{p}_2 = \mathbf{T}_b(\mathbf{T}_1\mathbf{R}_b\mathbf{T}_1^{-1})\mathbf{p}_1 = \mathbf{T}_b(\mathbf{T}_1\mathbf{R}_b\mathbf{T}_1^{-1})\mathbf{T}_a(\mathbf{T}_0\mathbf{R}_a\mathbf{T}_0^{-1})\mathbf{p}_0$
- where , $\mathbf{T}_a=\mathbf{T}_1-\mathbf{T}_0$, $\mathbf{T}_b=\mathbf{T}_2-\mathbf{T}_1$.



2D transformation (pose estimation)

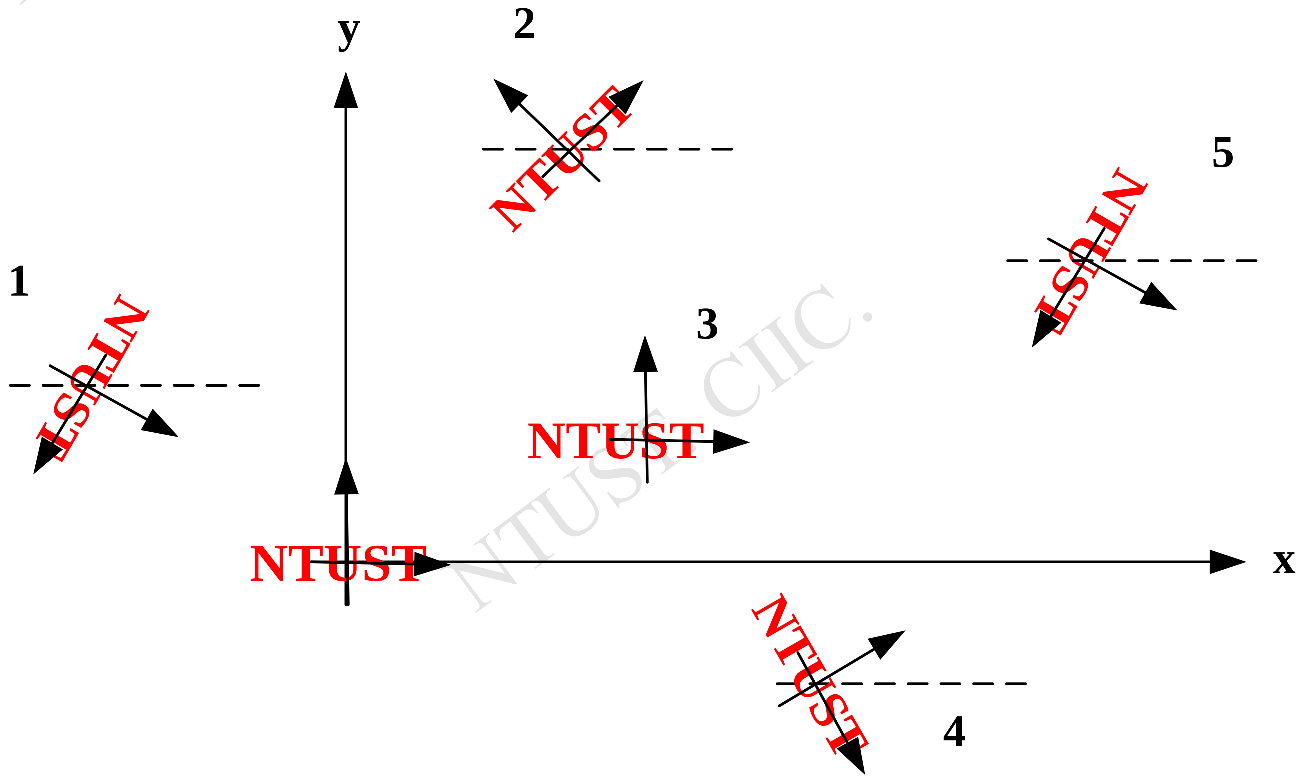
- A "pose" could be the status description for an object corresponding to the original.
- Recall, a composed transformation with a translation after a rotation, thus:



$$\mathbf{p}^* = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}$$

2D transformation (pose estimation)

- How could we estimate the pose? How could we transform an object?



Status 1: (-10, 5)@300°
Status 2: (7, 12)@45°
Status 3: (10, 4)@0°
Status 4: (14, -2)@290°
Status 5: (20, 8)@210°

2D transformation (pose estimation)

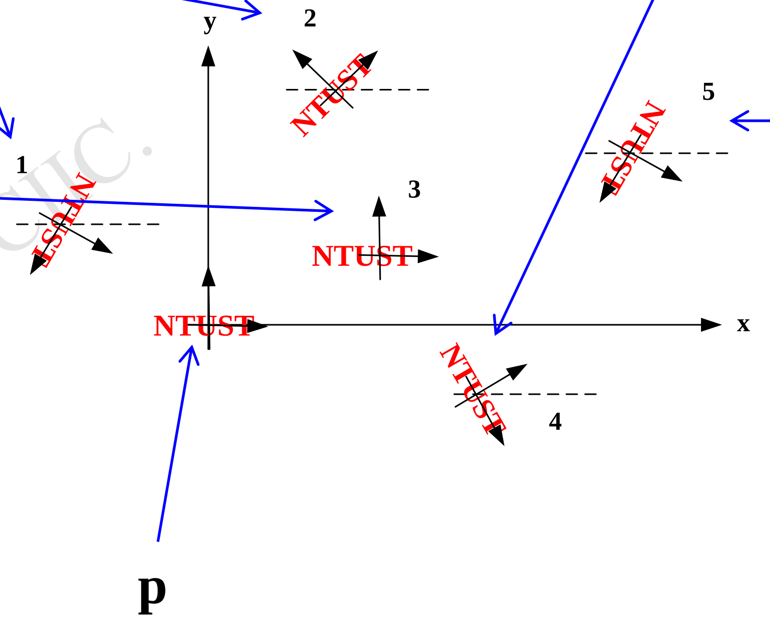
$$\begin{bmatrix} \cos 300^\circ & -\sin 300^\circ & -10 \\ \sin 300^\circ & \cos 300^\circ & 5 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}$$

$$\begin{bmatrix} \cos 45^\circ & -\sin 45^\circ & 7 \\ \sin 45^\circ & \cos 45^\circ & 12 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}$$

$$\begin{bmatrix} \cos 0^\circ & -\sin 0^\circ & 10 \\ \sin 0^\circ & \cos 0^\circ & 4 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}$$

$$\begin{bmatrix} \cos 290^\circ & -\sin 290^\circ & 14 \\ \sin 290^\circ & \cos 290^\circ & -2 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}$$

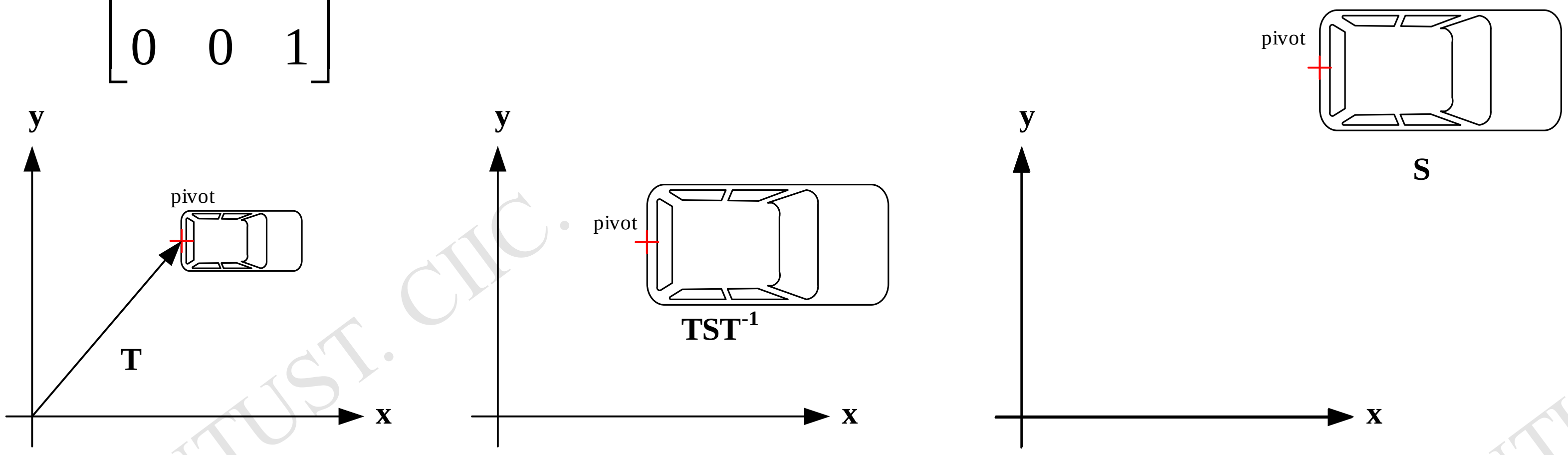
$$\begin{bmatrix} \cos 210^\circ & -\sin 210^\circ & 20 \\ \sin 210^\circ & \cos 210^\circ & 8 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}$$



2D transformation (example-remind)

- An uniform scale transformation is applied

$$\mathbf{S} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



A car

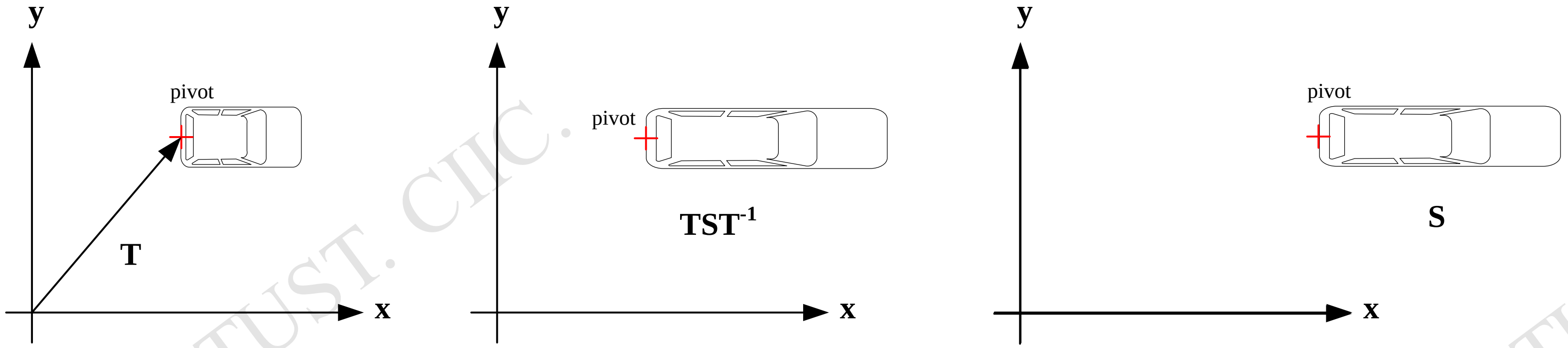
After uniform scale (relative to pivot)

After uniform scale (relative to the original)

2D transformation (example-remind)

- A scale transformation is applied

$$\mathbf{S} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



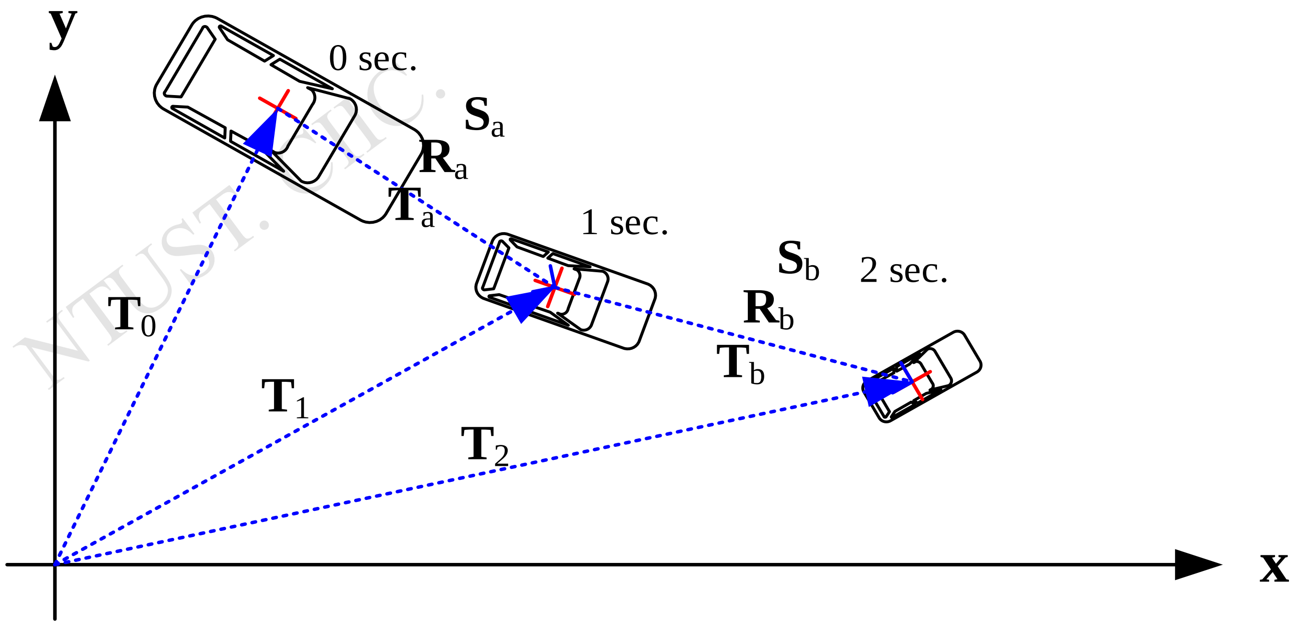
A car

After scale on *X* (relative to pivot)

