Introduction à l'Informatique Graphique

Lecture 2. Geometrical Transformations

Caroline Larboulette

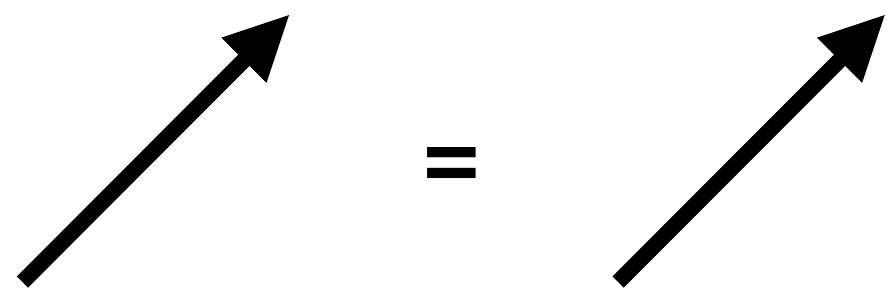
Overview

- Mathematical Preliminaries
- 2D Transformations
- Homogeneous Coordinates
- Composition of 2D Transformations
- 3D Transformations
- Composition of 3D Transformations
- Change of Coordinate System

Mathematical Preliminaries

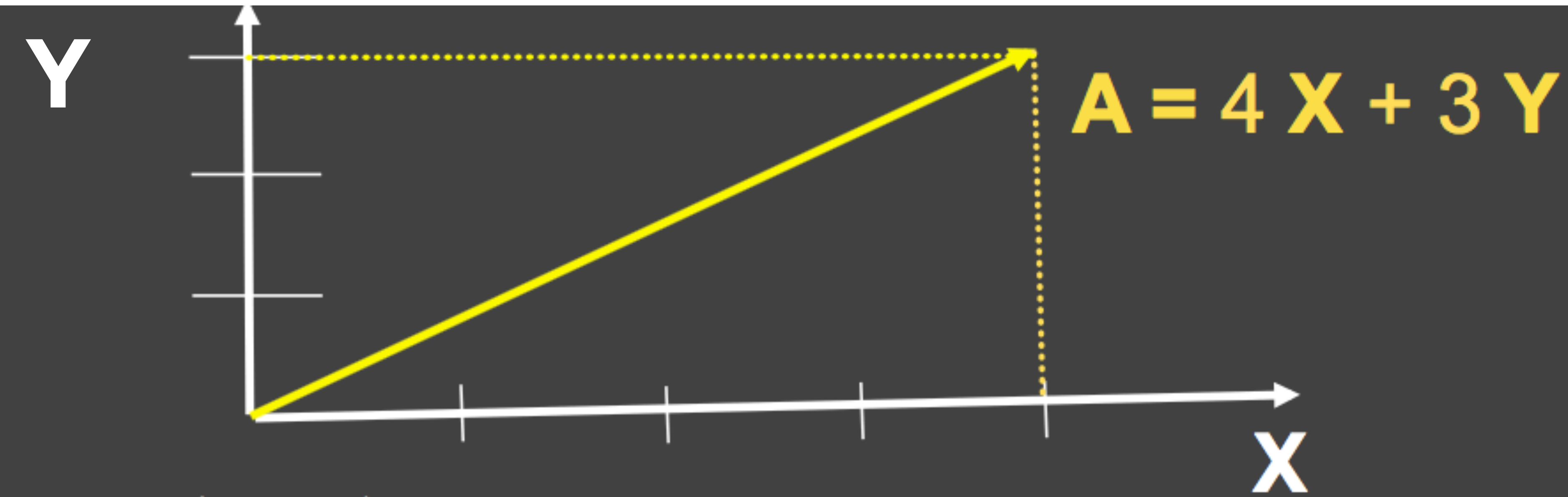
Vectors and Matrices

Vectors



- Usually written $\overrightarrow{a}$ or in bold $\mathbf{a}$
- Magnitude / norm / length written $||\overrightarrow{a}||$
- Gives length and direction, absolute position is not important
- Used to store displacements, offsets

Cartesian Coordinates



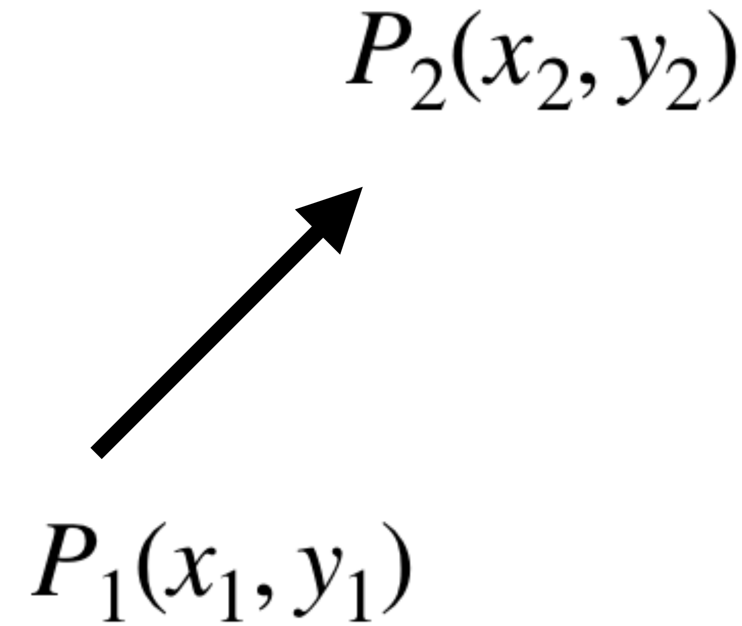
$$A = \begin{pmatrix} x \\ y \end{pmatrix} \qquad A^T = \begin{pmatrix} x & y \end{pmatrix} \qquad \|A\| = \sqrt{x^2 + y^2}$$

- X and Y can be any (usually orthogonal unit) vectors
- To normalize a vector: $\vec{a}_{unit} = \dfrac{\vec{a}}{||\vec{a}||}$

Cartesian Coordinates

- Creating a vector from two points

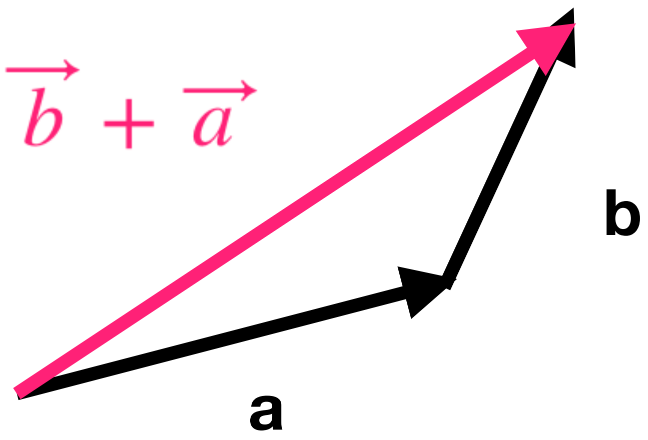
$$\vec{a} = \overrightarrow{P_1P_2} = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix}$$



- Vectors $\overrightarrow{P_1P_2}$ and $\overrightarrow{P_2P_1}$ are opposite

Vector Addition

$$\overrightarrow{c} = \overrightarrow{a} + \overrightarrow{b} = \overrightarrow{b} + \overrightarrow{a}$$

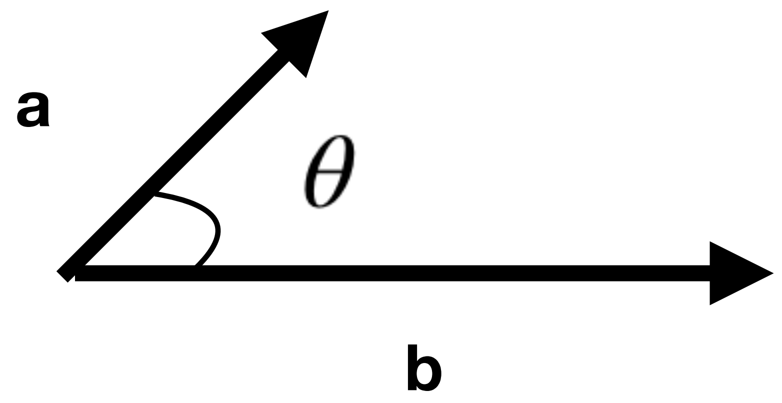


- In cartesian coordinates, simply add coordinates
- $\overrightarrow{c} = \overrightarrow{a} + \overrightarrow{b} = \overrightarrow{b} + \overrightarrow{a}$
  - $c_x = a_x + b_x$
  - $c_y = a_y + b_y$
  - $c_z = a_z + b_z$

Vector Multiplication

- Dot product
- Cross product
- Context
  - Orthonormal bases and coordinate frames
  - We use standard right-handed coordinate systems

Dot / Scalar Product



- Takes two vectors and gives a scalar

$$\overrightarrow{a}\,.\,\overrightarrow{b} = \overrightarrow{b}\,.\,\overrightarrow{a} = ||\,\overrightarrow{a}\,||\,.\,||\,\overrightarrow{b}\,||\,.\cos(\theta)$$

$$\theta = \cos^{-1}\left(\frac{\overrightarrow{a}\,.\,\overrightarrow{b}}{||\,\overrightarrow{a}\,||\,.\,||\,\overrightarrow{b}\,||}\right)$$

$$\overrightarrow{a}\,.(\overrightarrow{b} + \overrightarrow{c}\,) = \overrightarrow{a}\,.\,\overrightarrow{b} + \overrightarrow{a}\,.\,\overrightarrow{c}$$

$$(k\overrightarrow{a}\,)\,.\,\overrightarrow{b} = \overrightarrow{a}\,.(k\overrightarrow{b}\,) = k(\overrightarrow{a}\,.\,\overrightarrow{b}\,)$$

Dot / Scalar Product

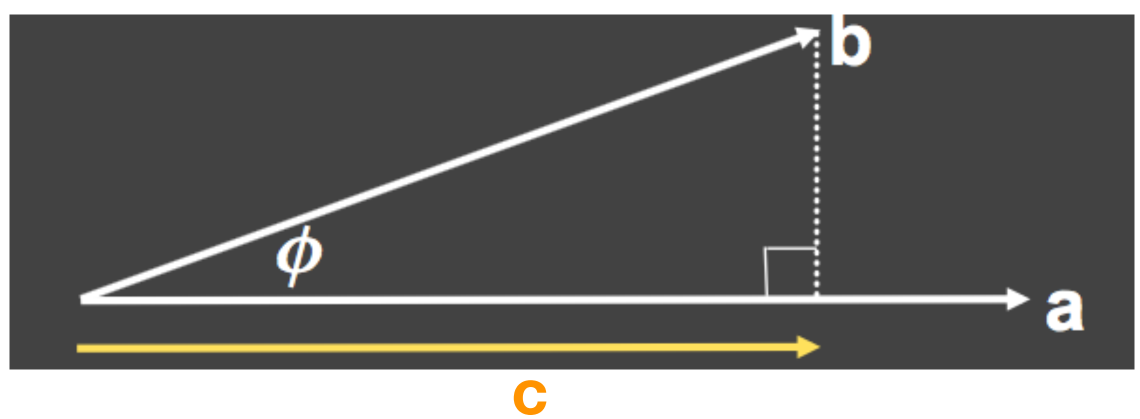
Computation

$$\overrightarrow{a} \cdot \overrightarrow{b} = \begin{bmatrix} x_a \\ y_a \\ z_a \end{bmatrix} \cdot \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = x_a \cdot x_b + y_a \cdot y_b + z_a \cdot z_b$$

Dot / Scalar Product

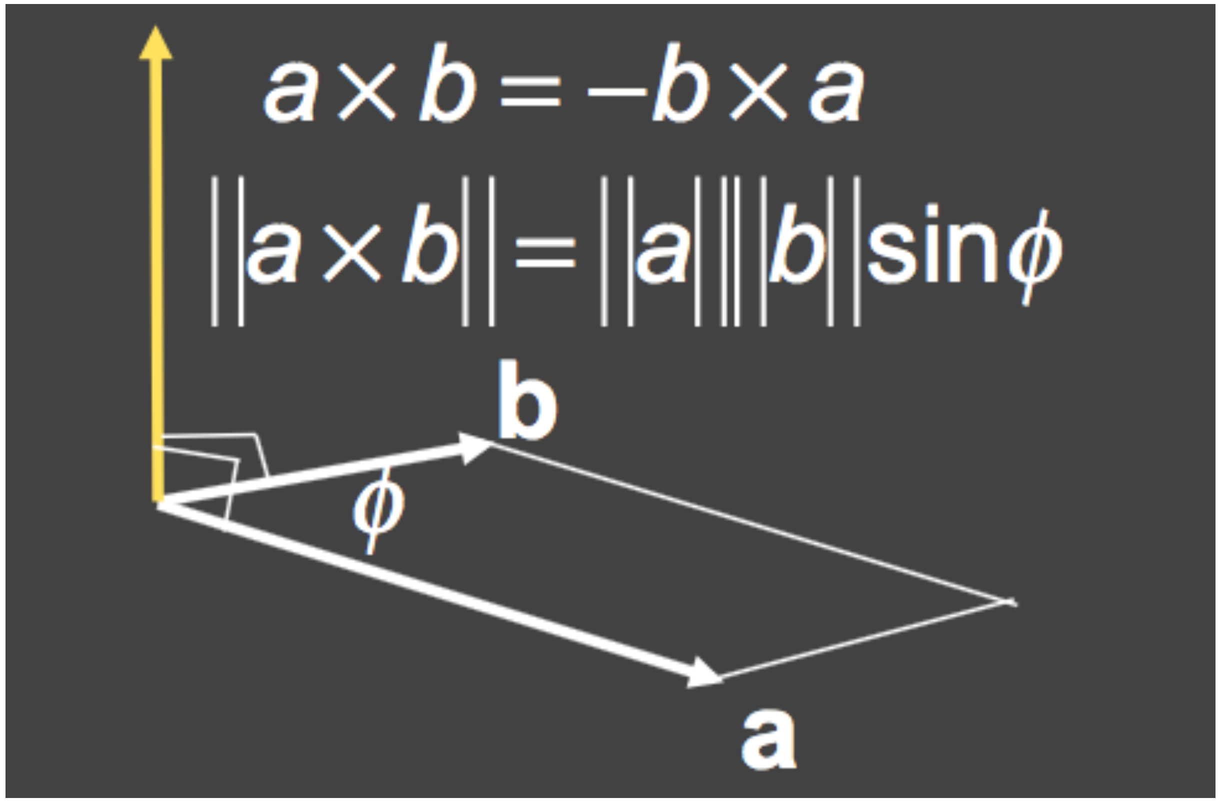
Applications in CG

- **Find angle between two vectors, e.g., between light source and surface for rendering**
- **Find projection of one vector on another, e.g. coordinates of points in arbitrary coordinate system**



- **If a is unit,** $\overrightarrow{a} \cdot \overrightarrow{b} = ||\overrightarrow{a}|| \cdot ||\overrightarrow{b}|| \cos\theta = ||\overrightarrow{b}|| \cos\theta = ||\overrightarrow{c}|| = c$

Cross / Vector Product



- Takes two vectors and gives a vector
- This vector is orthogonal to the first two
- Direction determined by right hand rule

Cross / Vector Product

Properties

- **Coordinate System Axes**

$$X \times Y = +Z$$
$$Y \times X = -Z$$
$$Y \times Z = +X$$
$$Z \times Y = -X$$
$$Z \times X = +Y$$
$$X \times Z = -Y$$

Cross / Vector Product
Properties

$$\overrightarrow{a} \times \overrightarrow{b} = -\overrightarrow{b} \times \overrightarrow{a}$$

$$||\overrightarrow{a} \times \overrightarrow{b}|| = ||\overrightarrow{a}|| \cdot ||\overrightarrow{b}|| \cdot \sin \phi$$

$$\overrightarrow{a} \times \overrightarrow{a} = 0$$

$$\overrightarrow{a} \times (\overrightarrow{b} + \overrightarrow{c}) = \overrightarrow{a} \times \overrightarrow{b} + \overrightarrow{a} \times \overrightarrow{c}$$

$$\overrightarrow{a} \times (k\overrightarrow{b}) = k(\overrightarrow{a} \times \overrightarrow{b})$$