After scale on *X* (relative to the original)

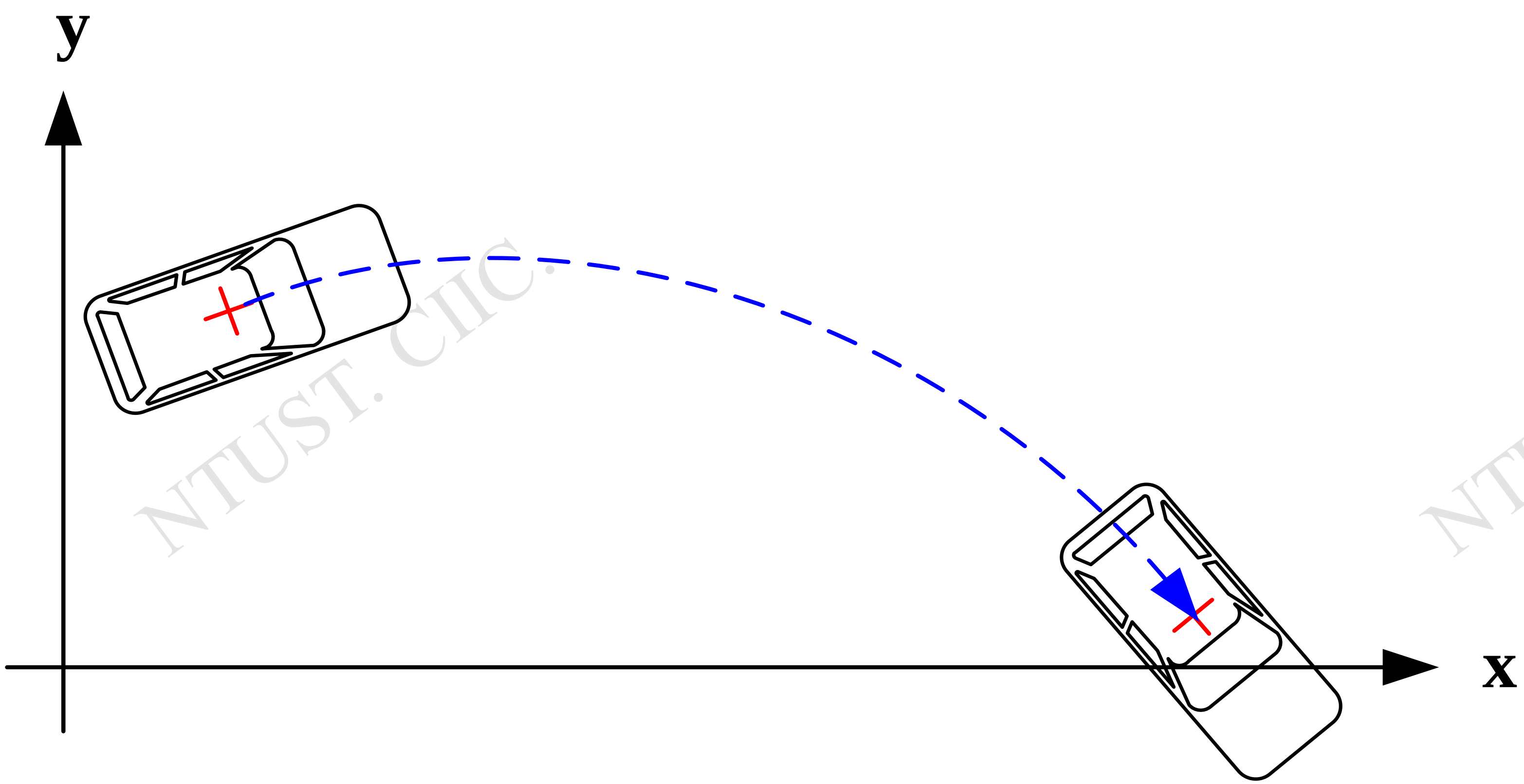
2D transformation (continuously)

- A continuously transformation, for self-scaling and self-rotation, will be…
- for first step: $$\mathbf{p}_1 = \mathbf{T}_a(\mathbf{T}_0\mathbf{S}_a\mathbf{R}_a\mathbf{T}_0^{-1})\mathbf{p}_0$$
- for next step: $$\mathbf{p}_2 = \mathbf{T}_b(\mathbf{T}_1\mathbf{S}_b\mathbf{R}_b\mathbf{T}_1^{-1})\mathbf{T}_a(\mathbf{T}_0\mathbf{S}_a\mathbf{R}_a\mathbf{T}_0^{-1})\mathbf{p}_0$$



2D transformation (continuous motion)

- How we make the "motion" clear
  - By applying matrixes to local coordinate.
  - By updating the pose of the object.

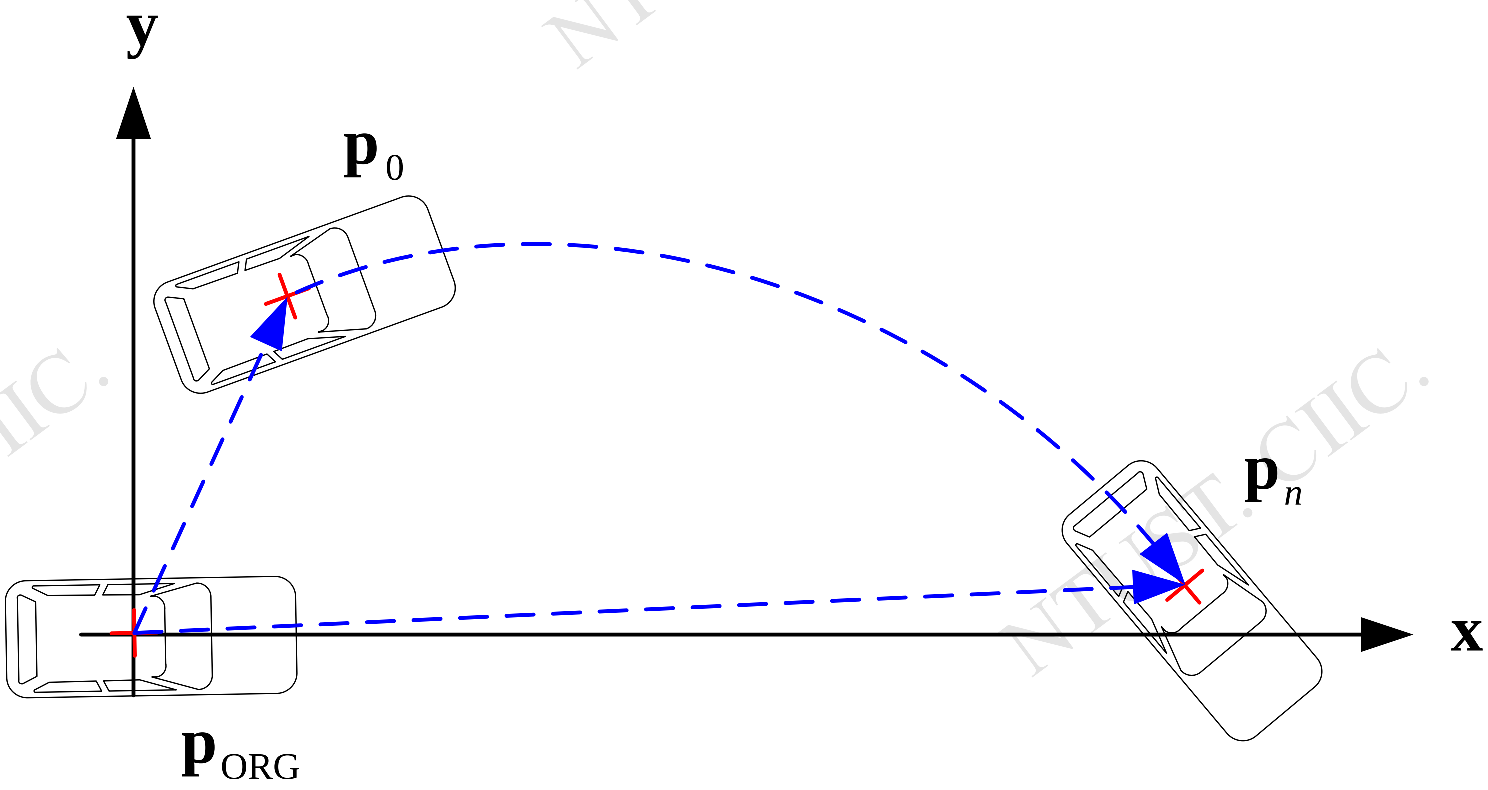


2D transformation (continuous motion)

- Form the other point of view, the status of object can be summarized as:

$$\mathbf{p}_0 = \begin{bmatrix} \cos\theta_0 & -\sin\theta_0 & t_{0x} \\ \sin\theta_0 & \cos\theta_0 & t_{0y} \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}_{\mathrm{ORG}}$$

$$\mathbf{p}_n = \begin{bmatrix} \cos\theta_n & -\sin\theta_n & t_{nx} \\ \sin\theta_n & \cos\theta_n & t_{ny} \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}_{\mathrm{ORG}}$$



2D transformation (continuous motion)

$$\mathbf{p}_n = \prod_{i=0}^{n-1} \mathbf{T}_{\mathrm{E}i}(\mathbf{T}_i \mathbf{R}_{\mathrm{E}i} \mathbf{T}_i^{-1})\mathbf{p}_0$$

$$= \prod_{i=0}^{n-1} \mathbf{T}_{\mathrm{E}i}(\mathbf{T}_i \mathbf{R}_{\mathrm{E}i} \mathbf{T}_i^{-1}) \begin{bmatrix} \cos\theta_0 & -\sin\theta_0 & t_{0x} \\ \sin\theta_0 & \cos\theta_0 & t_{0y} \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}_{\mathrm{ORG}}$$

$$= \begin{bmatrix} \cos\theta_n & -\sin\theta_n & t_{nx} \\ \sin\theta_n & \cos\theta_n & t_{ny} \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}_{\mathrm{ORG}}$$

$$\prod_{i=0}^{n-1} \mathbf{T}_{\mathrm{E}i}(\mathbf{T}_i \mathbf{R}_{\mathrm{E}i} \mathbf{T}_i^{-1}) \begin{bmatrix} \cos\theta_0 & -\sin\theta_0 & t_{0x} \\ \sin\theta_0 & \cos\theta_0 & t_{0y} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_n & -\sin\theta_n & t_{nx} \\ \sin\theta_n & \cos\theta_n & t_{ny} \\ 0 & 0 & 1 \end{bmatrix}$$

$$\prod_{i=0}^{n-1} \mathbf{T}_{\mathrm{E}i}(\mathbf{T}_i \mathbf{R}_{\mathrm{E}i} \mathbf{T}_i^{-1}) = \begin{bmatrix} \cos\theta_n & -\sin\theta_n & t_{nx} \\ \sin\theta_n & \cos\theta_n & t_{ny} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_0 & -\sin\theta_0 & t_{0x} \\ \sin\theta_0 & \cos\theta_0 & t_{0y} \\ 0 & 0 & 1 \end{bmatrix}^{-1}$$