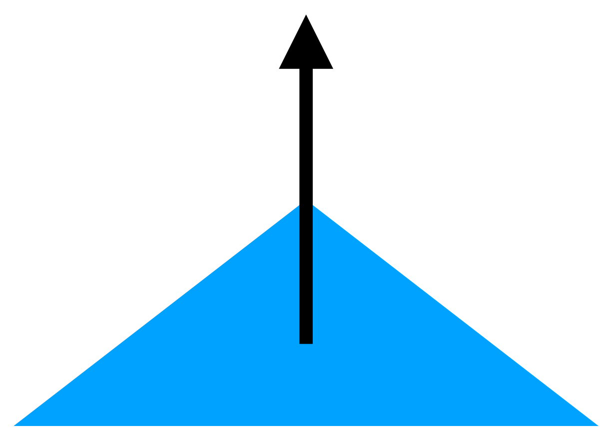
Cross / Vector Product

Computation

$$\overrightarrow{a} \times \overrightarrow{b} = \begin{bmatrix} x_a \\ y_a \\ z_a \end{bmatrix} \times \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} y_a \cdot z_b - z_a \cdot y_b \\ z_a \cdot x_b - x_a \cdot z_b \\ x_a \cdot y_b - y_a \cdot x_b \end{bmatrix}$$

Cross / Vector Product

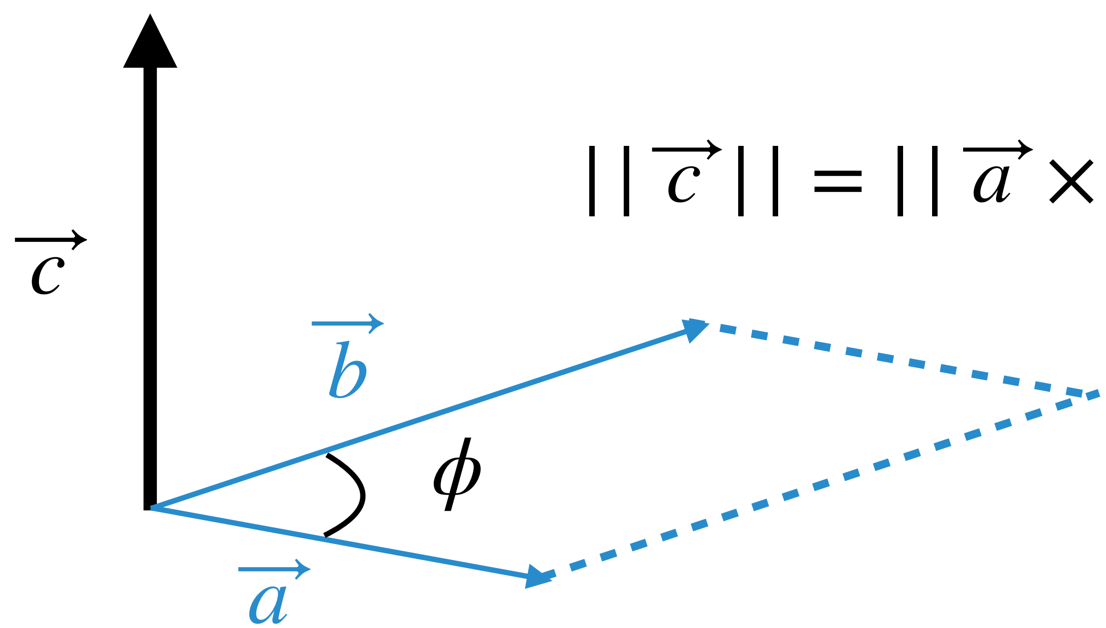
Applications



- Construction of coordinate systems
- Computation of surface normals (e.g., for lighting and shading computations)

Cross / Vector Product

Applications



$$||\overrightarrow{c}|| = ||\overrightarrow{a} \times \overrightarrow{b}|| = ||\overrightarrow{a}|| \cdot ||\overrightarrow{b}|| \cdot \sin\phi$$

- **The magnitude of the product equals the area of a parallelogram with the vectors for sides**
- **The magnitude of the product of two perpendicular vectors ($\sin\phi = \sin(90) = 1$) is the product of their lengths**
- **Area of triangle is half the magnitude**

Matrix

- Used to transform points or vectors
  - Translation, rotation, shear, scale …
- 3D visualisation
- Represent curves and surface
- Usually written with a capital letter A

Matrix

- Array of m rows, n columns = m.n numbers

$$\begin{pmatrix} 1 & 3 \\ 5 & 2 \\ 0 & 4 \end{pmatrix}$$

- Addition / subtraction, only for matrices of the same size, work element by element

$$(A+B)_{ij} = A_{ij} + B_{ij}$$
$$(A-B)_{ij} = A_{ij} - B_{ij}$$

$$\begin{bmatrix} 2 & -3 \\ 1 & 0 \\ -1 & 3 \end{bmatrix} + \begin{bmatrix} 9 & -5 \\ 0 & 13 \\ -1 & 3 \end{bmatrix} = \begin{bmatrix} 11 & -8 \\ 1 & 13 \\ -2 & 6 \end{bmatrix}$$

Scalar Multiplication

- c.A is obtained by multiplying each element of A by c

$$4\begin{bmatrix} 2 & -3 & 0 \\ 1 & 0 & 4 \\ -1 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 8 & -12 & 0 \\ 4 & 0 & 16 \\ -4 & 12 & 4 \end{bmatrix}$$

Matrix-Matrix Multiplication

- Number of columns of first matrix needs to be equal to number of rows of second matrix
- Element (ij) in production is dot product or row i of first matrix with column j of second matrix

$$\begin{pmatrix} 1 & 3 \\ 5 & 2 \\ 0 & 4 \end{pmatrix}\begin{pmatrix} 3 & 6 & 9 & 4 \\ 2 & 7 & 8 & 3 \end{pmatrix}$$

Matrix-Matrix Multiplication

- Number of columns of first matrix needs to be equal to number of rows of second matrix
- Element (ij) in production is dot product or row i of first matrix with column j of second matrix

$$\begin{pmatrix} 1 & 3 \\ 5 & 2 \\ 0 & 4 \end{pmatrix} \begin{pmatrix} 3 & 6 & 9 & 4 \\ 2 & 7 & 8 & 3 \end{pmatrix} = \begin{pmatrix} 9 & 27 & 33 & 13 \\ 19 & 44 & 61 & 26 \\ 8 & 28 & 32 & 12 \end{pmatrix}$$

Matrix-Matrix Multiplication

- Number of columns of first matrix needs to be equal to number of rows of second matrix
- Element (ij) in production is dot product or row i of first matrix with column j of second matrix

$$\begin{pmatrix} 1 & 3 \\ 5 & 2 \\ 0 & 4 \end{pmatrix} \begin{pmatrix} 3 & 6 & 9 & 4 \\ 2 & 7 & 8 & 3 \end{pmatrix} = \begin{pmatrix} 9 & 27 & 33 & 13 \\ 19 & 44 & 61 & 26 \\ 8 & 28 & 32 & 12 \end{pmatrix}$$

Matrix-Matrix Multiplication

- Number of columns of first matrix needs to be equal to number of rows of second matrix
- Element (ij) in production is dot product or row i of first matrix with column j of second matrix

$$\begin{pmatrix} 1 & 3 \\ 5 & 2 \\ 0 & 4 \end{pmatrix} \begin{pmatrix} 3 & 6 & 9 & 4 \\ 2 & 7 & 8 & 3 \end{pmatrix} = \begin{pmatrix} 9 & 27 & 33 & 13 \\ 19 & 44 & 61 & 26 \\ 8 & 28 & 32 & 12 \end{pmatrix}$$

Matrix-Matrix Multiplication

- Non-commutative
  - AB and BA are different (if different sizes, one is not possible)
- Associative and Distributive
  - $A(B+C) = AB + AC$
  - $(A+B)C = AC + BC$

Matrix-Vector Multiplication

- Just treat the vector as a column matrix (m x 1)
- Example: 2D reflection about y-axis

$$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -x \\ y \end{pmatrix}$$

Matrix Transpose

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}^T = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$$

$$(AB)^T = B^T A^T$$

Matrix Identity and Inverse

$$I_{3\times 3} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$AA^{-1} = A^{-1}A = I$$

$$(AB)^{-1} = B^{-1}A^{-1}$$

Vector Multiplication in Matrix From

- Dot product: a.b = aT.b

$$\begin{pmatrix} x_a & y_a & z_a \end{pmatrix} \begin{pmatrix} x_b \\ y_b \\ z_b \end{pmatrix} = \begin{pmatrix} x_a x_b + y_a y_b + z_a z_b \end{pmatrix}$$

- Cross product: a^b = A*.b

$$\begin{pmatrix} 0 & -z_a & y_a \\ z_a & 0 & -x_a \\ -y_a & x_a & 0 \end{pmatrix} \begin{pmatrix} x_b \\ y_b \\ z_b \end{pmatrix}$$

A* is dual matrix of vector a

2D Transformations

Translation

- Moves a point P to new point P'
- Displacement on x-axis = dx
- Displacement on y-axis = dy

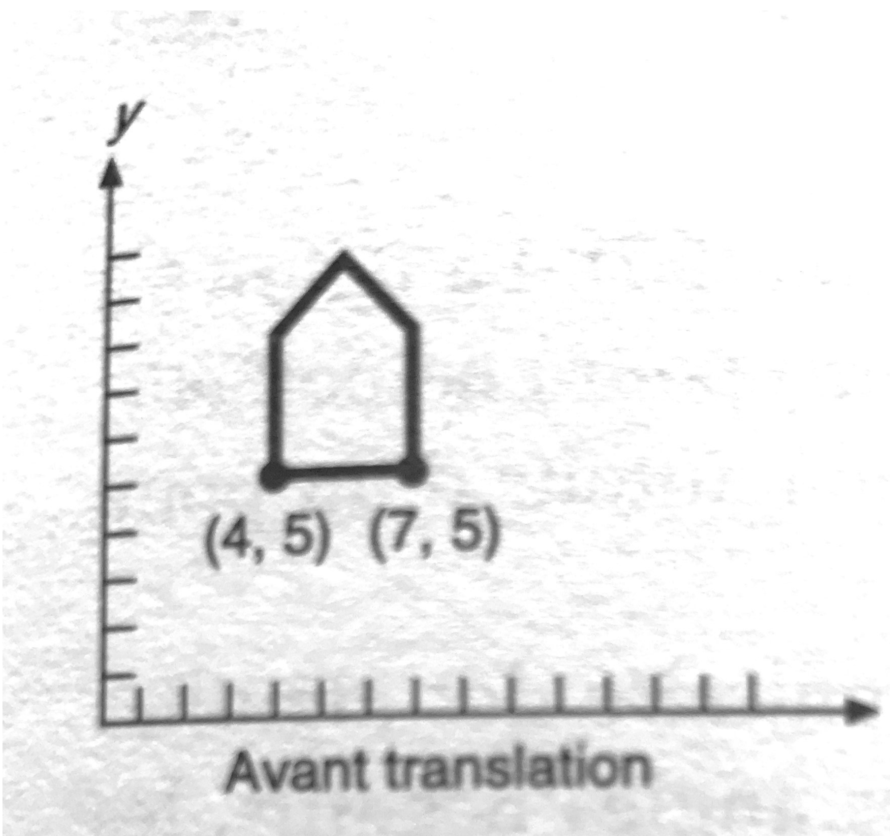
x' = x+dx , y' = y + dy

$$P \begin{bmatrix} x \\ y \end{bmatrix}, P' \begin{bmatrix} x' \\ y' \end{bmatrix}, T \begin{bmatrix} dx \\ dy \end{bmatrix}$$

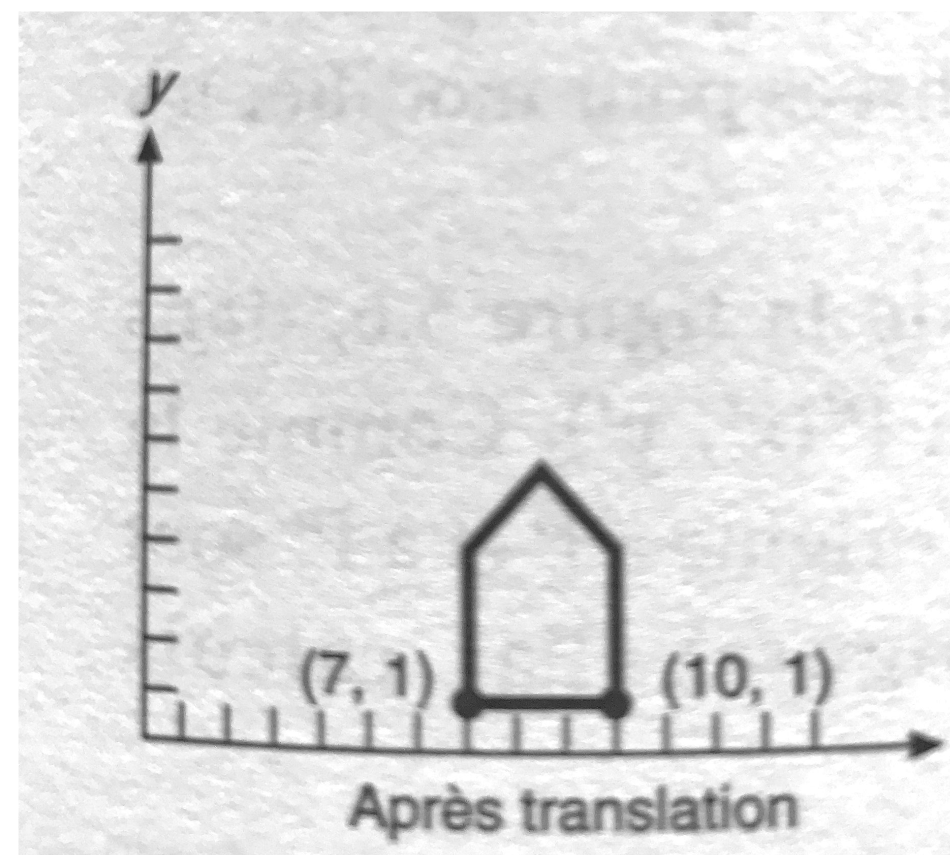
Then P' = P + T

Translation

- To translate an object, we apply the translation to the points defining the object



$$T\begin{bmatrix} 3 \\ -4 \end{bmatrix}$$



Scale

- Changes size of an object
- Homothecy relative to origin of coordinate system
- $x' = S_x \cdot x \ , \ y' = S_y \cdot y$
- P’ = S.P

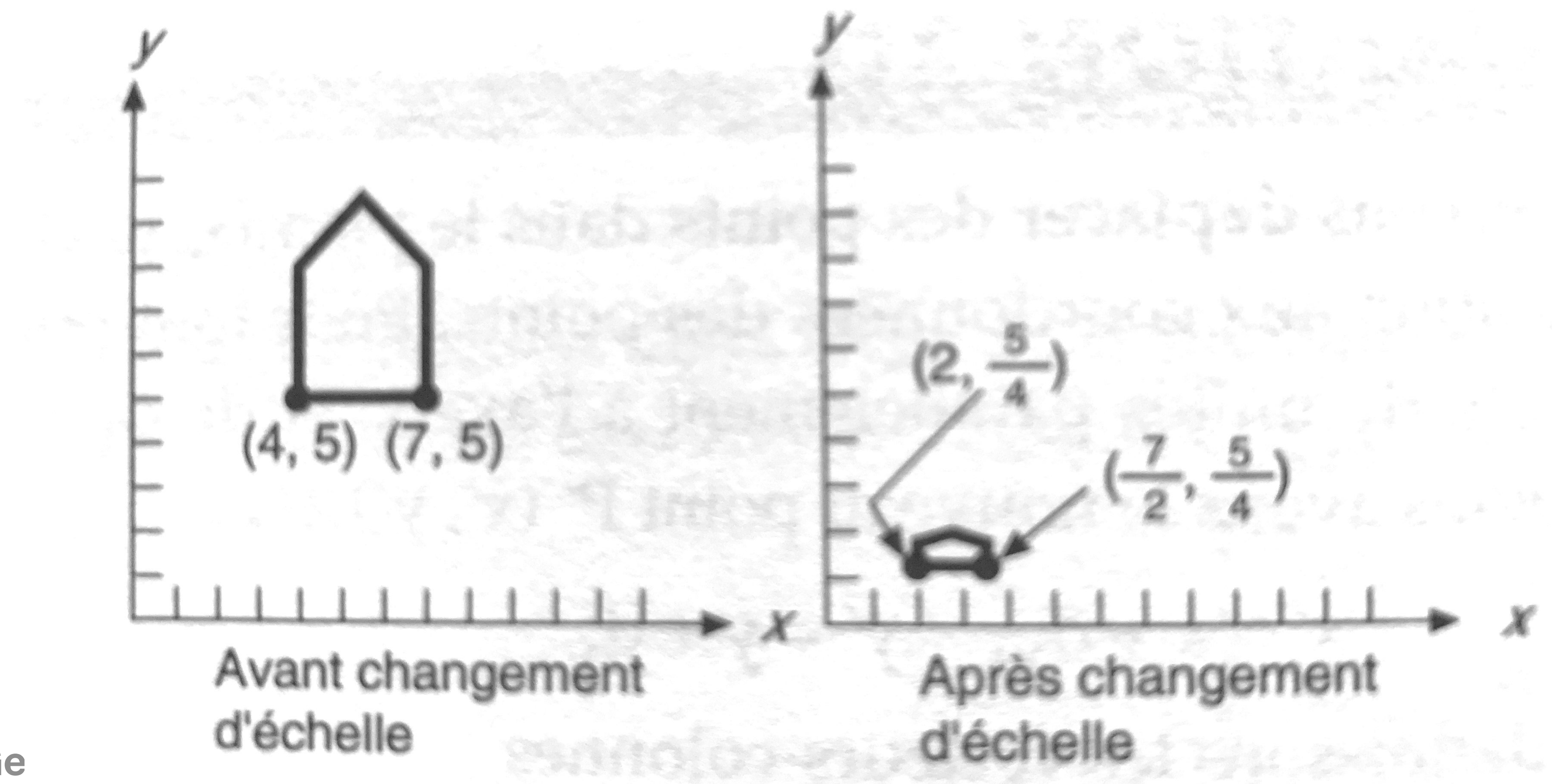
$$Scale(s_x, s_y) = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \qquad S^{-1} = \begin{pmatrix} s_x^{-1} & 0 \\ 0 & s_y^{-1} \end{pmatrix}$$

$$\begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} s_x x \\ s_y y \\ s_z z \end{pmatrix}$$

Scale

Example

- $S_x = 0.5$ **and** $S_y = 0.25$
- **if sx = sy then scale is uniform and proportions are not changed**



Shear

$$Shear = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix} \qquad S^{-1} = \begin{pmatrix} 1 & -a \\ 0 & 1 \end{pmatrix}$$

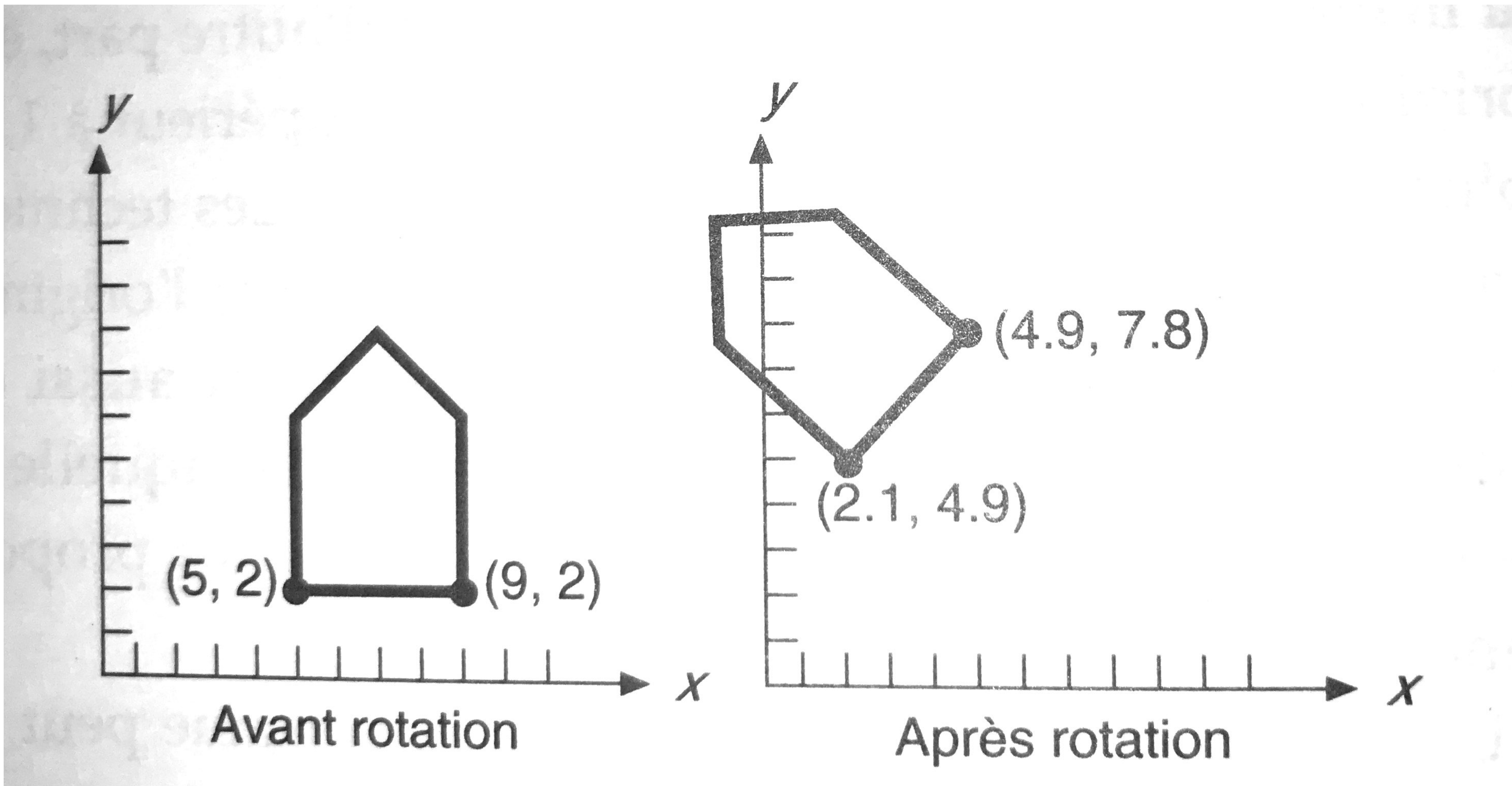
Rotation

- **Rotation of angle Theta around origin**
- $x' = x \cdot \cos\theta - y \cdot \sin\theta$
- $y' = x \cdot \sin\theta + y \cdot \cos\theta$
- **P’ = R.P**

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Rotation

- 45 deg rotation



Homogeneous Coordinates

To Sum Up

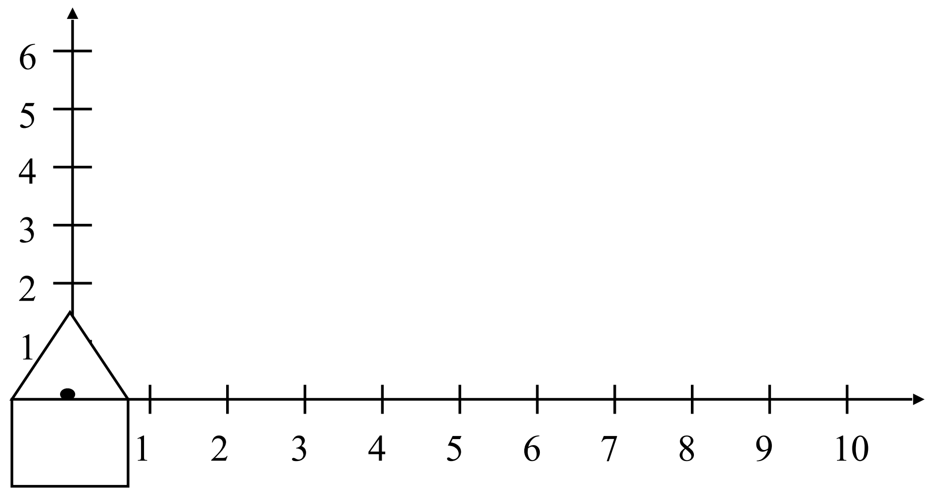
- Translation
  - P’ = T + P
- Scale
  - P’ = S.P
- Rotation
  - P’ = R.P
- Problem: translation is not a multiplication
- Solution: homogeneous coordinates

Composing Transforms

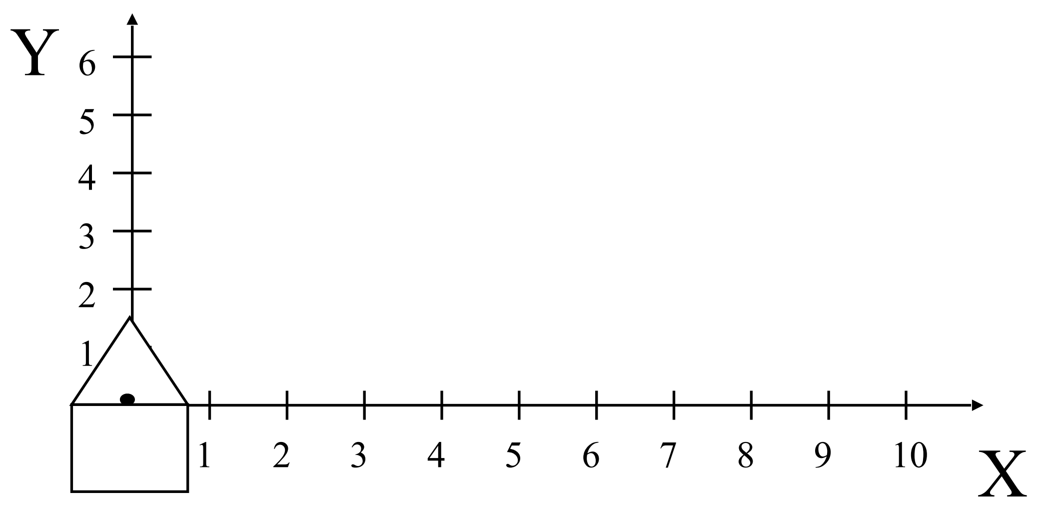
- Often want to combine transforms
  - First scale by 2, then rotate by 45 degrees
- X2 = S.X1
- X3 = R.X2
- X3 = R.(S.X1) = (R.S).X1
- But X3 != (S.R).X1
- Not commutative, order matters !

Not commutative

Translate by
$x = 6, y = 0$, then
rotate by 45^o



Rotate by 45^o,
then translate
by $x = 6, y = 0$



Composing Transforms

- What about the translation ?
- We do not multiply, we add !
- $P' = T + P$

Problem Formulation

- Example: move by 5 units on x-axis (y unchanged)
- How to obtain the transformation matrix ?

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x + 5 \\ y \end{bmatrix}$$

Homogeneous Coordinates

- Add a third homogeneous coordinate w (w=1) to point
- 3x3 matrix (4x4 matrix in 3D)
- Last row is set to 0 0 1

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x+5 \\ y \\ 1 \end{bmatrix}$$

Point Representation

$$P = \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} x/w \\ y/w \\ 1 \end{bmatrix}$$

- Divide by w to get (inhomogeneous) point
- Multiplication by w > 0 has no effect
- For w = 0, we have an infinite point

Composition of Translations

Homogeneous Coordinates

- P' = T1.P
- P'' = T2.P'
- P'' = (T2.T1).P
- Matrix form...

$$T(dx_1, dy_1) = T_1 = \begin{bmatrix} 1 & 0 & dx_1 \\ 0 & 1 & dy_1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(dx_2, dy_2) = T_2 = \begin{bmatrix} 1 & 0 & dx_2 \\ 0 & 1 & dy_2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(dx_2, dy_2) \cdot T(dx_1, dy_1) =$$

Composition of Scales
Homogeneous Coordinates

- P' = S1.P
- P'' = S2.P'
- P'' = (S2.S1).P
- Matrix form…

Rotation in Homogeneous Coordinates

- P’ = R(theta).P

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

Composition of Rotations

Homogeneous Coordinates

$$R_1(\theta) = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- P' = R1.P
- P'' = R2.P'
- P'' = (R2.R1).P
- Matrix form...

$$R_2(\theta) = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

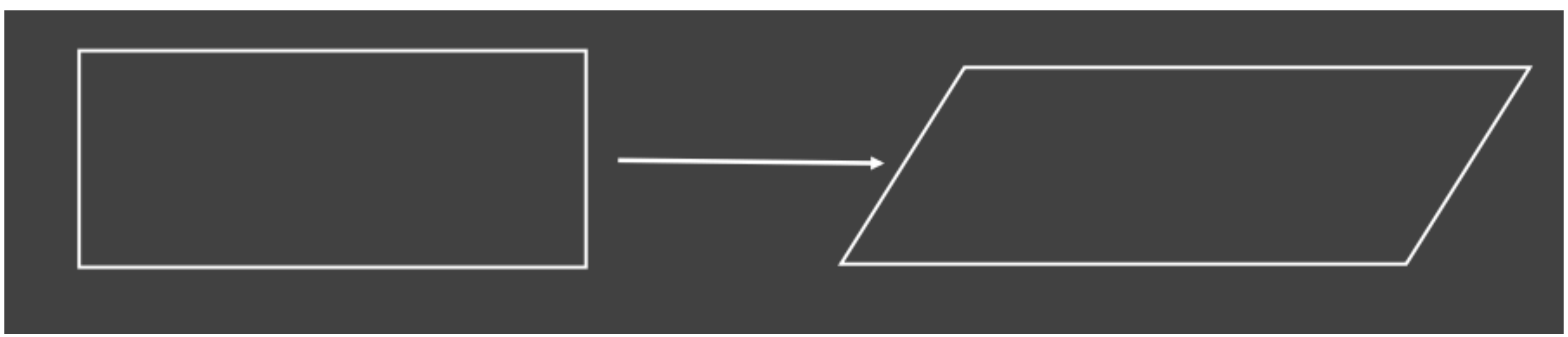
$$R_2(\theta) \,.\, R_1(\theta) =$$

Shear in Homogeneous Coordinates

- **Cx : shear along x (as a function of y)**

$$x' = x + a \cdot y$$
$$y' = y$$

$$C_x = \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Shear in Homogeneous Coordinates

- Cx : shear along x-axis (as a function of y)

$$x' = x + a \cdot y \qquad y' = y$$

$$C_x = \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Cy : shear along y-axis (as a function of x)

$$x' = x \qquad y' = y + b \cdot x$$

$$C_y = \begin{bmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Composition of 2D Transformations

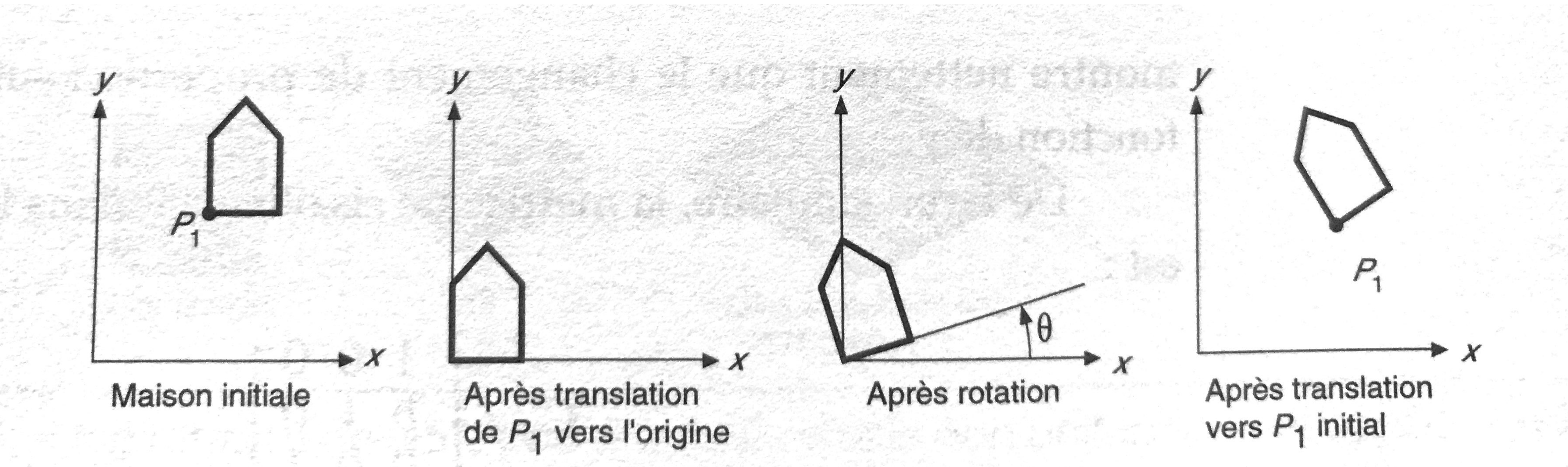
Composition

- We want to combine R, S and T to obtain general affine transformations
- More efficient ! Application of a single transformation instead of a sequence