2D transformation (continuous motion)

$$\prod_{i=0}^{n-1}\mathbf{T}_{\mathrm{E}i}(\mathbf{T}_i\mathbf{R}_{\mathrm{E}i}\mathbf{T}_i^{-1})=\begin{bmatrix}\cos\theta_n & -\sin\theta_n & t_{nx}\\ \sin\theta_n & \cos\theta_n & t_{ny}\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\cos\theta_0 & -\sin\theta_0 & 0\\ \sin\theta_0 & \cos\theta_0 & 0\\ 0 & 0 & 1\end{bmatrix}^{-1}\begin{bmatrix}1 & 0 & t_{0x}\\ 0 & 1 & t_{0y}\\ 0 & 0 & 1\end{bmatrix}^{-1}$$

$$\prod_{i=0}^{n-1}\mathbf{T}_{\mathrm{E}i}(\mathbf{T}_i\mathbf{R}_{\mathrm{E}i}\mathbf{T}_i^{-1})=\begin{bmatrix}\cos\theta_n & -\sin\theta_n & t_{nx}\\ \sin\theta_n & \cos\theta_n & t_{ny}\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\cos(-\theta_0) & -\sin(-\theta_0) & -(t_{0x}\cos(-\theta_0)-t_{0y}\sin(-\theta_0))\\ \sin(-\theta_0) & \cos(-\theta_0) & -(t_{0x}\sin(-\theta_0)+t_{0y}\cos(-\theta_0))\\ 0 & 0 & 1\end{bmatrix}$$

3D transformation (extend 2D matrix)

Translation：
$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scaling：
$$\begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotate θ along **z** axis：
$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotate θ along **y** axis：
$$\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotate θ along **x** axis：
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3D transformation: Affine mapping

- Similar to 2D, 3D transformation is one kind of “affine mapping”, which is described as a 4x4 matrix (in homogenous coordinate)

$$\mathbf{p}^* = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{p}$$

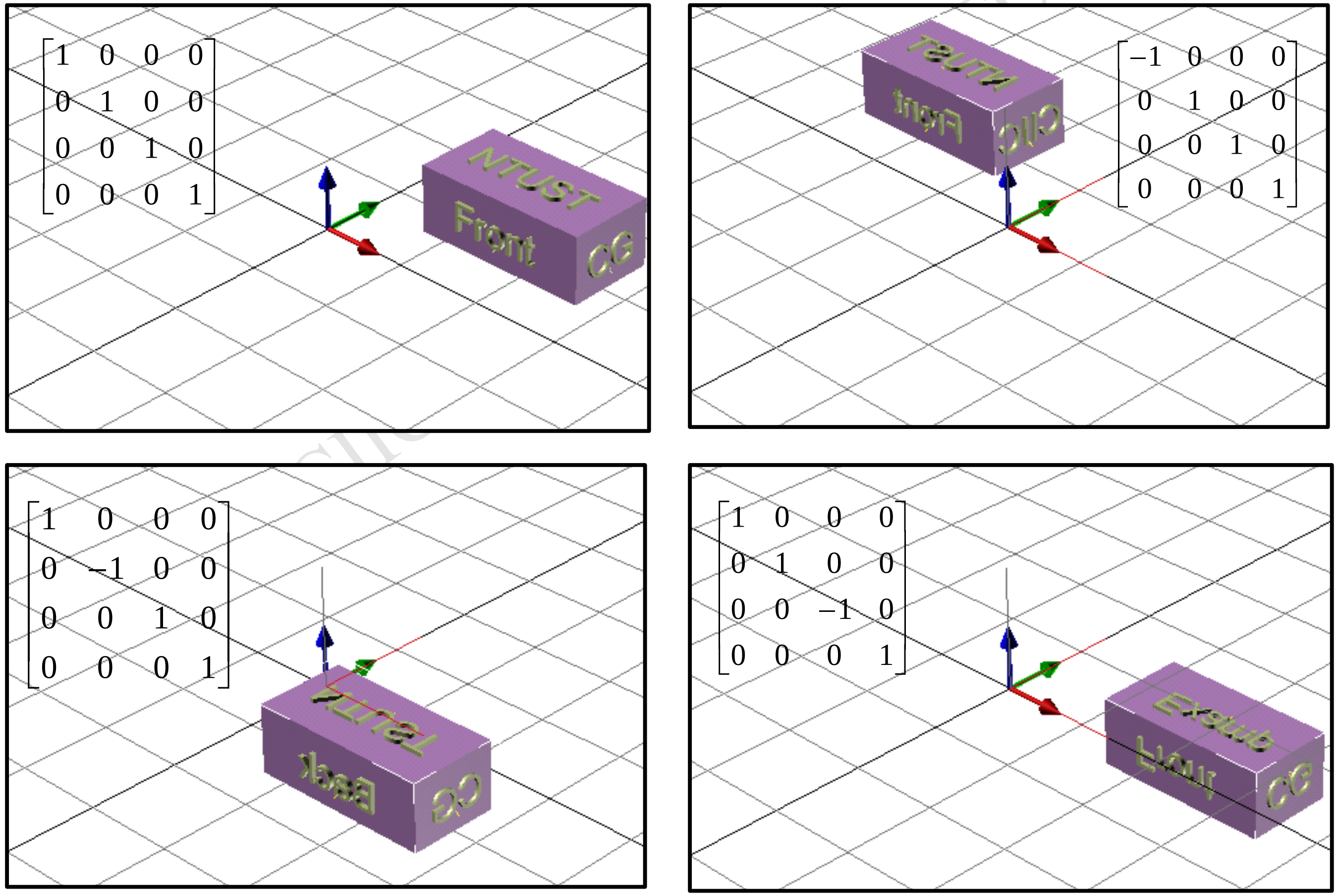
3D transformation: Affine mapping

矩陣	性質
$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$	不做任何改變
$\begin{bmatrix}-1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$	對 x 軸方向鏡射(以 yz 軸所構成平面對稱)
$\begin{bmatrix}1&0&0&0\\0&-1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$	對 y 軸方向鏡射

$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&-1&0\\0&0&0&1\end{bmatrix}$	對 z 軸方向鏡射
$\begin{bmatrix}-1&0&0&0\\0&-1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$	先對 x 軸方向再對 y 軸方向進行鏡射，或先對 y 軸方向再對 x 軸方向進行鏡射
$\begin{bmatrix}-1&0&0&0\\0&1&0&0\\0&0&-1&0\\0&0&0&1\end{bmatrix}$	先對 x 軸方向再對 z 軸方向進行鏡射，或先對 z 軸方向再對 x 軸方向進行鏡射
$\begin{bmatrix}1&0&0&0\\0&-1&0&0\\0&0&-1&0\\0&0&0&1\end{bmatrix}$	先對 y 軸方向再對 z 軸方向進行鏡射，或先對 z 軸方向再對 y 軸方向進行鏡射

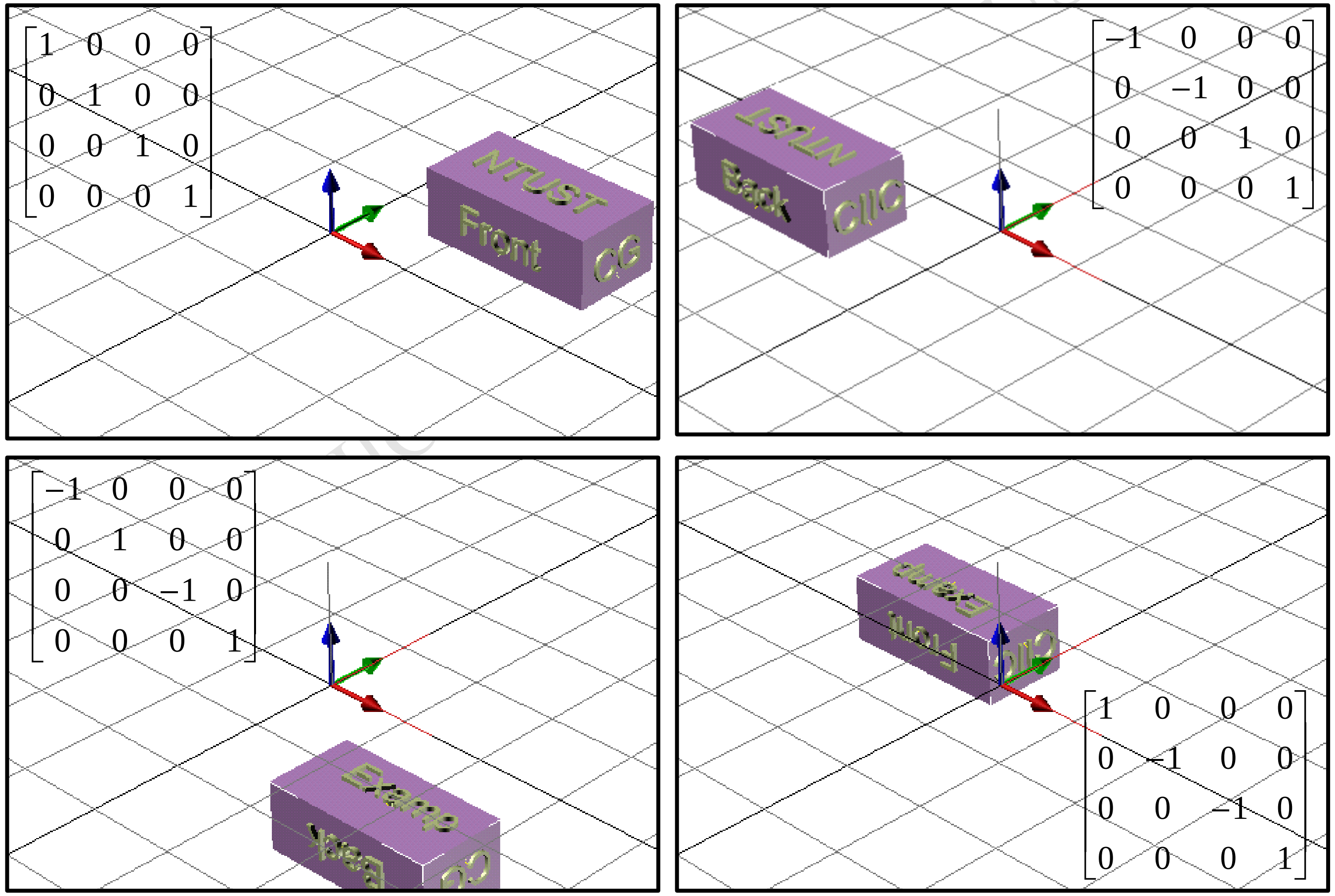
3D transformation: Affine mapping

- A 3D “Mirror” operation—example



3D transformation: Affine mapping

- A 3D “Mirror” operation—example cont.



3D transformation: Affine mapping

- The general form consists of a 3 x 3 mapping operation and a shift operation.
- In a orthogonal coordinate, the form could be decomposed into a translation after a rotation.

$$\mathbf{p}^* = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}\mathbf{p} = \begin{bmatrix} 1 & 0 & 0 & a_{14} \\ 0 & 1 & 0 & a_{24} \\ 0 & 0 & 1 & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\mathbf{p}$$

3D transformation: Affine mapping

- Thus, the 3D transformation will be

$$\begin{bmatrix} x_p^* \\ y_p^* \\ z_p^* \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} + \begin{bmatrix} a_{14} \\ a_{24} \\ a_{34} \end{bmatrix}$$

- It can be written as

$$\begin{bmatrix} x_p^* \\ y_p^* \\ z_p^* \end{bmatrix} = x_p \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix} + y_p \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix} + z_p \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix} + 1 \begin{bmatrix} a_{14} \\ a_{24} \\ a_{34} \end{bmatrix}$$

- Where, new vectors are defined:

$$\mathbf{a} = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} a_{14} \\ a_{24} \\ a_{34} \end{bmatrix}$$

3D transformation: Affine mapping

- The result after 3D transformation becomes: the linear combination of four vectors with corresponding factor, i.e. the same with previous vector (vertex).