Rotation Around Point

- We want to rotate around a point P1 (instead of the origin)
- Decomposition into 3 problems:
  1. Translate P1 (the object) to origin
  2. Rotate the object
  3. Translate the object back in P1

Rotation Around Point



$$T(x_1, y_1)\,.\,R(\theta)\,.\,T(-x_1, -y_1) = \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & -x_1 \\ 0 & 1 & -y_1 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotation Around Point

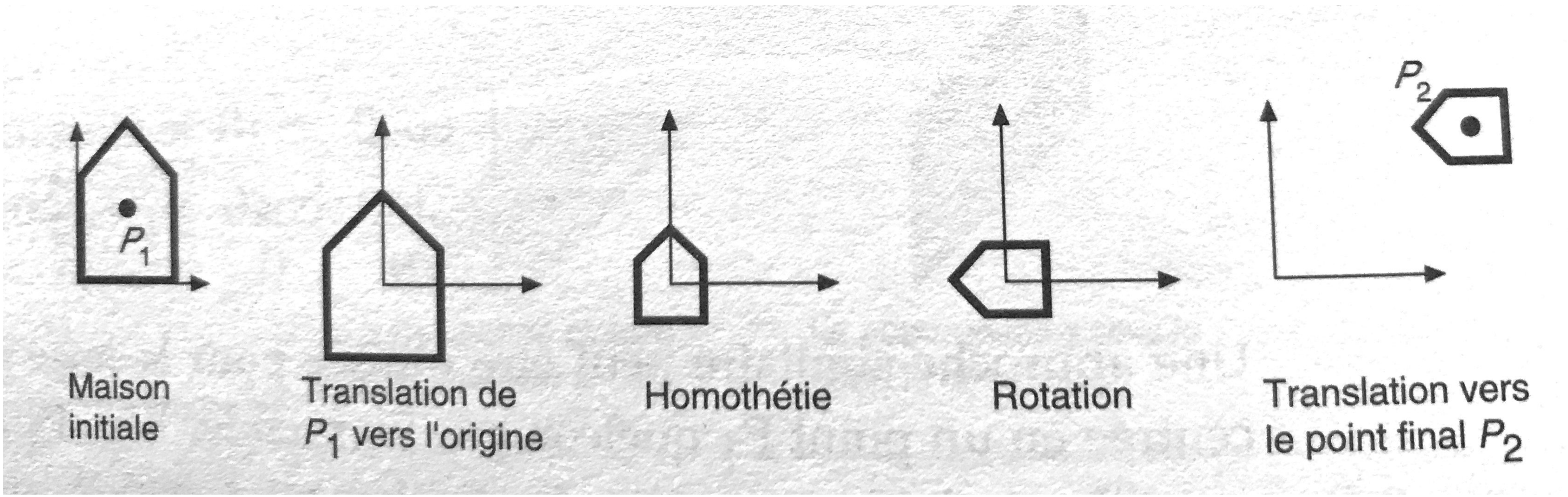
$$T(x_1, y_1)\,.\,R(\theta)\,.\,T(-x_1, -y_1) = \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & -x_1 \\ 0 & 1 & -y_1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta) & -\sin(\theta) & x_1(1 - \cos(\theta)) + y_1 \sin(\theta) \\ \sin(\theta) & \cos(\theta) & y_1(1 - \cos(\theta)) - x_1 \sin(\theta) \\ 0 & 0 & 1 \end{bmatrix}$$

Scale and Rotation Around Point

- We want to scale an object (around P1, its center), rotate around P1, then translate to P2
- The sequence of actions is
  - Translate P1 to origin
  - Scale the object
  - Rotate the object
  - Translate P1 to P2

Scale and Rotation Around Point



- T(x2,y2).R(theta).S(sx,sy).T(-x1,-y1)

Inverting Composite Transform

- M = M1.M2.M3
- Option 1: Find M-1
- Option 2: Inverse each transform and swap order

$$
\begin{aligned}
M &= M_1 M_2 M_3 \\
M^{-1} &= M_3^{-1} M_2^{-1} M_1^{-1} \\
M^{-1}M &= M_3^{-1}(M_2^{-1}(M_1^{-1}M_1)M_2)M_3
\end{aligned}
$$

Inverting Translate

- If you translate by X, the inverse is translating by -X

$$T\text{-}1 = \begin{bmatrix} 1 & 0 & -dx \\ 0 & 1 & -dy \\ 0 & 0 & 1 \end{bmatrix}$$

Inverting Scale

- If you scale something by a, the inverse scaling is 1/a

S-1 = $$\begin{bmatrix} 1/s_x & 0 & 0 \\ 0 & 1/s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Inverting Rotate

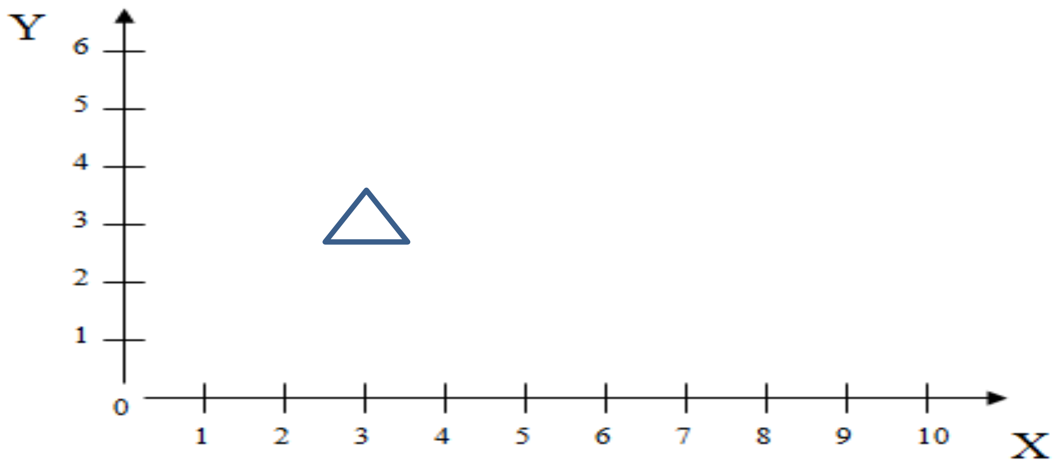
- The inverse of a rotation by θ is a rotation by - θ

$$\cos(-\theta) = \cos(\theta)$$
$$\sin(-\theta) = -\sin(\theta)$$

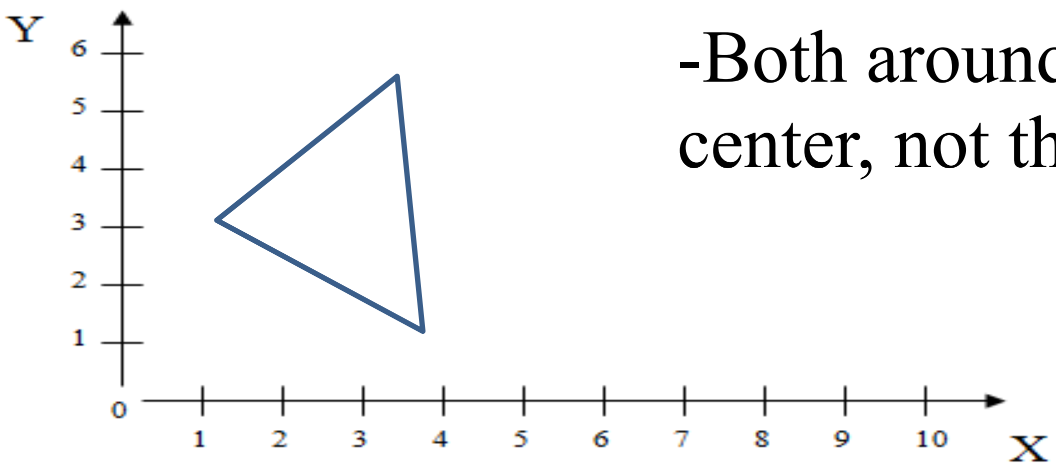
$$R(\theta)^{-1} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Composition (an example) (2D) (1/2)

- Start:



Goal:



-Rotate 90^o
-Uniform scale 4x
-Both around object's center, not the origin

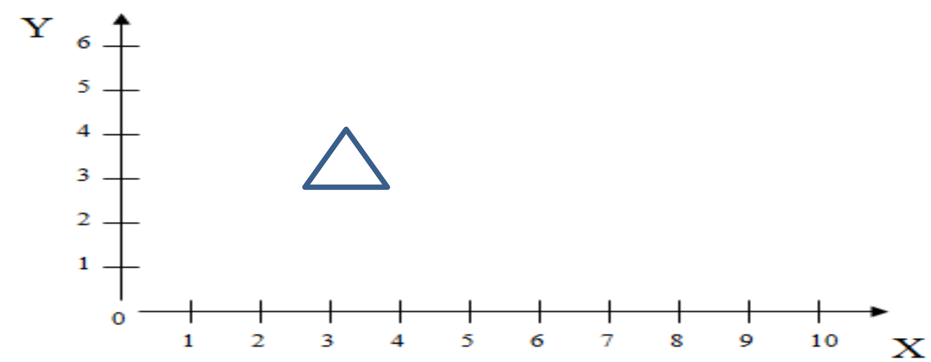
- Important concept: make the problem simpler
- Translate object to origin first, scale, rotate, and translate back:

$$T^{-1}RST = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 90 & -\sin 90 & 0 \\ \sin 90 & \cos 90 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix}$$

- Apply to all vertices

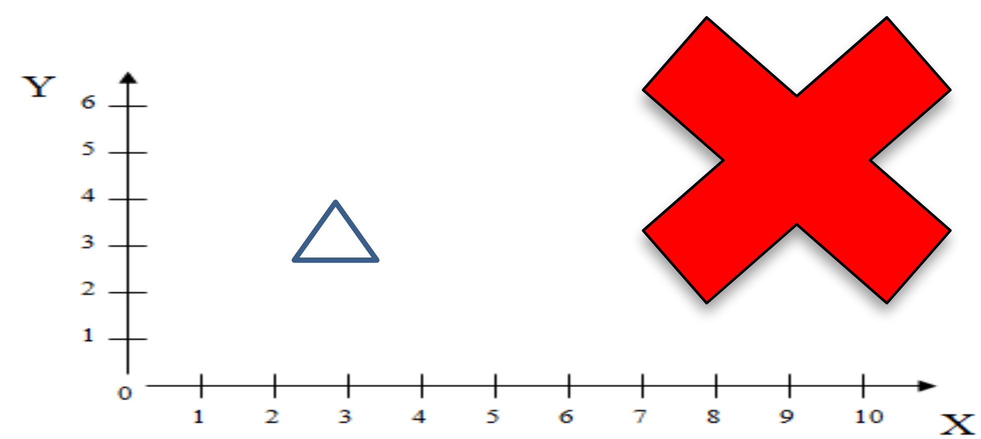
Composition (an example) (2D) (2/2)

- ***T^{-1}RST***



- But what if we mixed up the order? Let's try ***RT^{-1}ST***:

$$\begin{bmatrix} \cos 90 & -\sin 90 & 0 \\ \sin 90 & \cos 90 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix}$$



- Oops! We scaled properly, but when we rotated the object, it's center was not at the origin, so its position was shifted. Order matters!

General Matrix

$$M = \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

- The 2x2 sub-matrix r11-r22 corresponds to the composition of rotations and scales
- tx, ty correspond to the composition of translations

3D

Transformations

3D Transformations

- In 2D: 3x3 matrices
- In 3D: 4x4 matrices
- We use the homogeneous coordinates the same way

$$P(x, y, z) = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} x/w \\ y/w \\ z/w \\ 1 \end{bmatrix}$$

General Translation Matrix

$$T = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} I_3 & T \\ 0 & 1 \end{bmatrix}$$

$$P' = T \cdot P = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x + T_x \\ y + T_y \\ z + T_z \\ 1 \end{bmatrix} = P + T$$

General Scale Matrix

$$S(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P' = S \,.\, P = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} s_x \,.\, x \\ s_y \,.\, y \\ s_z \,.\, z \\ 1 \end{bmatrix}$$

Rotation about z-axis

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

- Similarly, we can define the rotation around the other two axes

Rotation about x-axis

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

Rotation about y-axis

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

Shear along xy-axis

Function of z

$$x' = x + c_x \cdot z$$
$$y' = y + c_y \cdot z$$
$$z' = z$$

$$C_{xy}(c_x, c_y) = \begin{bmatrix} 1 & 0 & c_x & 0 \\ 0 & 1 & c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Shear along yz-axis

Function of x

$$x' = x$$
$$y' = y + c_y \cdot x$$
$$z' = z + c_z \cdot x$$

$$C_{yz}(c_y, c_z) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ c_y & 1 & 0 & 0 \\ c_z & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Shear along zx-axis

Function of y

$$
\begin{aligned}
x' &= x + c_x \cdot y \\
y' &= y \\
z' &= z + c_z \cdot y
\end{aligned}
$$

$$
C_{xy}(c_x, c_y) = \begin{bmatrix} 1 & c_x & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & c_z & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Composition of 3D Transformations

Combining Translations and Rotations

- Rotation first : order matters ! Not correct if translation first ! (next slide)

$$P' = (T \cdot R) \cdot P = M \cdot P = R \cdot P + T$$

$$M = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} R_{11} & R_{12} & R_{13} & 0 \\ R_{21} & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \left[\begin{array}{ccc|c} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}$$

Combining Translations and Rotations

- If translation first, not correct !

$$P' = (R \cdot T) \cdot P = M \cdot P = R \cdot (P + T) = R \cdot P + R \cdot T$$

$$M = \begin{bmatrix} R_{11} & R_{12} & R_{13} & 0 \\ R_{21} & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{3\times3} & R_{3\times3} \cdot T_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix}$$

Inverse Scaling

$$S(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S^{-1}(s_x, s_y, s_z) = \begin{bmatrix} 1/s_x & 0 & 0 & 0 \\ 0 & 1/s_y & 0 & 0 \\ 0 & 0 & 1/s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Inverse Translation

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

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & -T_x \\ 0 & 1 & 0 & -T_y \\ 0 & 0 & 1 & -T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Inverse Rotations

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

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

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

Example in 3D!