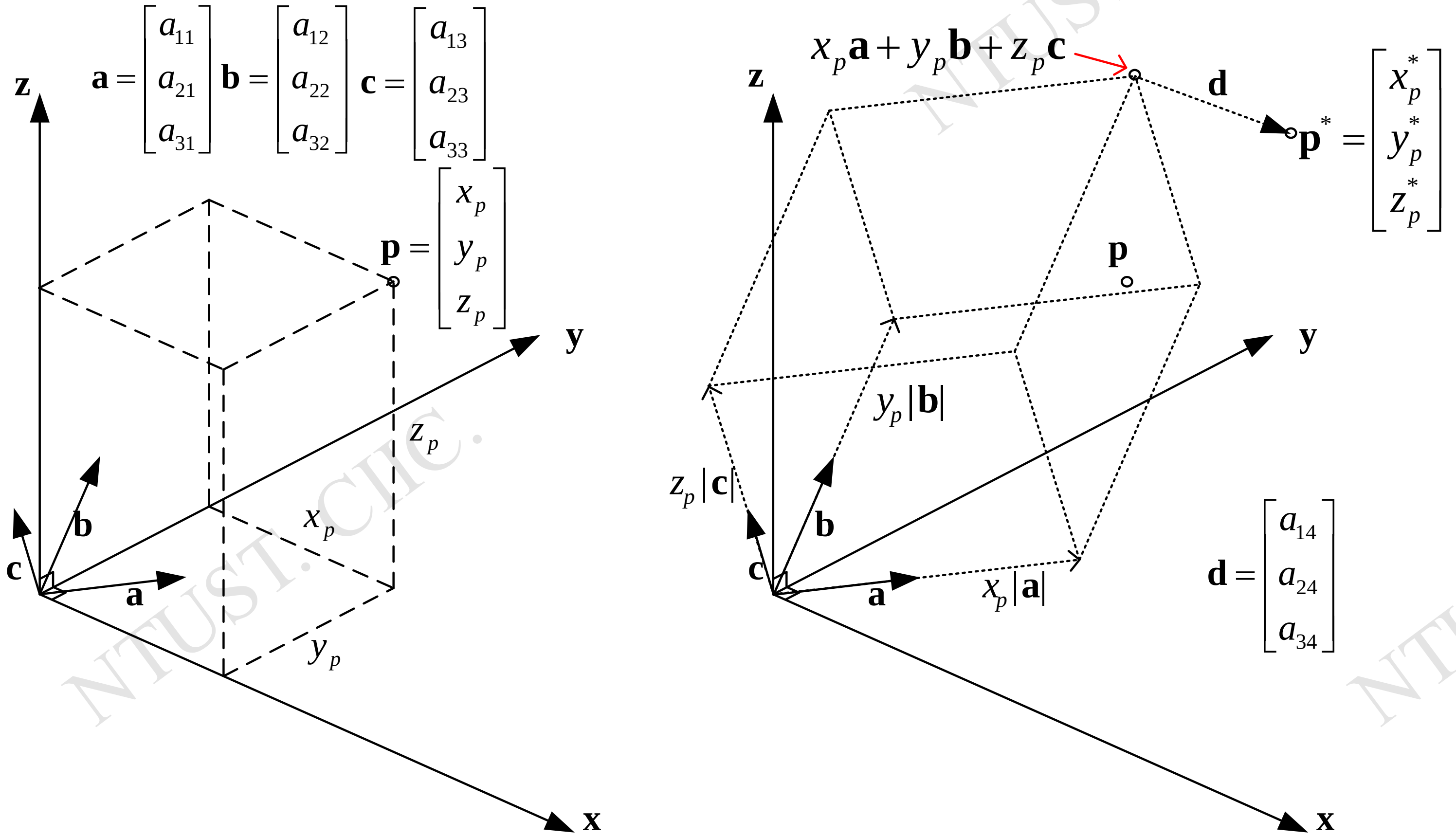
$$\begin{bmatrix} x_p^* \\ y_p^* \\ z_p^* \end{bmatrix} = x_p \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix} + y_p \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix} + z_p \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix} + 1 \begin{bmatrix} a_{14} \\ a_{24} \\ a_{34} \end{bmatrix}$$

- In other words,

$$\begin{bmatrix} x_p^* \\ y_p^* \\ z_p^* \end{bmatrix} = x_p \mathbf{a} + y_p \mathbf{b} + z_p \mathbf{c} + 1\mathbf{d}$$

3D transformation: Affine mapping

- Visualization of this operation:



3D transformation: 3D translation

- 3D translation is to add a vector the all vertexes of an object.

$$\begin{bmatrix} x_p^* \\ y_p^* \\ z_p^* \end{bmatrix} = \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} x_p + t_x \\ y_p + t_y \\ z_p + t_z \end{bmatrix}$$

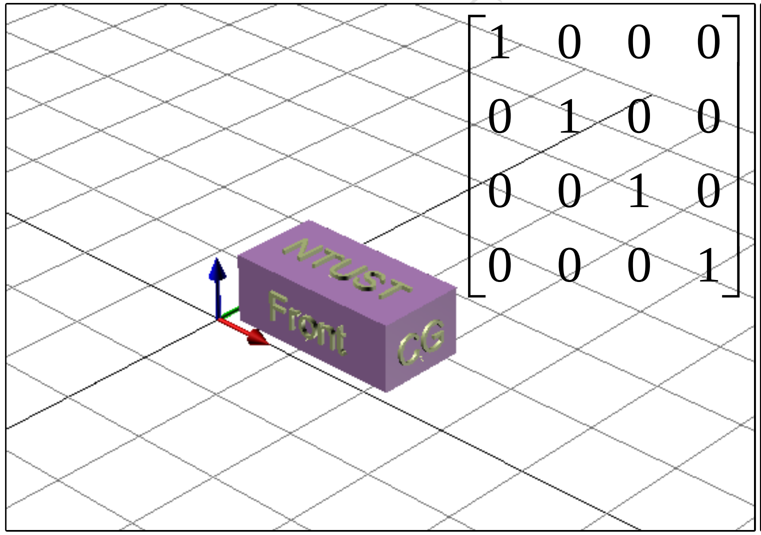
- For convenience, the operation can be stored as a 4x4 matrix

$$\begin{bmatrix} x_p^* \\ y_p^* \\ z_p^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}$$

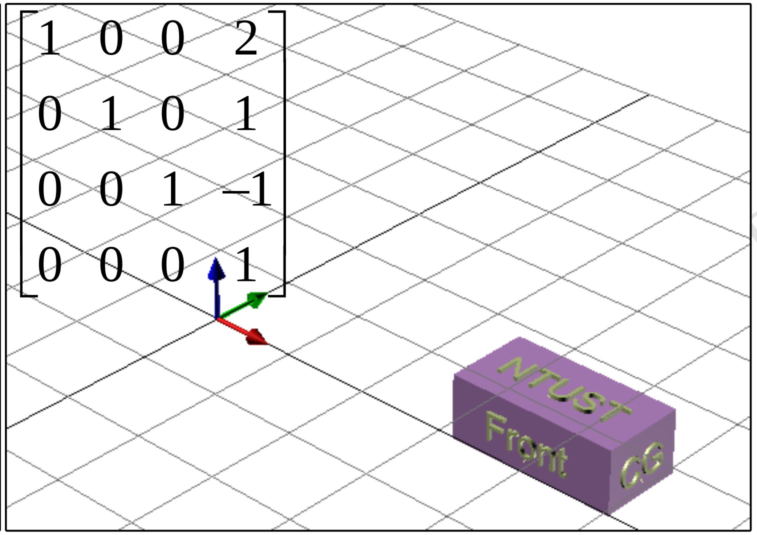
3D transformation: 3D translation

- Example: an object is translated with [2, 1, -1].
- The reference coordinate is the same either to the original or to pivot.

$$\begin{bmatrix} x_p^* \\ y_p^* \\ z_p^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}$$



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



$$\begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3D transformation: 3D scaling

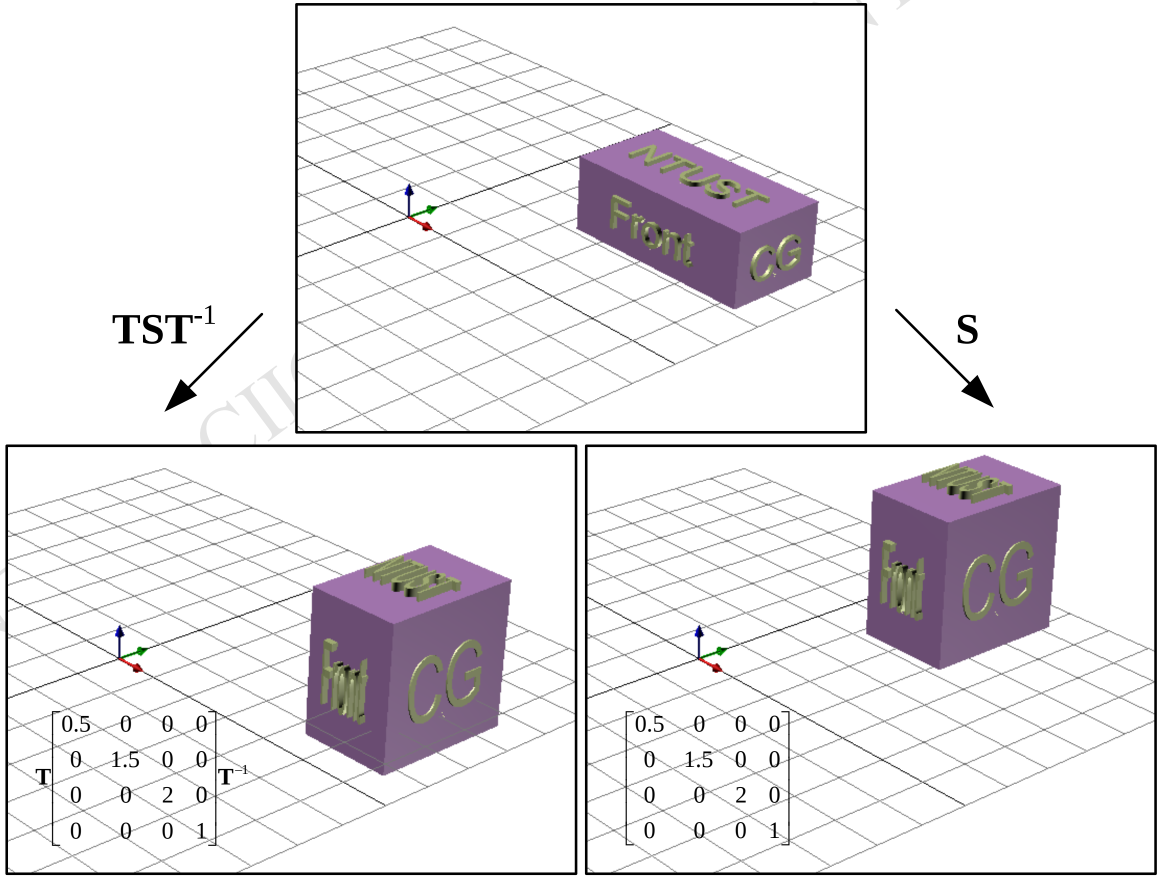
- A 3D scaling operation, in matrix form is:

$$\begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- which is referred to the world coordinate (the original).

3D transformation: 3D scaling

- Example: scaling according to either pivot or world coordinate

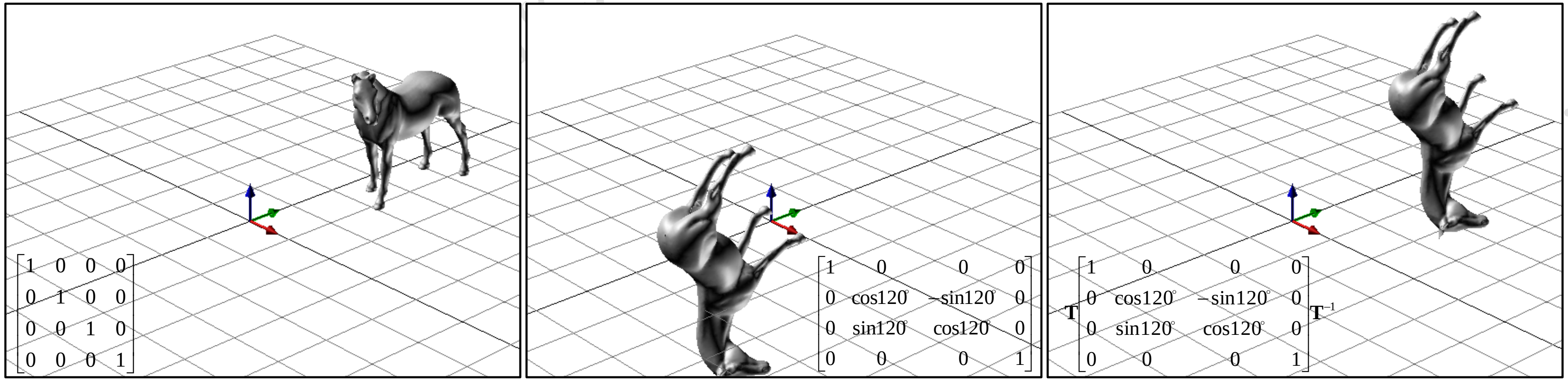


3D transformation—rotation along axis

- Rotation along x, y, or z axis will be

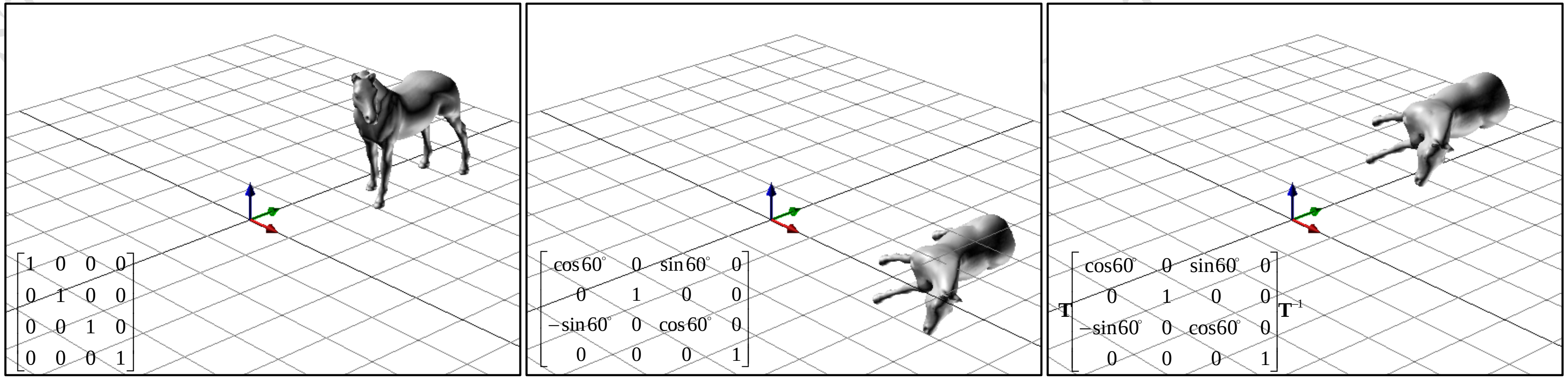
$$\text{rotation along } \mathbf{z} \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{rotation along } \mathbf{y} \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{rotation along } \mathbf{x} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