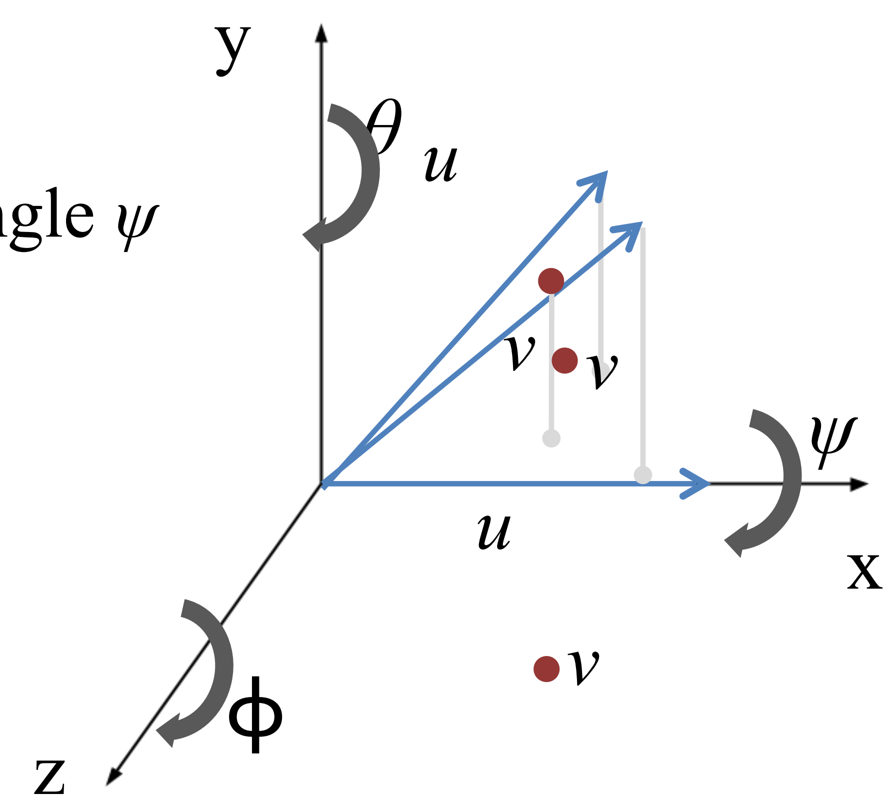
- Let's take some 3D object, say a cube centered at (2,2,2)
- Rotate clockwise in object's space by 30° around x axis, 60° around y, and 90° around z
- Scale in object space by 1 in the x, 2 in the y, 3 in the z
- Translate by (2,2,4) in world space
- Transformation sequence: $\boldsymbol{T T_0^{-1} S_{xy} R_{xy} R_{zx} R_{yz} T_o}$, where $\boldsymbol{T_0}$ translates to (0,0):

$$\underbrace{\begin{bmatrix}1&0&0&2\\0&1&0&2\\0&0&1&4\\0&0&0&1\end{bmatrix}}_{\boldsymbol{T}}\underbrace{\begin{bmatrix}1&0&0&2\\0&1&0&2\\0&0&1&2\\0&0&0&1\end{bmatrix}}_{\boldsymbol{T_0^{-1}}}\underbrace{\begin{bmatrix}1&0&0&0\\0&2&0&0\\0&0&3&0\\0&0&0&1\end{bmatrix}}_{\boldsymbol{S_{xy}}}\underbrace{\begin{bmatrix}\cos 90&\sin 90&0&0\\-\sin 90&\cos 90&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}}_{\boldsymbol{R_{xy}}}\underbrace{\begin{bmatrix}\cos 60&0&-\sin 60&0\\0&1&0&0\\\sin 60&0&\cos 60&0\\0&0&0&1\end{bmatrix}}_{\boldsymbol{R_{zx}}}\underbrace{\begin{bmatrix}1&0&0&0\\0&\cos 30&\sin 30&0\\0&-\sin 30&\cos 30&0\\0&0&0&1\end{bmatrix}}_{\boldsymbol{R_{yz}}}\underbrace{\begin{bmatrix}1&0&0&-2\\0&1&0&-2\\0&0&1&-2\\0&0&0&1\end{bmatrix}}_{\boldsymbol{T_0}}$$

Composition of Rotations

- Not commutative
- Rotate by x, then y is not the same as rotate by y then x
- Order of applying rotations matters
  - R1*R2 different from R2*R1
- Solution, apply 3 rotations (Euler angles)

Rotating axis by axis (2/2)



- It would still be difficult to find the 3 angles to rotate by, given arbitrary axis u and specified angle ψ
- Solution? Make the problem easier by mapping u to one of the principal axes

- **Step 1**: Find a θ to rotate around y axis to put u in the xy plane
- **Step 2**: Then find a ϕ to rotate around the z axis to align u with the x axis

Now that **u** *is in a convenient alignment, we can do our transformation rotation for vertex* v*:*

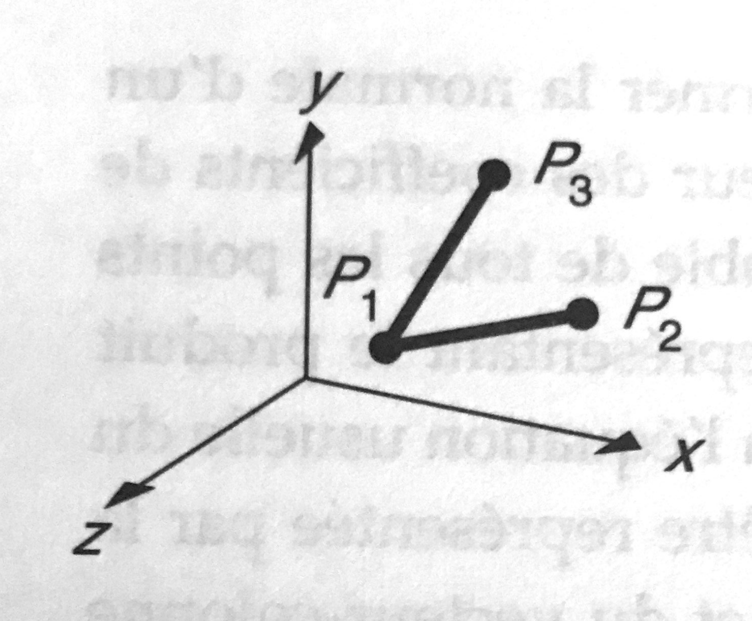
- **Step 3**: Rotate v by ψ around x axis (which is coincident with u axis)
- **Step 4**: Finally, undo the alignment rotations (inverse).

The only rotation we've preserved is the one around axis u *by* ψ, *which was our goal*

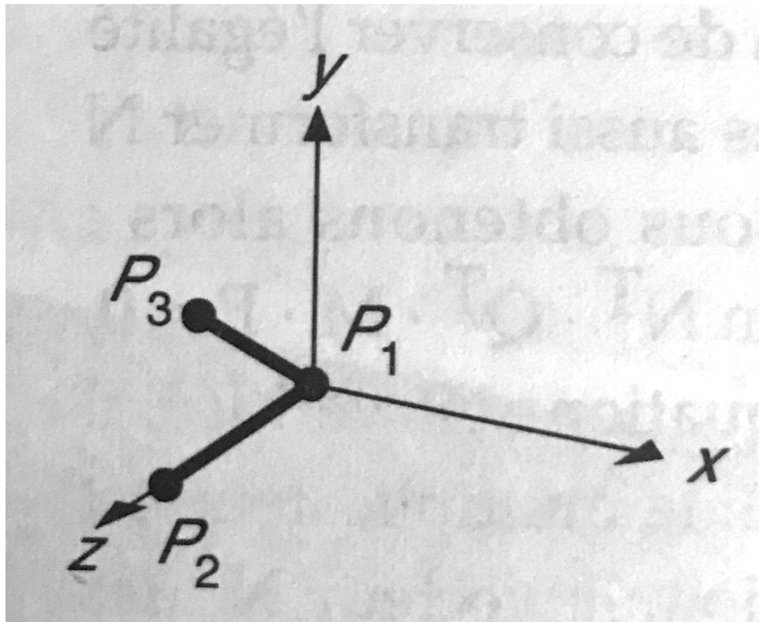
- Rotation matrix: $\boldsymbol{M} = \boldsymbol{R}_{zx}{}^{-1}(\theta)\boldsymbol{R}_{xy}{}^{-1}(\phi)\boldsymbol{R}_{yz}(\psi)\boldsymbol{R}_{xy}(\phi)\boldsymbol{R}_{zx}(\theta)$

Transformation of Segments

- We want to transform P1P2 and P1P3

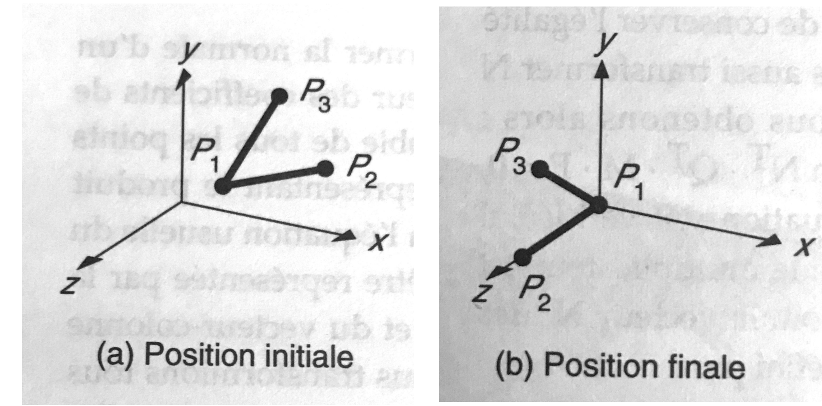


(a) Position initiale



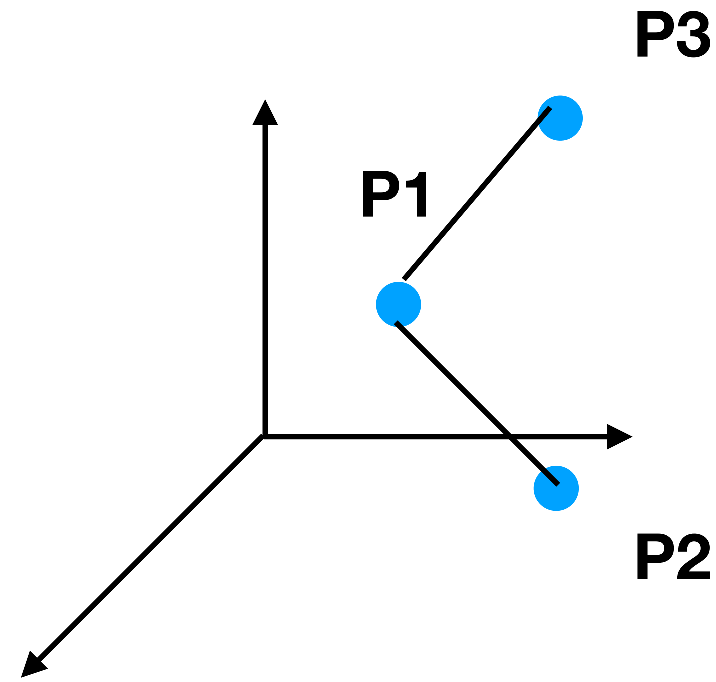
(b) Position finale

Transformation of Segments

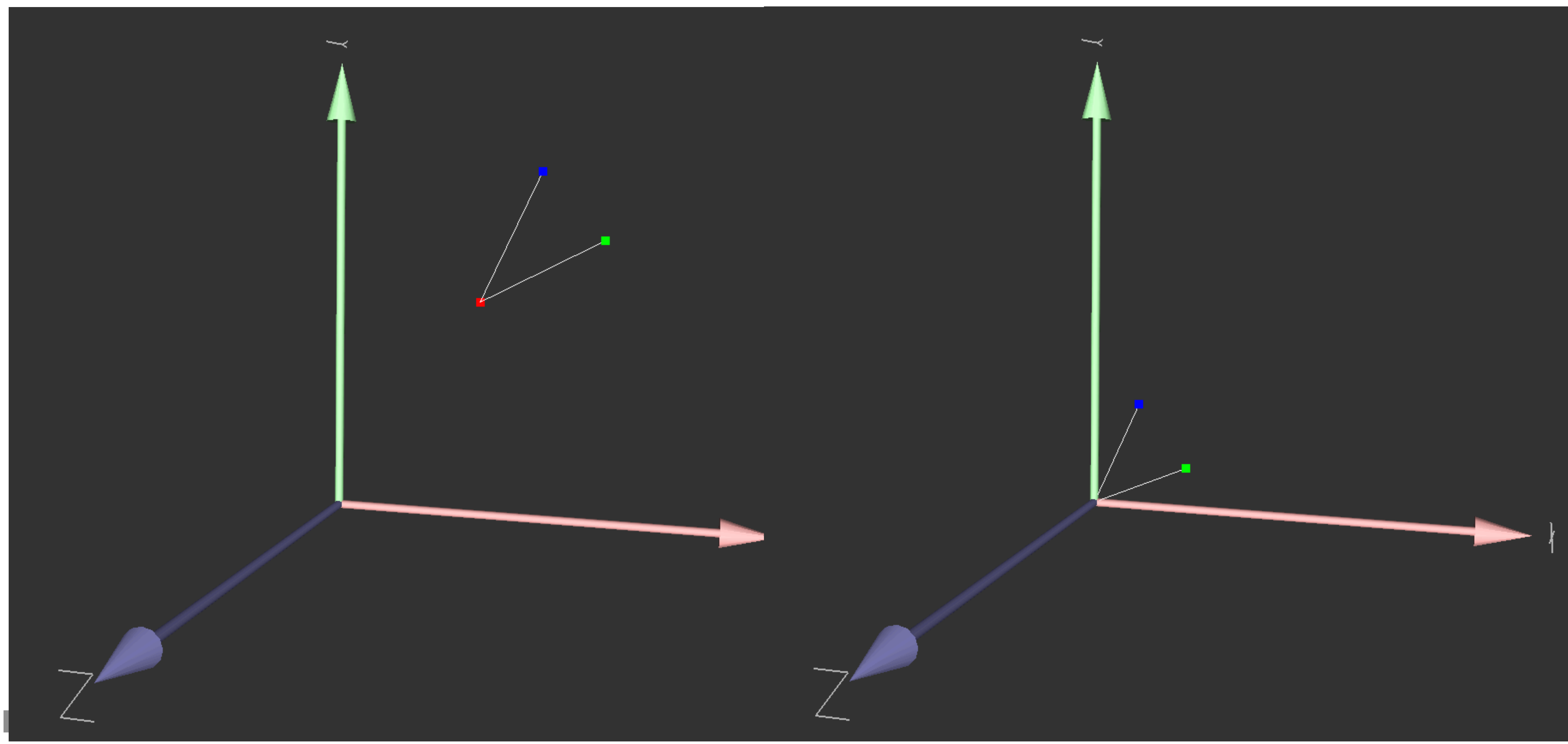


- We want to transform P1P2 and P1P3
    1. Move P1 to origin
    2. Rotate around y-axis to move P1P2 into (y,z) plane
    3. Rotate around x-axis to have P1P2 onto z-axis
    4. Rotate around z-axis to move P1P3 into (y,z) plane

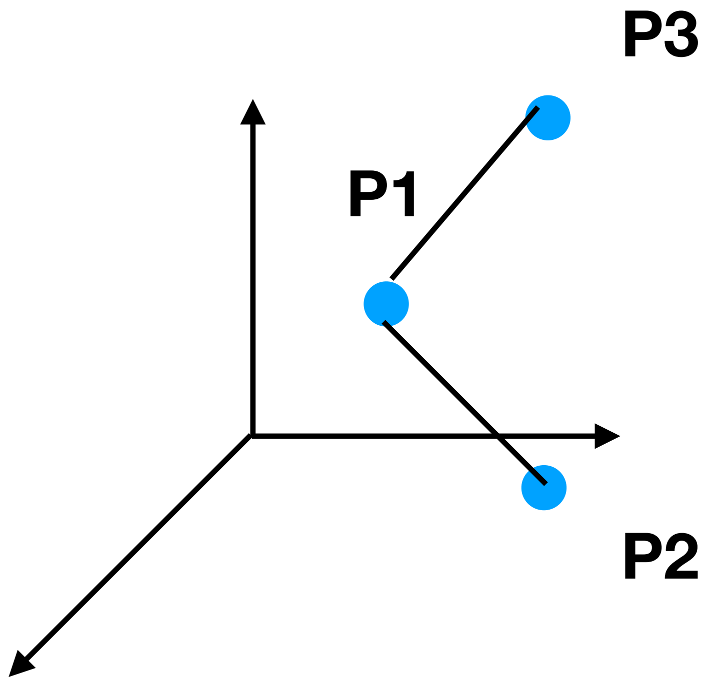
Transformation of Segments



- Step 1: Move P1 to origin



Transformation of Segments

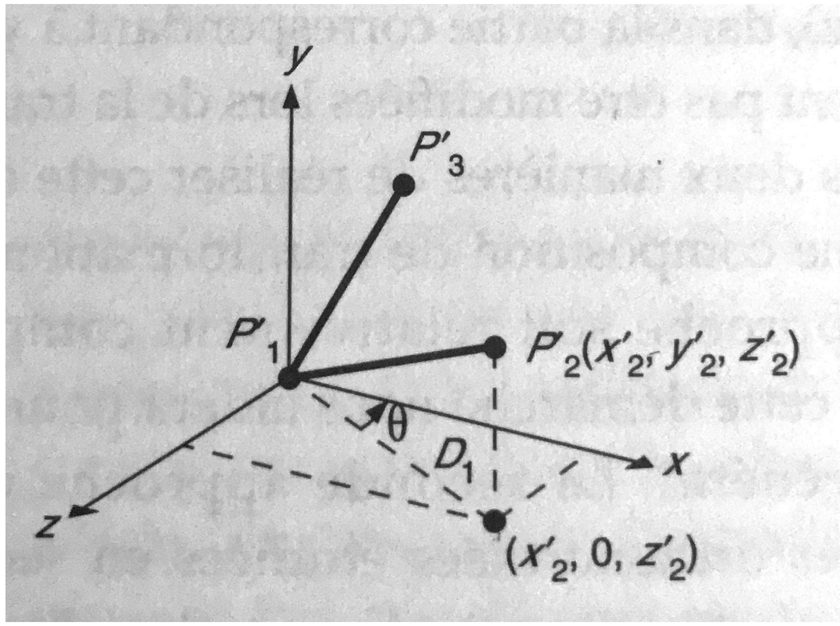


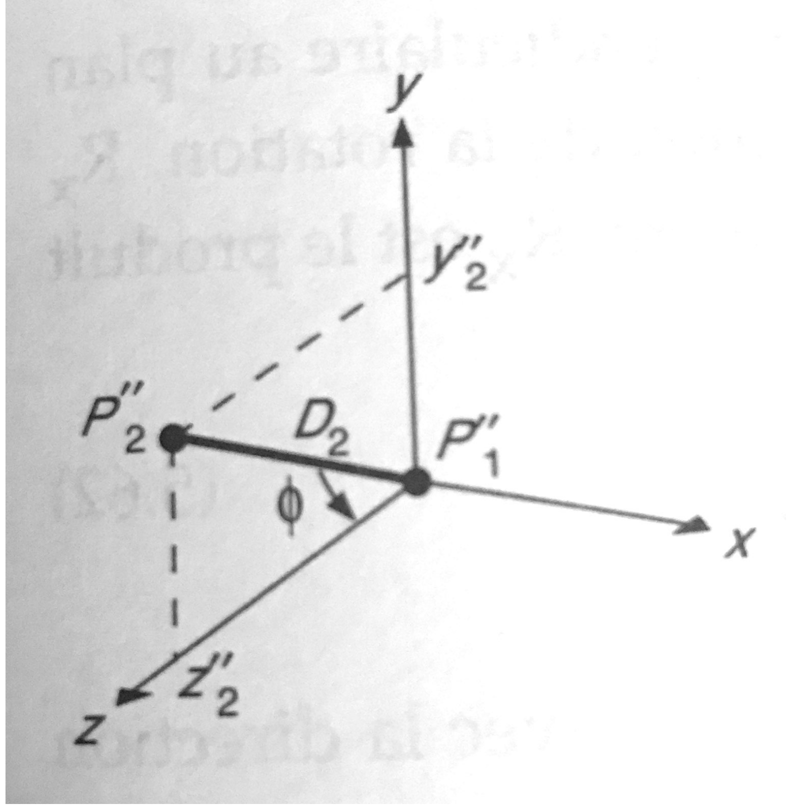
- Step 1: Move P1 to origin

$$T(-x_1, -y_1, -z_1) = \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformation of Segments