example: rotation along **x**, and on pivot

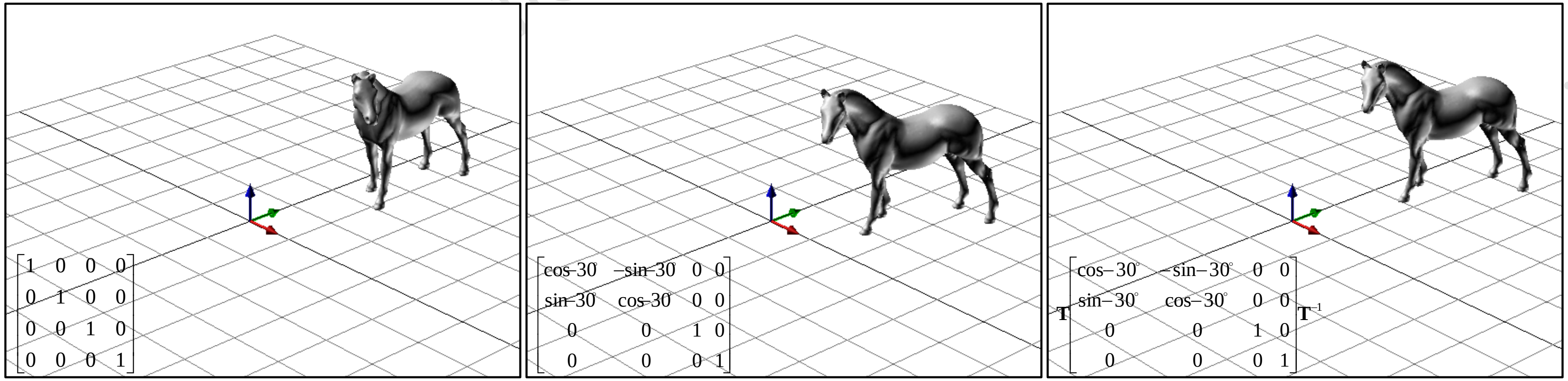


3D transformation—rotation along axis

example: rotation along **y**, and on pivot

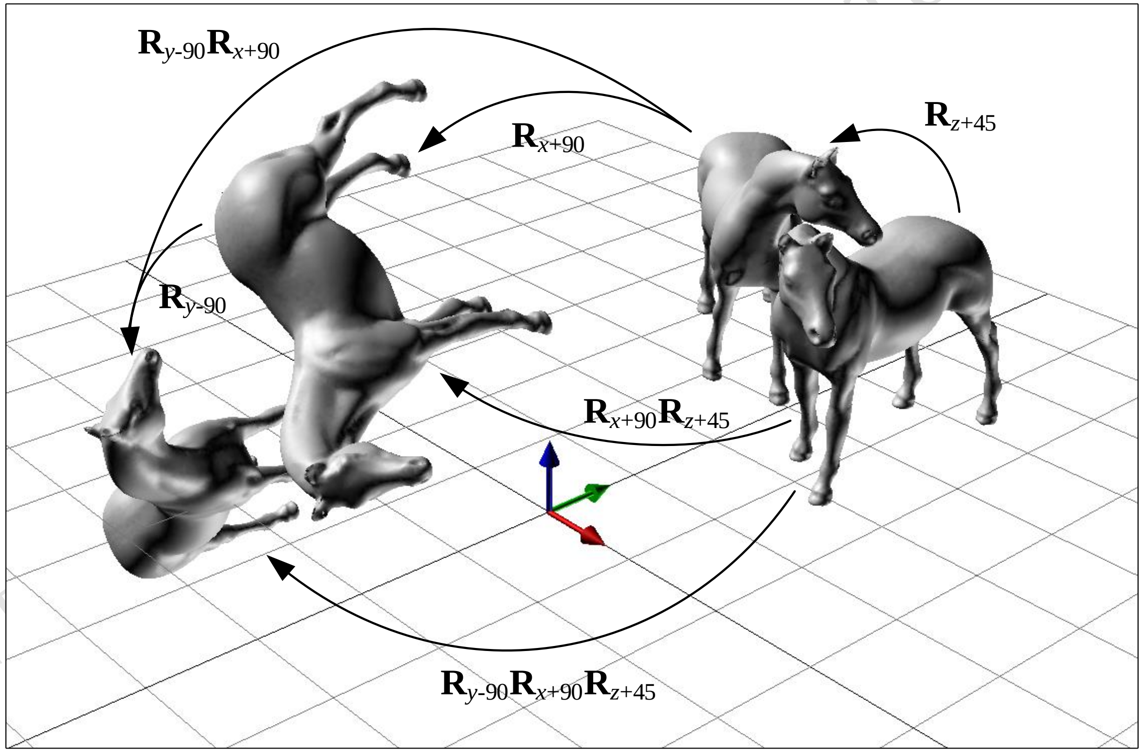


example: rotation along **z**, and on pivot

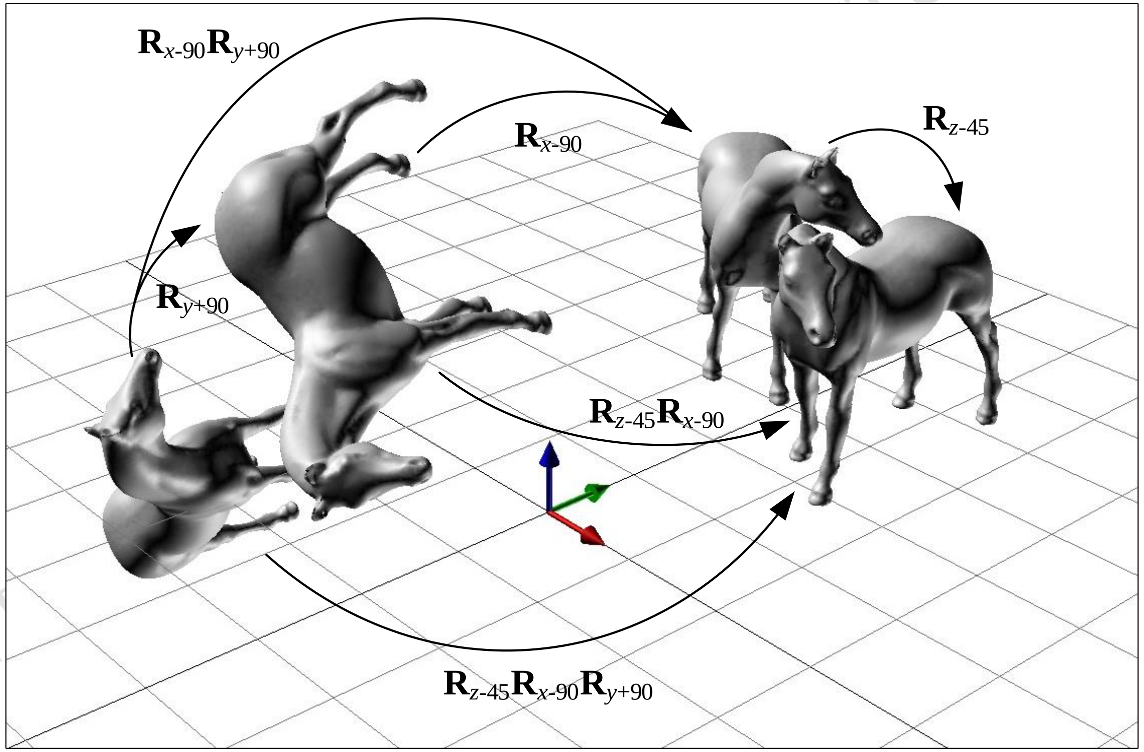


3D transformation—example

$$\mathbf{R}_{z-45}\mathbf{R}_{x-90}\mathbf{R}_{y+90}\mathbf{R}_{y-90}\mathbf{R}_{x+90}\mathbf{R}_{z+45} = \mathbf{I}$$

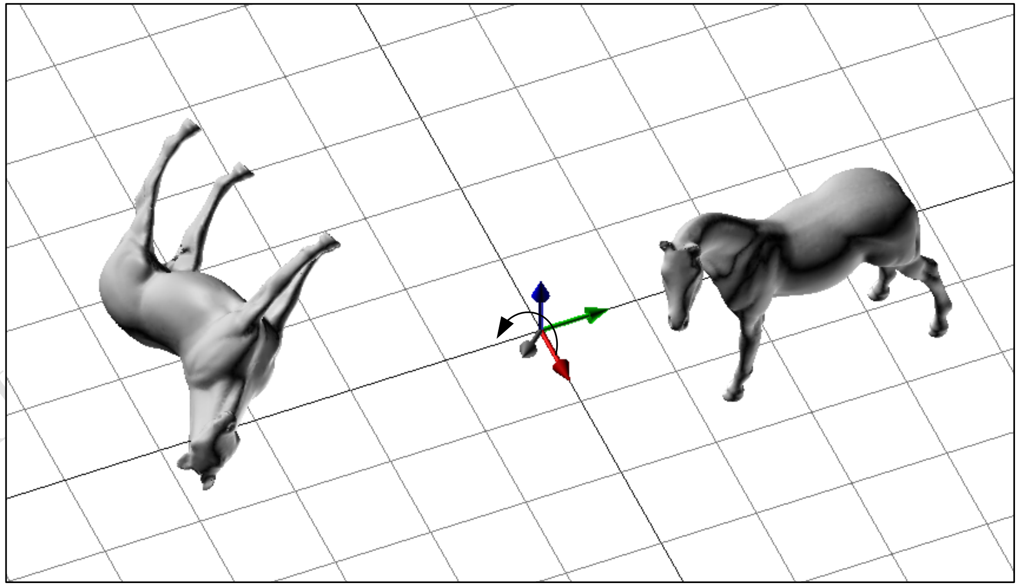


3D transformation—example



3D rotation according to any direction

$$\mathbf{R}_\theta = \begin{bmatrix} (1-a^2)\cos\theta + a^2 & -ab\cos\theta - c\sin\theta + ab & -ac\cos\theta + b\sin\theta + ac & 0 \\ -ab\cos\theta + c\sin\theta + ab & (1-b^2)\cos\theta + b^2 & -bc\cos\theta - a\sin\theta + bc & 0 \\ -ac\cos\theta - b\sin\theta + ac & -bc\cos\theta + a\sin\theta + bc & (1-c^2)\cos\theta + c^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



3D rotation according to any direction

- Proof: assume rotating axes is an unit vector $[a, b, c]^T$

$$a^2 + b^2 + c^2 = 1$$

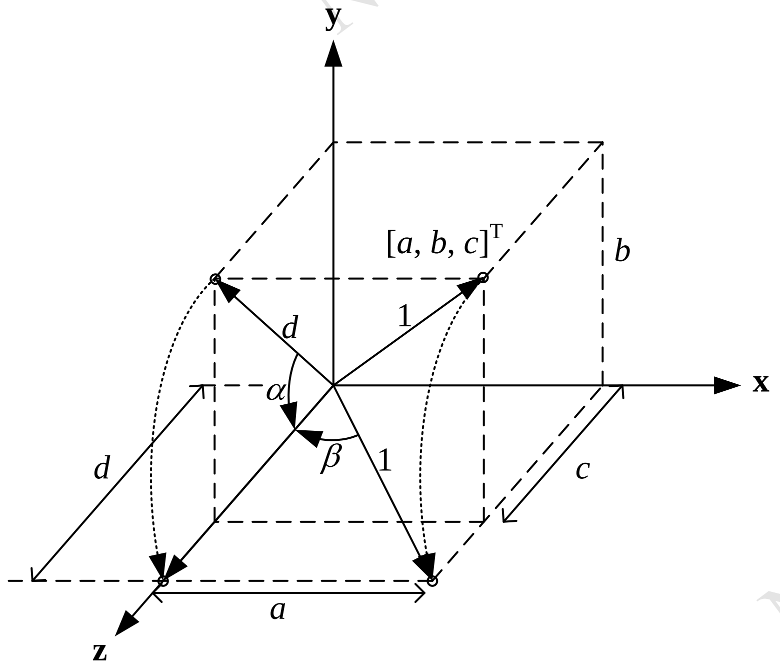
$$d = \sqrt{b^2 + c^2}$$

$$\cos\alpha = \frac{c}{d}$$

$$\sin\alpha = \frac{b}{d}$$

$$\cos\beta = d$$

$$\sin\beta = a$$



$$\mathbf{R}_\theta = \mathbf{R}_x(-\alpha)\mathbf{R}_y(\beta)\mathbf{R}_z(\theta)\mathbf{R}_y(-\beta)\mathbf{R}_x(\alpha)$$

3D rotation according to any direction

$$\mathbf{R}_\theta = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos-\alpha & -\sin-\alpha & 0 \\ 0 & \sin-\alpha & \cos-\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

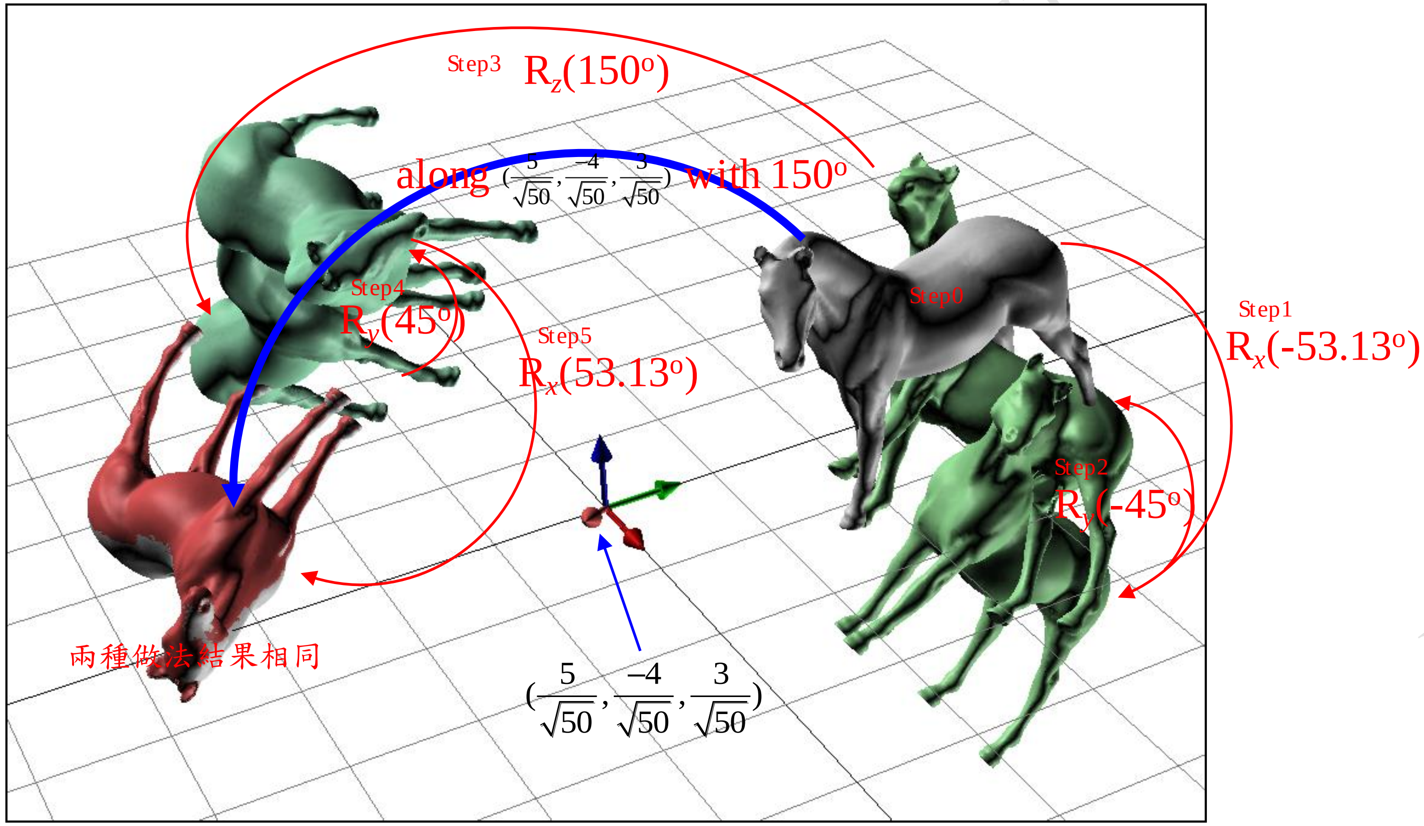
$$\cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos-\beta & 0 & \sin-\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin-\beta & 0 & \cos-\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Finally,

$$\mathbf{R}_\theta = \begin{bmatrix} (1-a^2)\cos\theta + a^2 & -ab\cos\theta - c\sin\theta + ab & -ac\cos\theta + b\sin\theta + ac & 0 \\ -ab\cos\theta + c\sin\theta + ab & (1-b^2)\cos\theta + b^2 & -bc\cos\theta - a\sin\theta + bc & 0 \\ -ac\cos\theta - b\sin\theta + ac & -bc\cos\theta + a\sin\theta + bc & (1-c^2)\cos\theta + c^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3D rotation according to any direction

- Example: 150° rotation according to $(\frac{5}{\sqrt{50}}, \frac{-4}{\sqrt{50}}, \frac{3}{\sqrt{50}})$



3D rotation by Quaternion

- Rotation matrix to Quaternion

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow [p \quad q \quad r \quad w]$$

$$w = \frac{\sqrt{1 + a_{11} + a_{22} + a_{33}}}{2}$$

$$p = \frac{a_{32} - a_{23}}{4w}$$

$$q = \frac{a_{13} - a_{31}}{4w}$$

$$r = \frac{a_{21} - a_{12}}{4w}$$

3D rotation by Quaternion

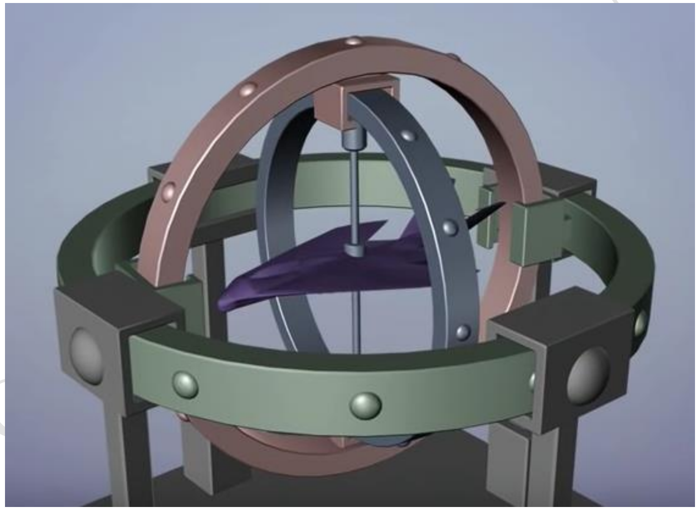
- Quaternion to rotation matrix

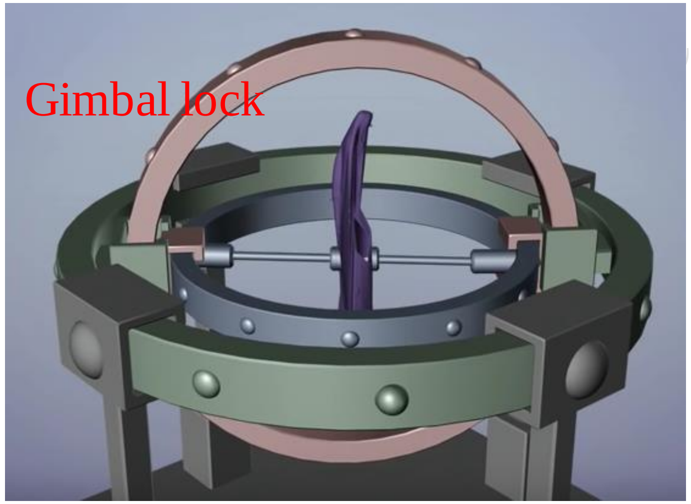
$$[p \quad q \quad r \quad w] \Rightarrow \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$a_{11} = 1 - 2q^2 - 2r^2$$
$$a_{12} = 2pq - 2rw$$
$$a_{13} = 2pr + 2qw$$
$$a_{21} = 2pq + 2rw$$
$$a_{22} = 1 - 2p^2 - 2r^2$$
$$a_{23} = 2qr - 2pw$$
$$a_{31} = 2pr - 2qw$$
$$a_{32} = 2qr + pw$$
$$a_{33} = 1 - 2p^2 - 2q^2$$

Gimbal lock

- The loss of one degree of freedom of rotation





https://www.youtube.com/watch?v=zc8b2Jo7mno
https://www.cinema5d.com/5-affordable-gimbal-stabilizers-cheaper-dji-ronin/

Performance issue

- Compare with the matrix operations

Operation case1: matrix44 x vector

$$\mathbf{p}^* = \mathbf{M}^m{}_{4\times4}[\mathbf{M}^{m-1}{}_{4\times4}\ldots(\mathbf{M}^1{}_{4\times4}\mathbf{p}_{4\times n})]$$

Operation case2: matrix44 x matrix44, and matrix44 x vector

$$\mathbf{p}^* = [\prod_{i=1}^{m}(\mathbf{M}^i{}_{4\times4})]\mathbf{p}_{4\times n}$$

case	Matrix operation	No. of multiplication	No. of addition	No. of store
1	$\mathbf{M}_{4\times4}\mathbf{p}_{4\times1}$	16	12	4
2	$\mathbf{M}_{4\times4}\mathbf{R}_{4\times4}$	64	48	16

Performance issue—cont.

- Assume multiplication costs 4 times of clock-tick of addition. And the performances of “addition” and “store” are almost the same.
- Note: m is the no. of $\mathbf{M}_{44}$, n is no. of vertex
- Case 1 totally needs 76mn times of one addition operation

$$\mathbf{p}^* = \mathbf{M}^m{}_{4\times4}[\mathbf{M}^{m-1}{}_{4\times4}\ldots(\mathbf{M}^1{}_{4\times4}\mathbf{p}_{4\times n})]$$

- Case 2 needs 304(m-1)+76n of one addition operation

$$\mathbf{p}^* = [\prod_{i=1}^{m}(\mathbf{M}^i{}_{4\times4})]\mathbf{p}_{4\times n}$$

- In case of $n > 4$, “case 2” performs better than “case 1”

Performance issue—cont.

- Single instruction, multiple data (SIMD).