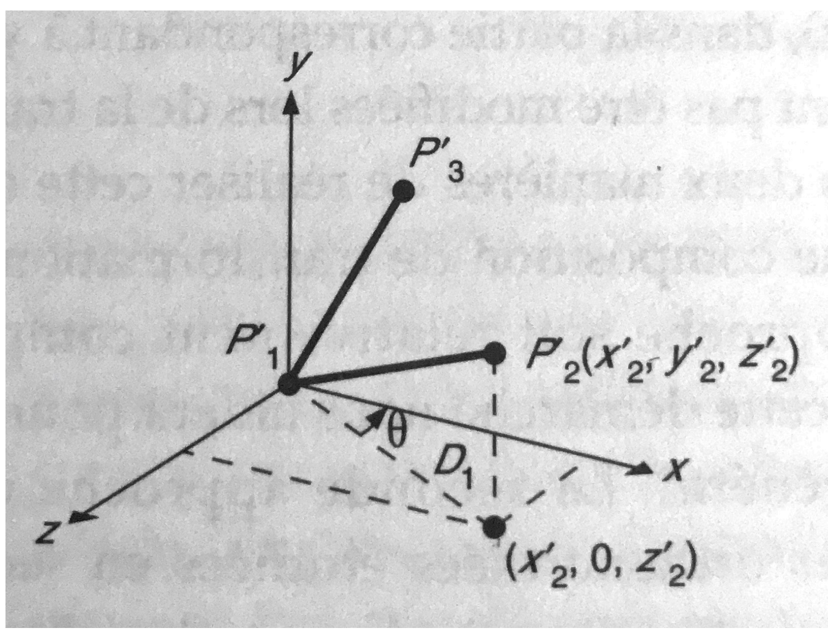
- Step 2: Rotate around y-axis





Transformation of Segments

- **Step 2: Rotate around y-axis** $-(90-\theta)$



$$\cos(\theta-90)=\sin(\theta)=\frac{z_2'}{D_1}=\frac{z_2-z_1}{D_1}$$

$$\sin(\theta-90)=-\cos(\theta)=-\frac{x_2'}{D_1}=-\frac{x_2-x_1}{D_1}$$

avec $D_1=\sqrt{(z_2')^2+(x_2')^2}=\sqrt{(z_2-z_1)^2+(x_2-x_1)^2}$

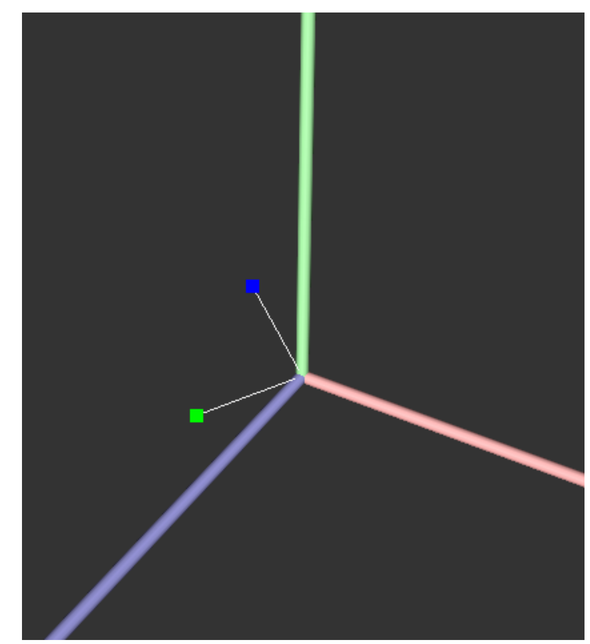
Transformation of Segments

$$\cos(\theta - 90) = \sin(\theta) = \frac{z_2'}{D_1} = \frac{z_2 - z_1}{D_1}$$

$$\sin(\theta - 90) = -\cos(\theta) = -\frac{x_2'}{D_1} = -\frac{x_2 - x_1}{D_1}$$

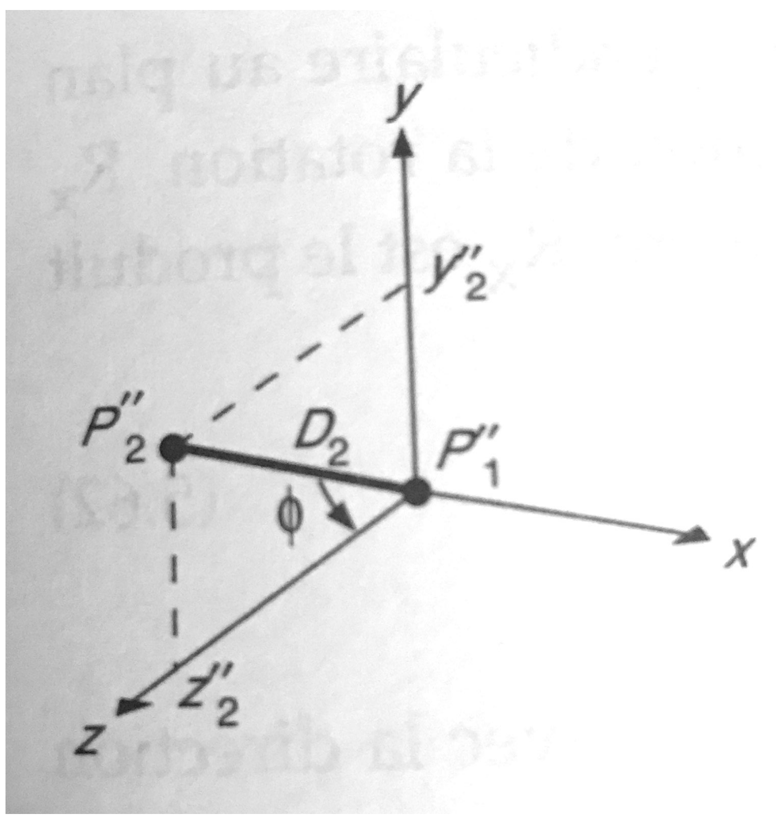
avec $D_1 = \sqrt{(z_2')^2 + (x_2')^2} = \sqrt{(z_2 - z_1)^2 + (x_2 - x_1)^2}$

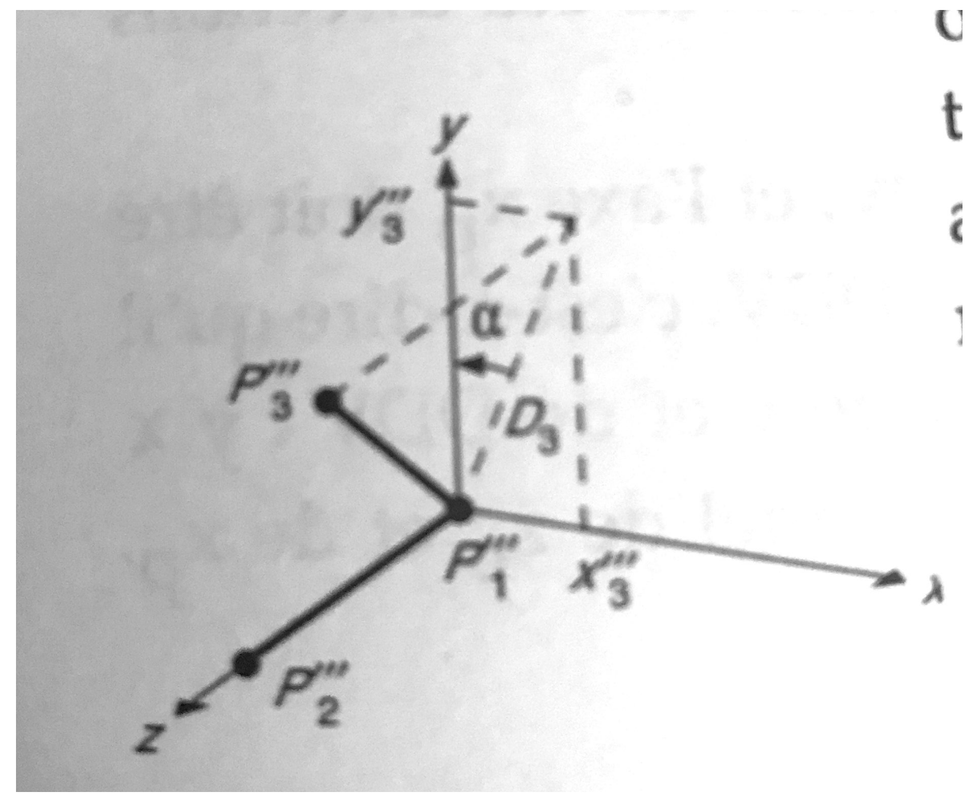
$$R_y(\theta - 90) = \begin{bmatrix} \frac{z_2'}{D_1} & 0 & \frac{-x_2'}{D_1} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{x_2'}{D_1} & 0 & \frac{z_2'}{D_1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Transformation of Segments

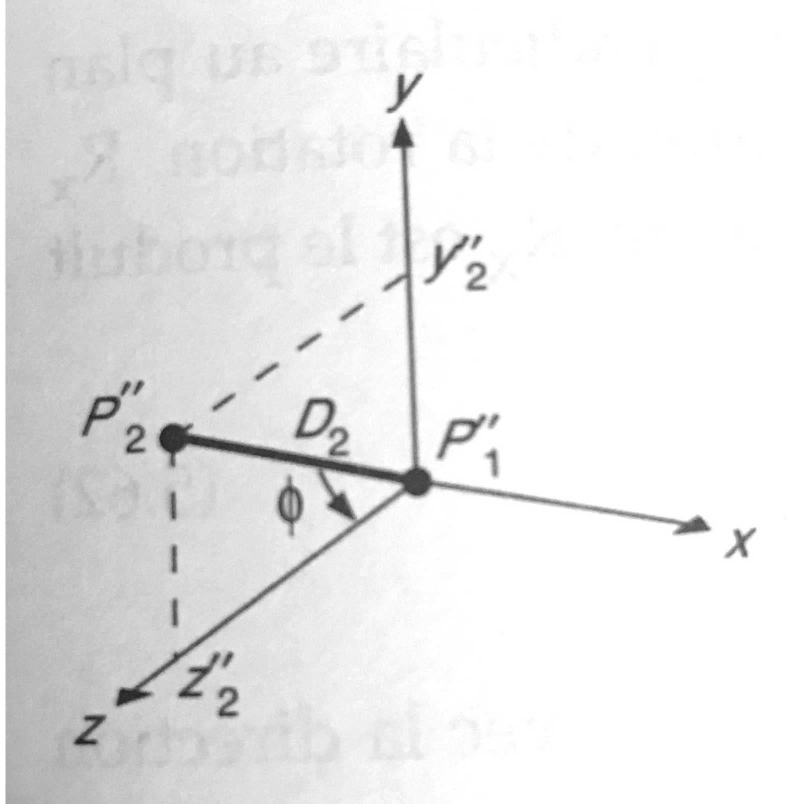
- **Step 3: Rotate around x-axis**





Transformation of Segments

- **Step 3: Rotate around x-axis** ϕ



$$\cos(\phi) = \frac{z_2''}{D_2}$$

$$\sin(\phi) = \frac{y_2''}{D_2}$$

$$D_2 = ||P_1''P_2''|| = ||P_1P_2|| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Transformation of Segments

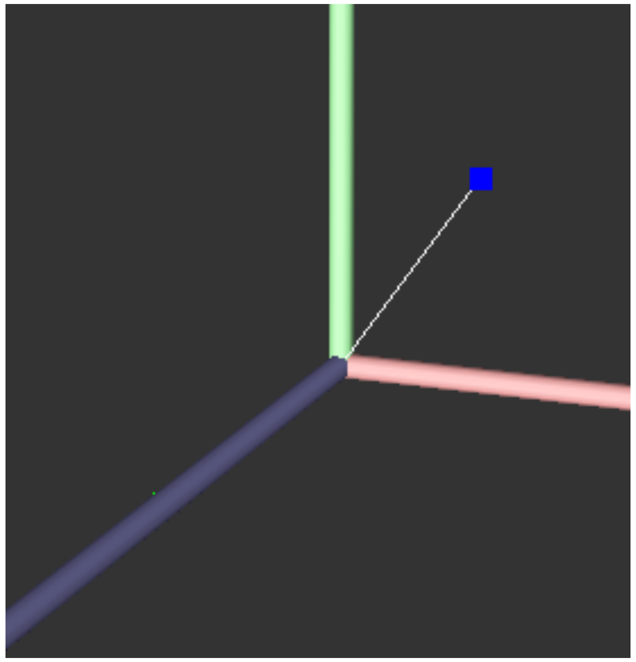
$$\cos(\phi) = \frac{z_2''}{D_2}$$

- **Step 3: Rotate around x-axis**

$$\sin(\phi) = \frac{y_2''}{D_2}$$

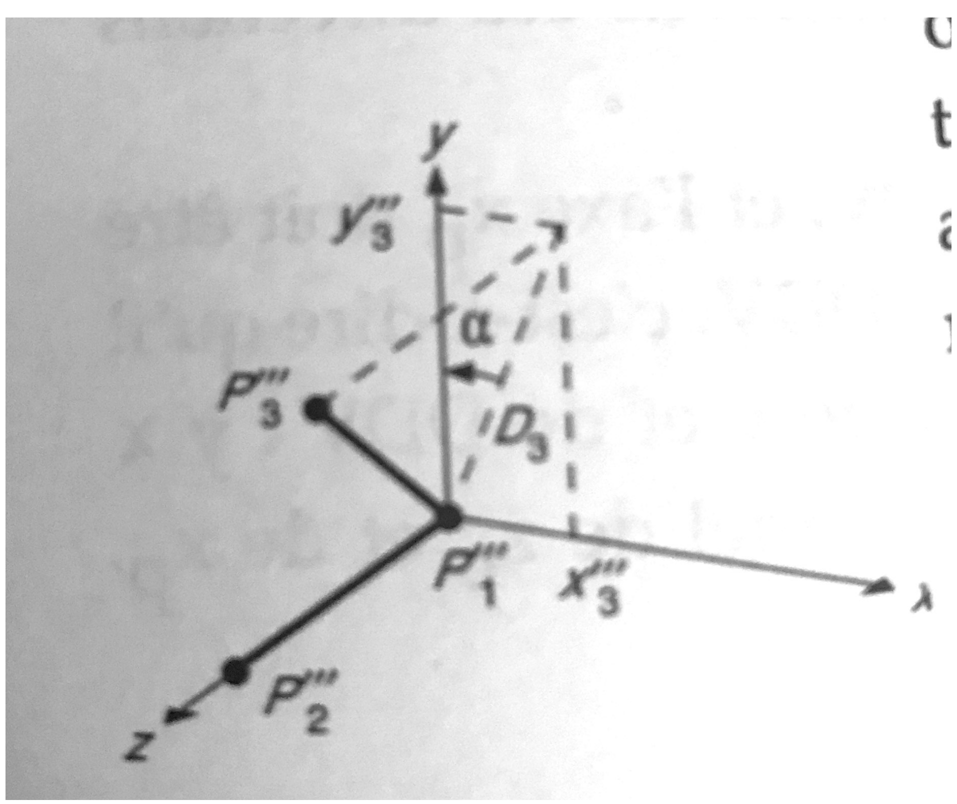
$$D_2 = ||P_1''P_2''|| = ||P_1P_2|| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

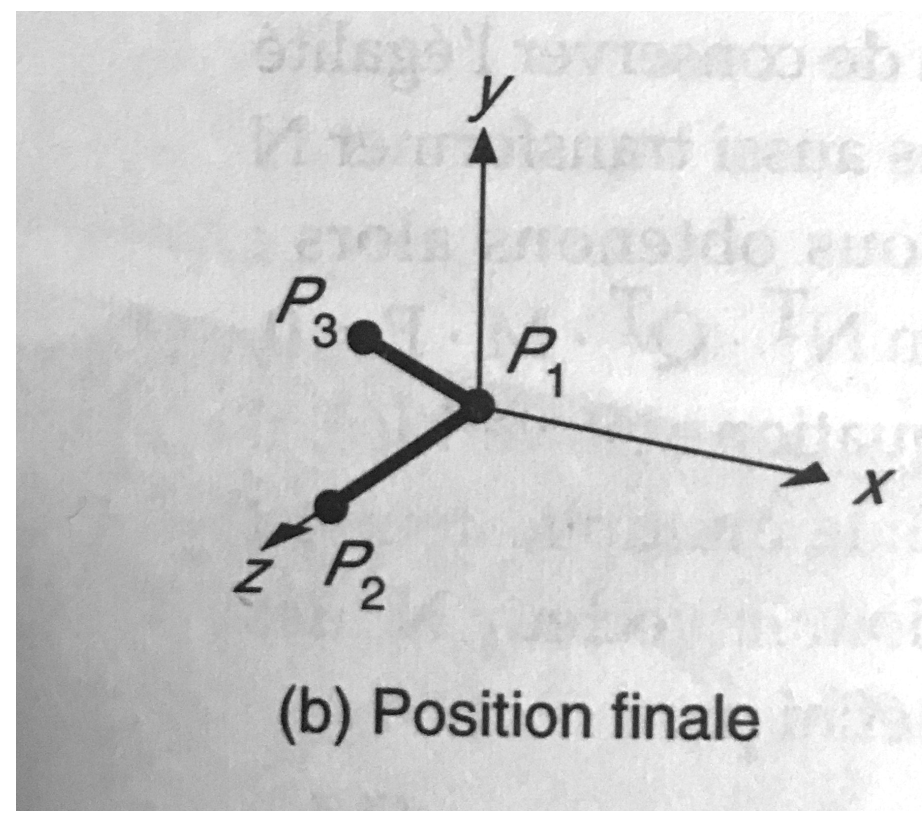
$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{z_2''}{D_2} & -\frac{y_2''}{D_2} & 0 \\ 0 & \frac{y_2''}{D_2} & \frac{z_2''}{D_2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Transformation of Segments

- Step 4 : Rotate around z-axis

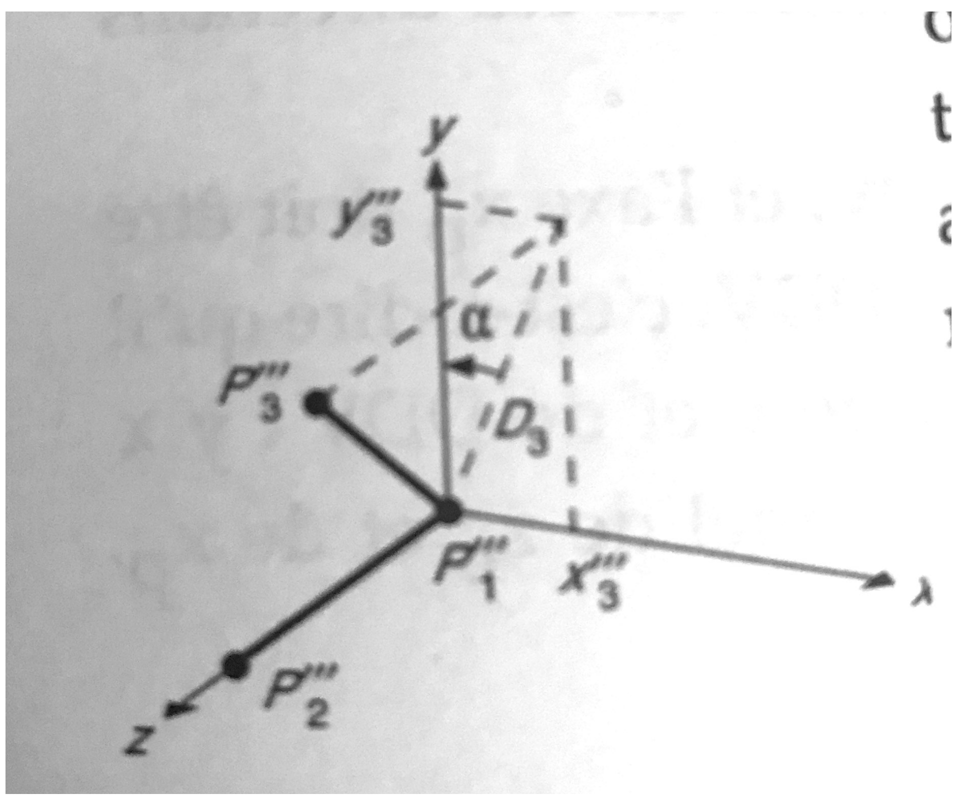




(b) Position finale

Transformation of Segments

- Step 4 : Rotate around z-axis α



$$\cos(\alpha) = \frac{y_3'''}{D_3}$$

$$\sin(\alpha) = \frac{x_3'''}{D_3}$$

$$D_3 = \sqrt{{x_3'''}^2 + {y_3'''}^2}$$

Transformation of Segments

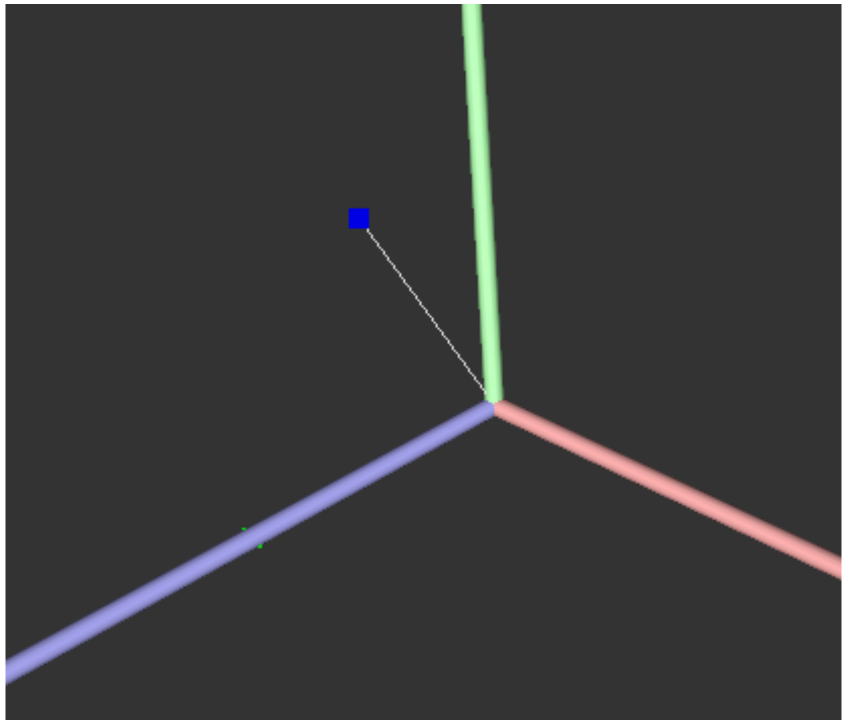
- **Step 4 : Rotate around z-axis**

$$\cos(\alpha) = \frac{y_3'''}{D_3}$$

$$\sin(\alpha) = \frac{x_3'''}{D_3}$$

$$D_3 = \sqrt{x_3'''^2 + y_3'''^2}$$

$$R_z(\alpha) = \begin{bmatrix} \frac{y_3'''}{D_3} & -\frac{x_3'''}{D_3} & 0 & 0 \\ \frac{x_3'''}{D_3} & \frac{y_3'''}{D_3} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Transformation of Segments

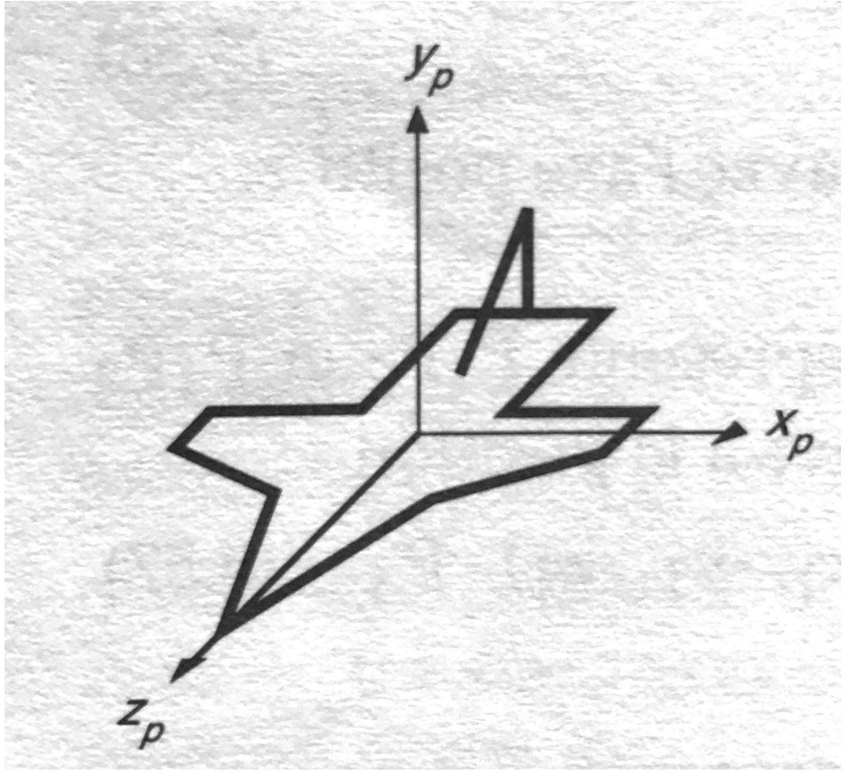
- **Matrice de transformation finale**

$$R_z(\alpha) \, . \, R_x(\phi) \, . \, R_y(\theta - 90) \, . \, T(-x_1, -y_1, -z_1) = R \, . \, T$$

$$\begin{bmatrix} \frac{y_3'''}{D_3} & -\frac{x_3'''}{D_3} & 0 & 0 \\ \frac{x_3'''}{D_3} & \frac{y_3'''}{D_3} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{z_2''}{D_2} & -\frac{y_2''}{D_2} & 0 \\ 0 & \frac{y_2''}{D_2} & \frac{z_2''}{D_2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{z_2'}{D_1} & 0 & \frac{-x_2'}{D_1} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{x_2'}{D_1} & 0 & \frac{z_2'}{D_1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

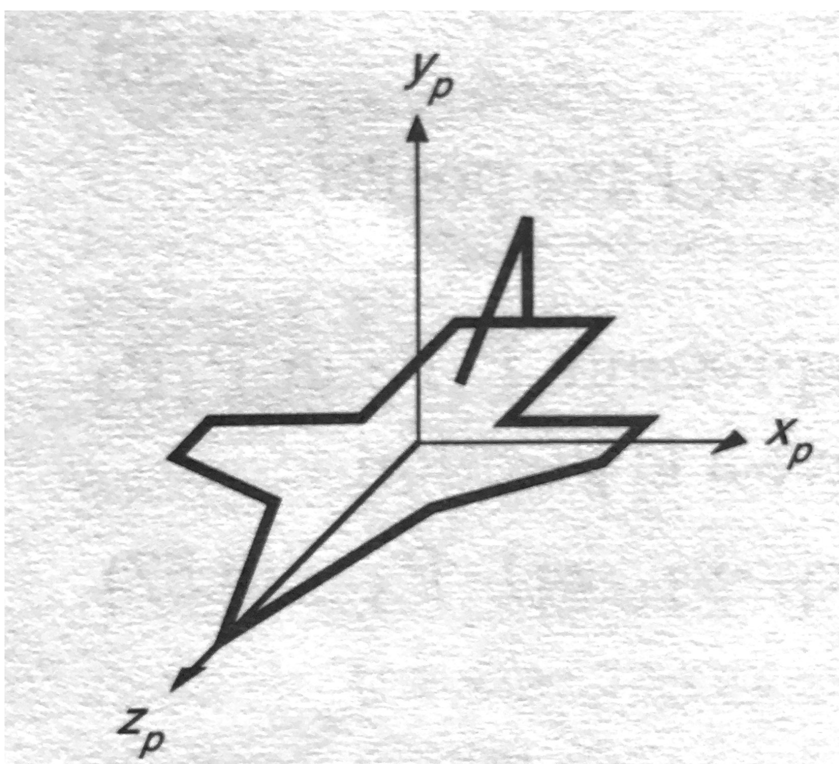
Transformation of Object

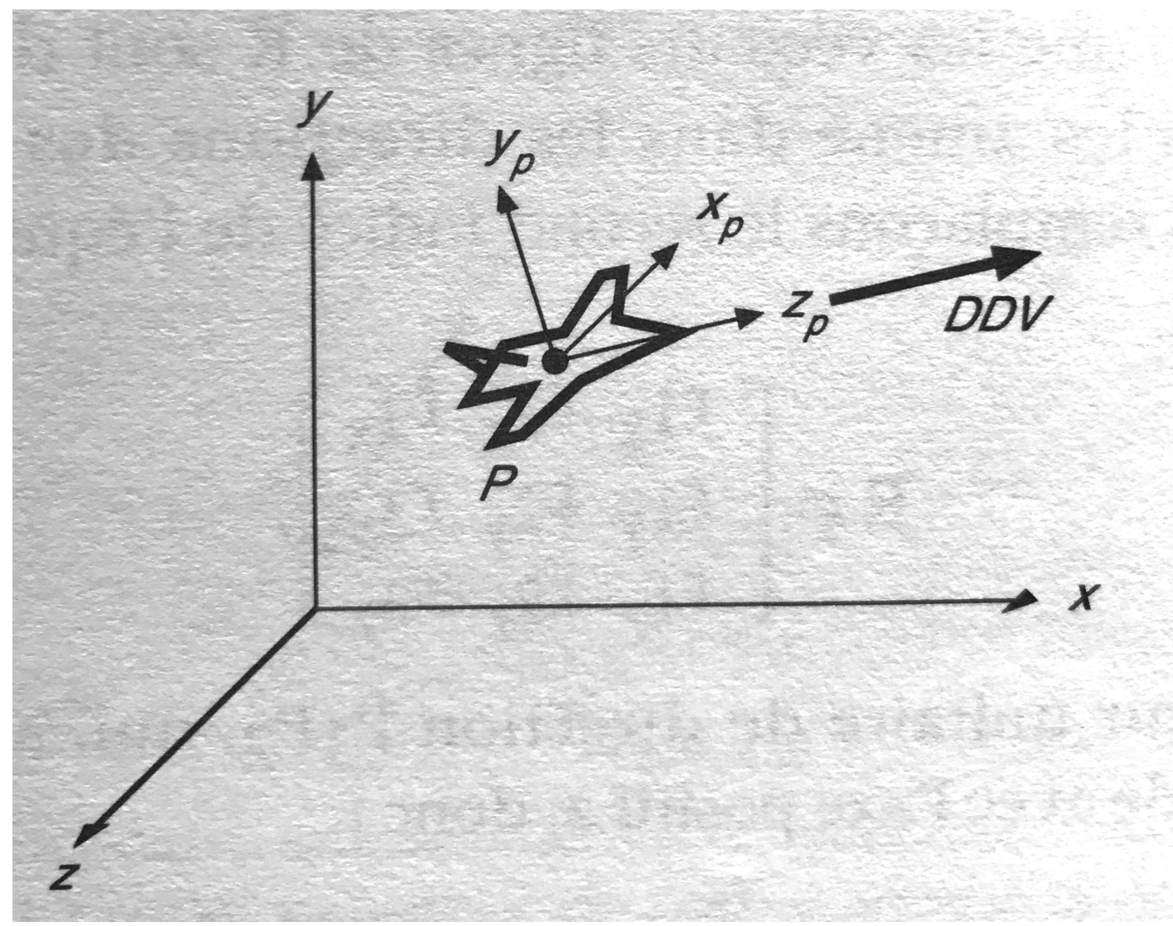
- Another way to look at the problem is to work in local coordinate system and align each axis
  - First column of rotation matrix is x-axis
  - Second column is y-axis
  - Third column is z-axis