- Total time = “data move in” + operation (speed up) + “data move out”

$$C_{4\times 4} = A_{4\times 4} B_{4\times 4}$$

$$C_{4\times 1} = A_{4\times 4} B_{4\times 1}$$

3D transformation—pose estimation

$$\mathbf{p}^* = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{p} = \begin{bmatrix} 1 & 0 & 0 & a_{14} \\ 0 & 1 & 0 & a_{24} \\ 0 & 0 & 1 & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boxed{\begin{matrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{matrix}} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 & 0 \end{matrix} & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}$$

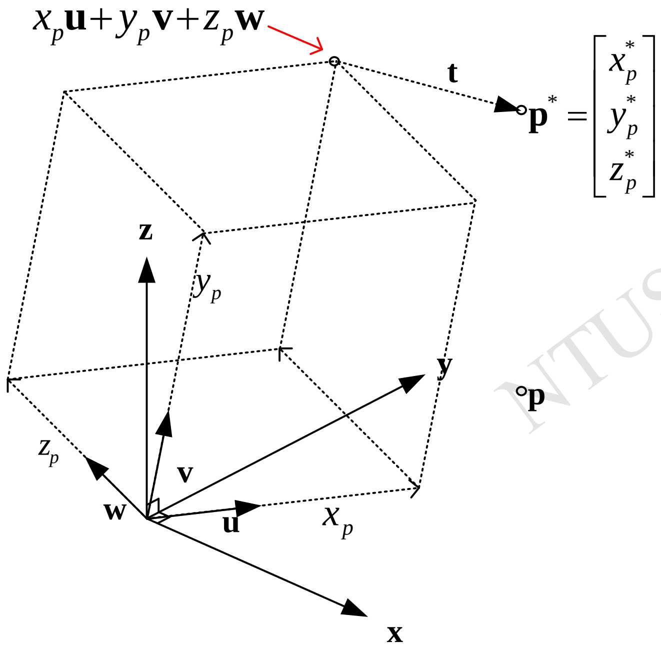
$$\mathbf{u} = [a_{11} \quad a_{21} \quad a_{31} \quad 0]^{\mathrm{T}}$$

$$\mathbf{v} = [a_{12} \quad a_{22} \quad a_{32} \quad 0]^{\mathrm{T}}$$

$$\mathbf{w} = [a_{13} \quad a_{23} \quad a_{33} \quad 0]^{\mathrm{T}}$$

$$\mathbf{t} = [a_{14} \quad a_{24} \quad a_{34} \quad 1]^{\mathrm{T}}$$

$$\mathbf{p}^* = x_p\mathbf{u} + y_p\mathbf{v} + z_p\mathbf{w} + \mathbf{t}$$



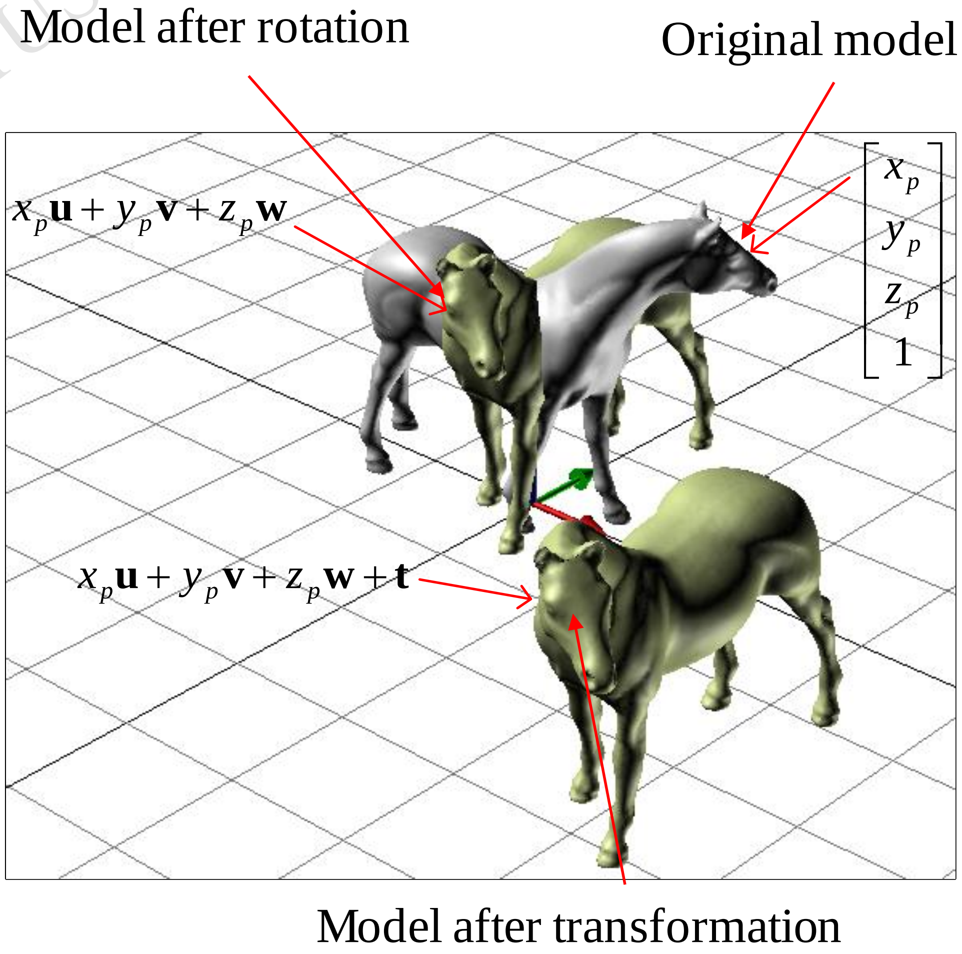
3D transformation—pose estimation

- Similar to 2D, the 3D transformation can be decomposed into one translation (shift) after one rotation (orientation), if the transformation is orthogonal.

$$\mathbf{p}^* = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{p} = \begin{bmatrix} 1 & 0 & 0 & a_{14} \\ 0 & 1 & 0 & a_{24} \\ 0 & 0 & 1 & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}$$

3D transformation—pose estimation

- Example

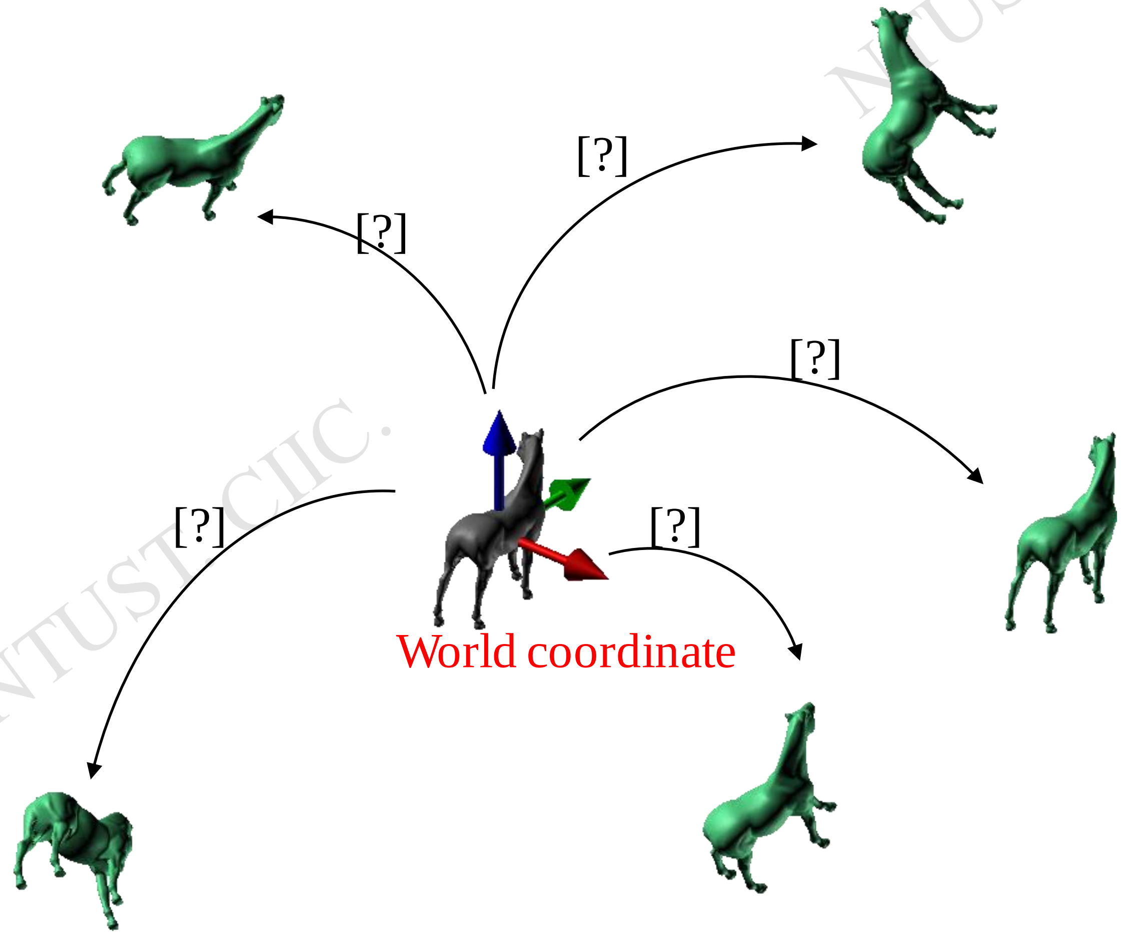


$$\mathbf{u}=[a_{11} \quad a_{21} \quad a_{31} \quad 0]^{\mathrm{T}}$$
$$\mathbf{v}=[a_{12} \quad a_{22} \quad a_{32} \quad 0]^{\mathrm{T}}$$
$$\mathbf{w}=[a_{13} \quad a_{23} \quad a_{33} \quad 0]^{\mathrm{T}}$$
$$\mathbf{t}=[a_{14} \quad a_{24} \quad a_{34} \quad 1]^{\mathrm{T}}$$

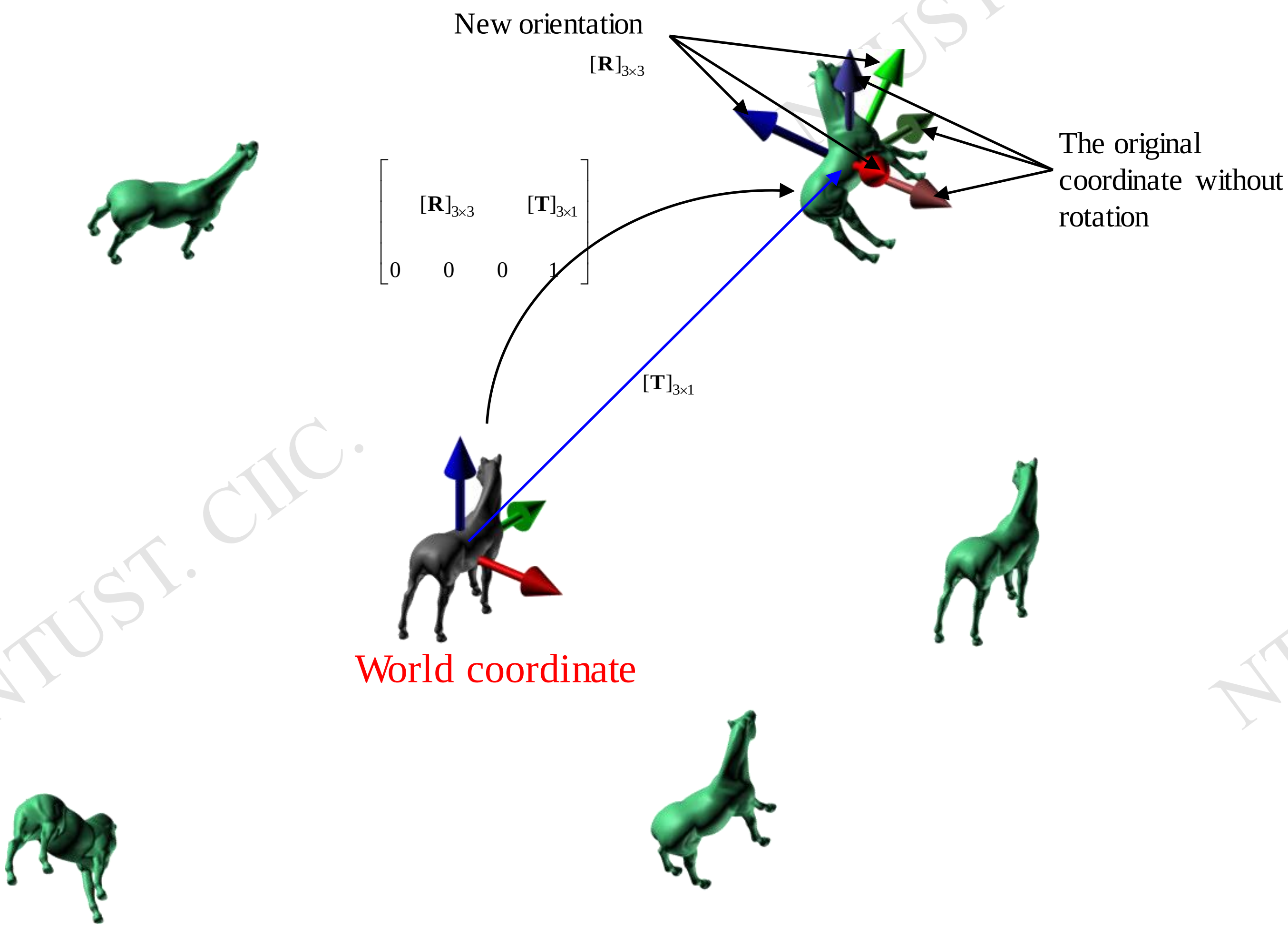
$$\mathbf{p}^*=x_p\mathbf{u}+y_p\mathbf{v}+z_p\mathbf{w}+\mathbf{t}$$

3D transformation—pose estimation

- How to ?

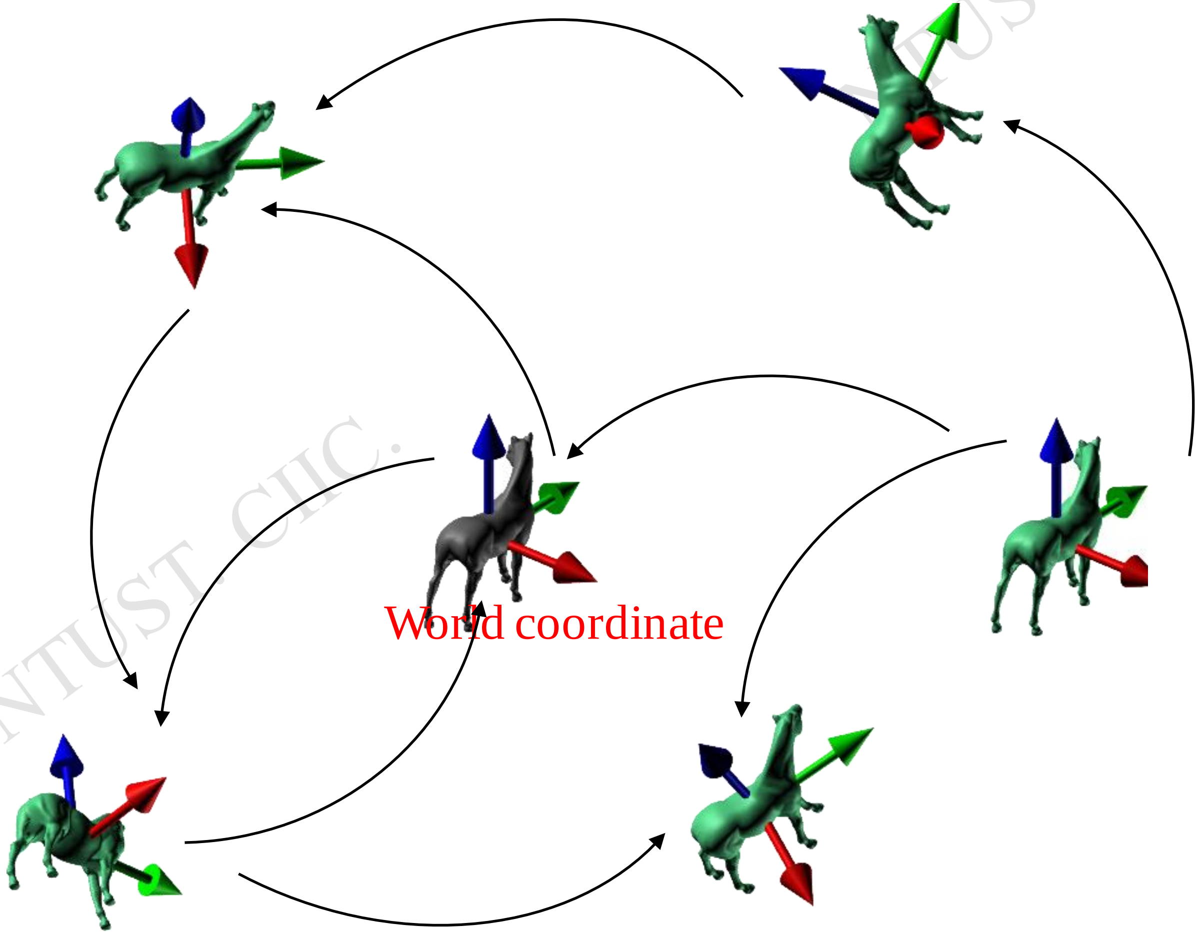


3D transformation—pose estimation

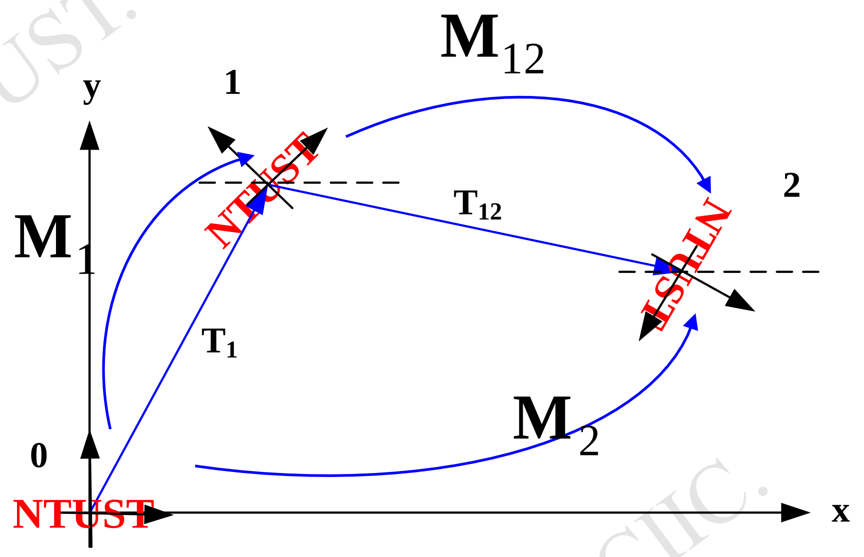


3D transformation—pose estimation

- Combine two transformation matrixes



Short summary: recall 2D transformation



If we know
$\text{Obj1} = \mathbf{M}_1 * \text{Obj0}$
$\text{Obj2} = \mathbf{M}_2 * \text{Obj0}$

Consider Obj1→Obj2 Transformation Matrix ($\mathbf{M}_{12}$)

Method-1

$$\because \text{Obj2} = \mathbf{M}_2 \text{Obj0} = \mathbf{M}_2 (\mathbf{M}_1)^{-1} \text{Obj1}$$

$$\therefore \mathbf{M}_{12} = \mathbf{M}_2 (\mathbf{M}_1)^{-1}$$

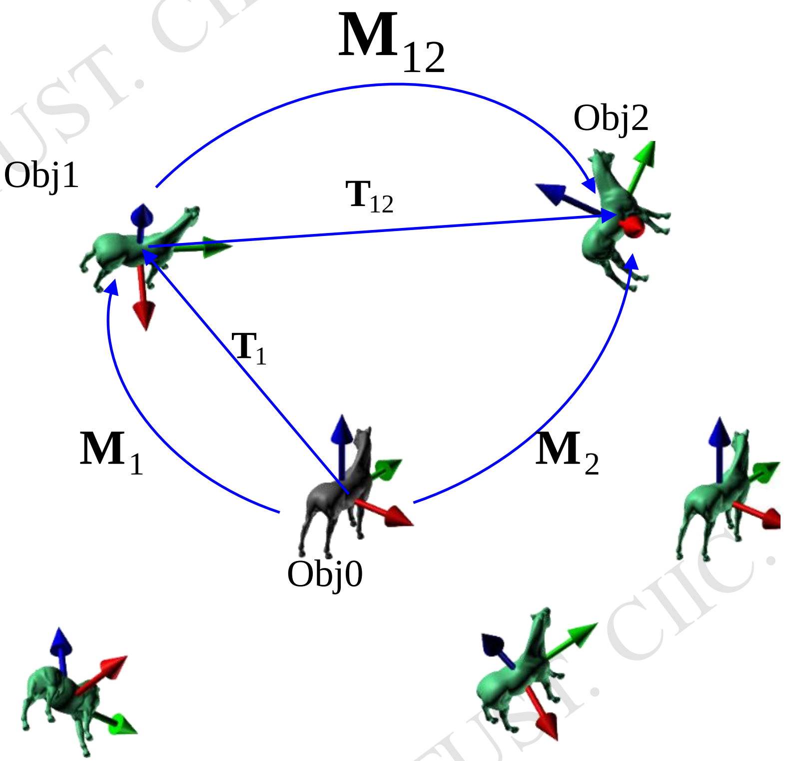
Method-2

Initially, rotate Obj1 to consistent with Obj2 under the same orientation(**R**), then, translate Obj1 to destination.

$$\because \text{Obj2} = \mathbf{T}_{12} \mathbf{T}_1 \mathbf{R} \mathbf{T}_1^{-1} \text{Obj1}$$

$$\therefore \mathbf{M}_{12} = \mathbf{T}_{12} \mathbf{T}_1 \mathbf{R} \mathbf{T}_1^{-1}$$

Short summary: recall 3D transformation



If we know
$\text{Obj1} = \mathbf{M}_1 * \text{Obj0}$
$\text{Obj2} = \mathbf{M}_2 * \text{Obj0}$

Similar to 2D case
Method-1

$$\because \text{Obj2} = \mathbf{M}_2 \text{Obj0} = \mathbf{M}_2 (\mathbf{M}_1)^{-1} \text{Obj1}$$
$$\therefore \mathbf{M}_{12} = \mathbf{M}_2 (\mathbf{M}_1)^{-1}$$

Method-2

$$\because \text{Obj2} = \mathbf{T}_{12} \mathbf{T}_1 \mathbf{R} \mathbf{T}_1^{-1} \text{Obj1}$$
$$\therefore \mathbf{M}_{12} = \mathbf{T}_{12} \mathbf{T}_1 \mathbf{R} \mathbf{T}_1^{-1}$$

How to determine an orthogonal coordinate

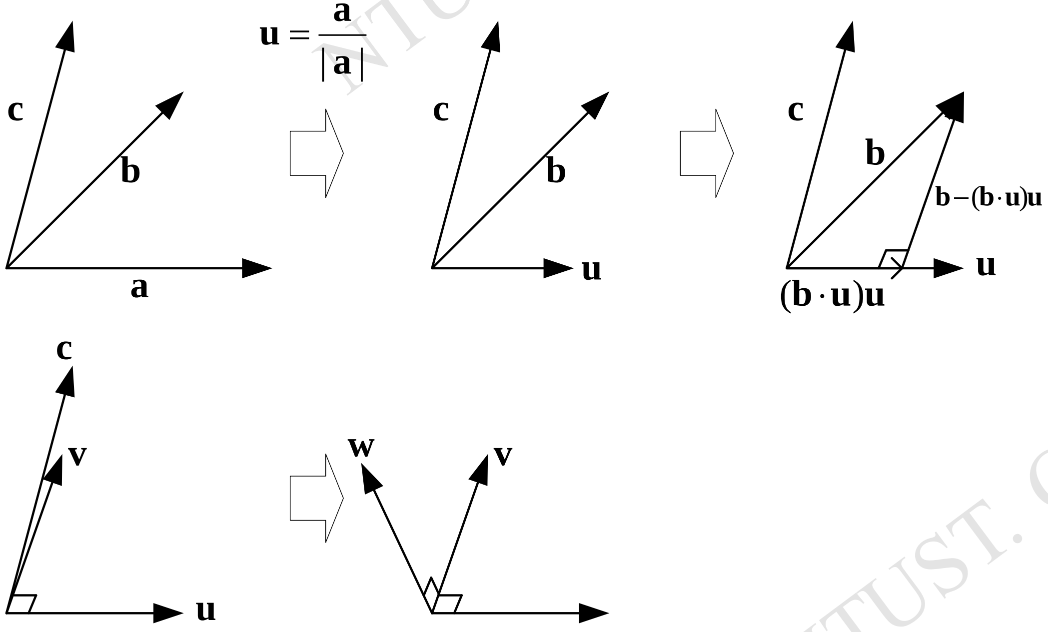
- Gram-Schmidt

$$\mathbf{u} = \frac{\mathbf{a}}{|\mathbf{a}|}$$

$$\mathbf{v} = \frac{\mathbf{b} - (\mathbf{b}\cdot\mathbf{u})\mathbf{u}}{|\mathbf{b} - (\mathbf{b}\cdot\mathbf{u})\mathbf{u}|}$$

$$\mathbf{w} = \frac{\mathbf{c} - (\mathbf{c}\cdot\mathbf{u})\mathbf{u} - (\mathbf{c}\cdot\mathbf{v})\mathbf{v}}{|\mathbf{c} - (\mathbf{c}\cdot\mathbf{u})\mathbf{u} - (\mathbf{c}\cdot\mathbf{v})\mathbf{v}|}$$

or $\mathbf{w} = \mathbf{u} \times \mathbf{v}$



- Note: this method can't guarantee to follow "Right-Hand Rule"

色彩與照明科技研究所
Graduate Institute of Color and Illumination Technology

Find us on: facebook