Transformation of Object

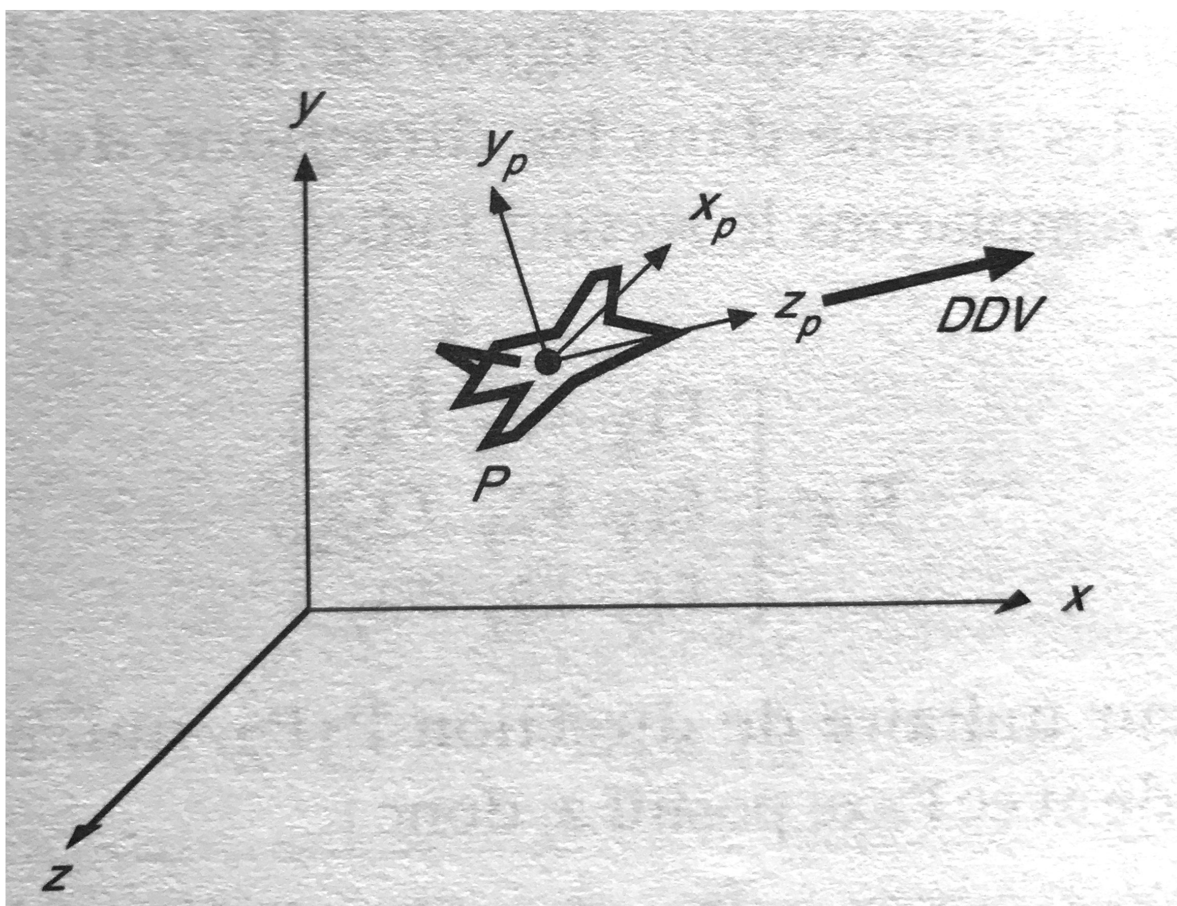
- We want plane to have horizontal wings follow direction DDV





Transformation of Object

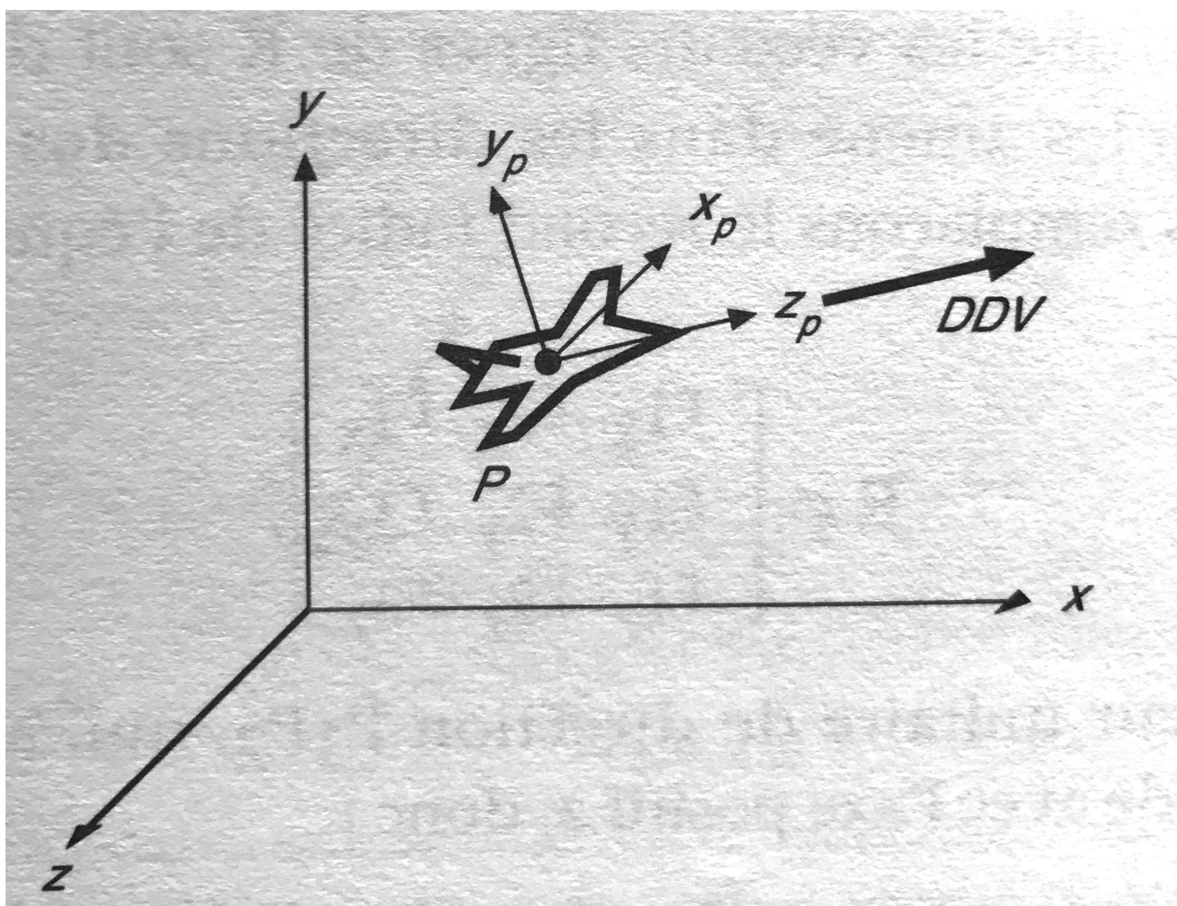
- We want plane to have horizontal wings follow direction DDV



$$\overrightarrow{z_p} = \overrightarrow{DDV}$$

Transformation of Object

- We want plane to have horizontal wings a follow direction DDV



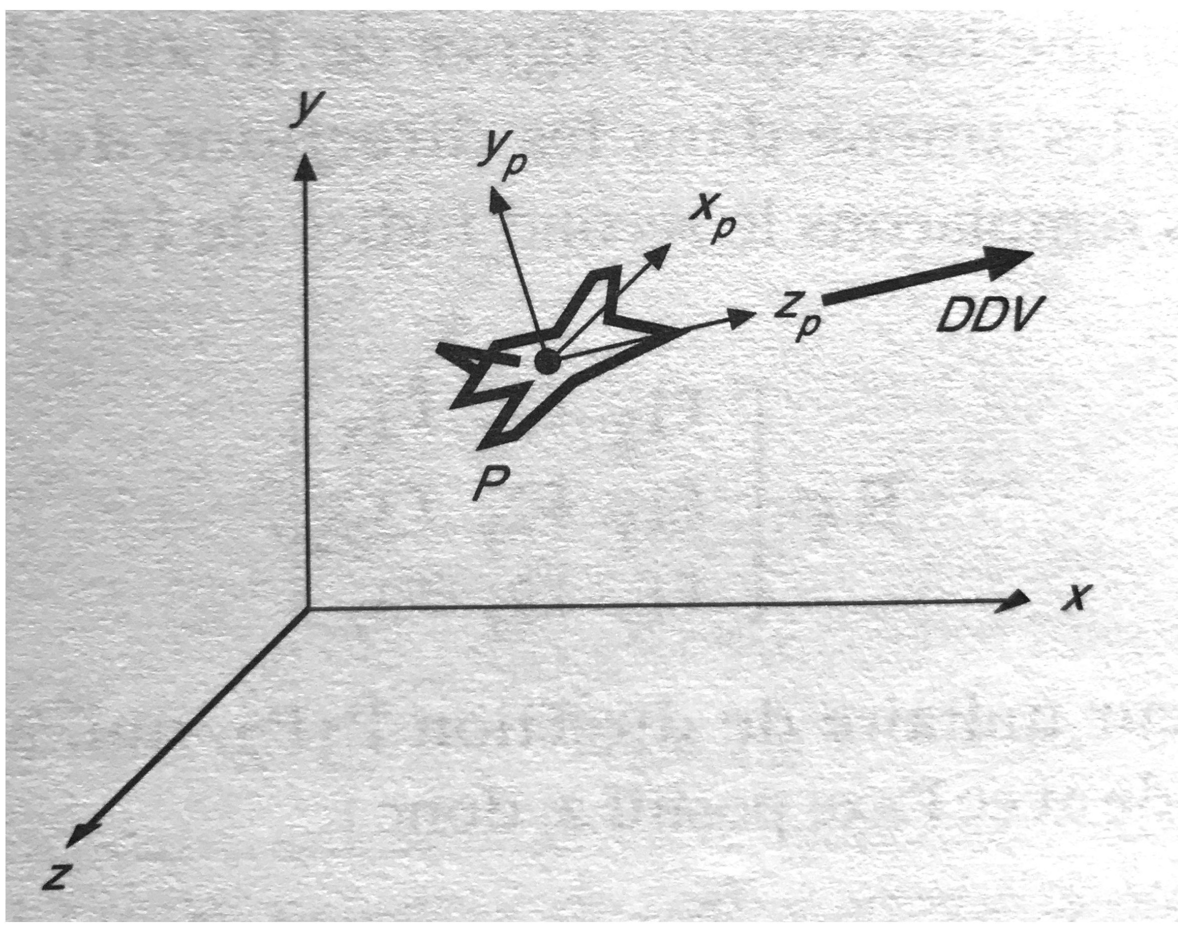
$$\overrightarrow{z_p} = \overrightarrow{DDV}$$

$\overrightarrow{x_p}$ **perpendicular to** $\overrightarrow{z_p}$ **(by definition)**

$\overrightarrow{x_p}$ **perpendicular to** $\overrightarrow{y_{world}}$ **(horizontal)**

Transformation of Object

- We want plane to have horizontal wings a follow direction DDV



$$\overrightarrow{z_p} = \overrightarrow{DDV}$$

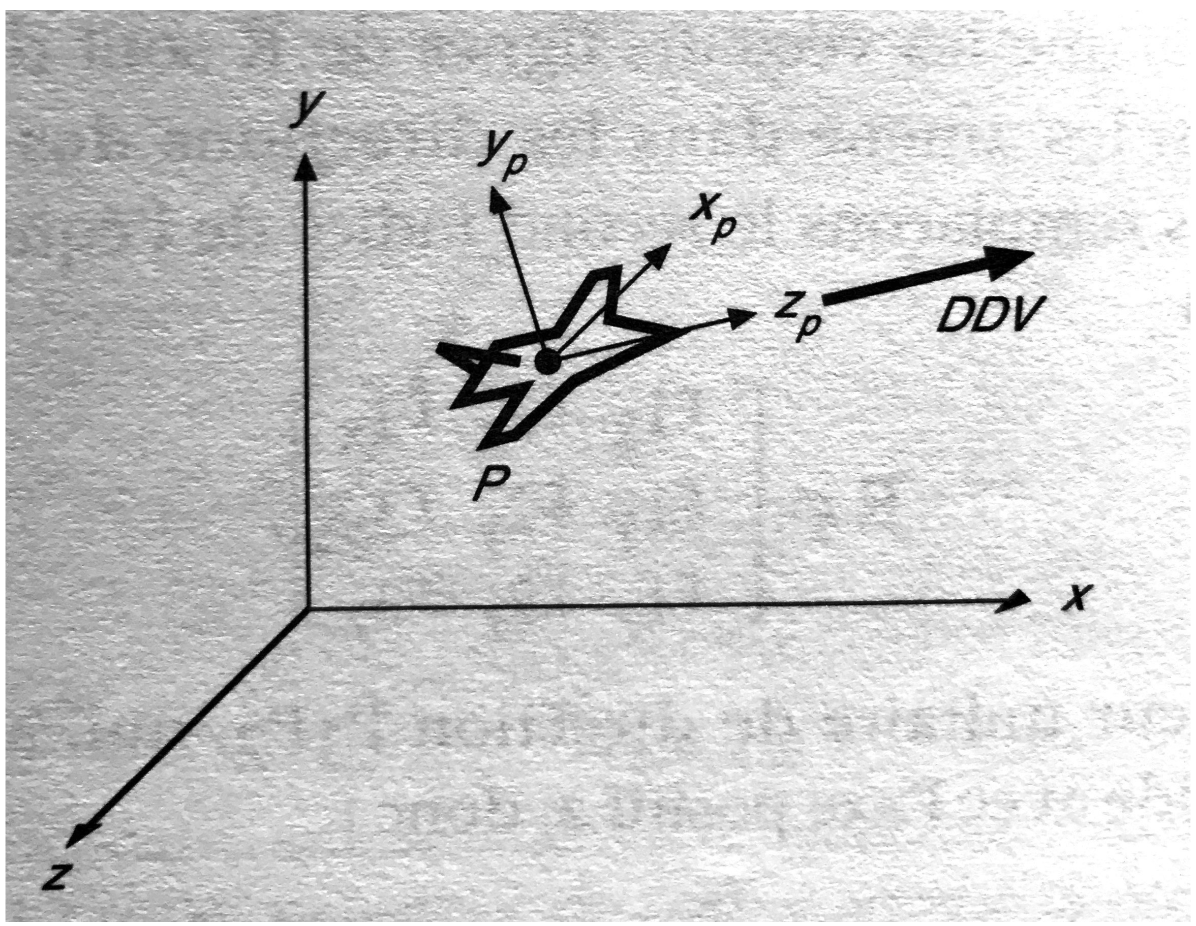
$\overrightarrow{x_p}$ **perpendicular to** $\overrightarrow{z_p}$ **(by definition)**

$\overrightarrow{x_p}$ **perpendicular to** $\overrightarrow{y_{world}}$ **(horizontal)**

$$\overrightarrow{x_p} = \overrightarrow{y} \wedge \overrightarrow{DDV}$$

Transformation of Object

- We want plane to have horizontal wings a follow direction DDV



$$\overrightarrow{z_p} = \overrightarrow{DDV}$$

$$\overrightarrow{x_p} = \overrightarrow{y} \wedge \overrightarrow{DDV}$$

$\overrightarrow{y_p}$ perpendicular to $\overrightarrow{x_p}$ and $\overrightarrow{z_p}$ (by definition)

$$\overrightarrow{y_p} = \overrightarrow{z_p} \wedge \overrightarrow{x_p} = \overrightarrow{DDV} \wedge (\overrightarrow{y} \wedge \overrightarrow{DDV})$$

Transformation of Object

- We want plane to have horizontal wings a follow direction DDV

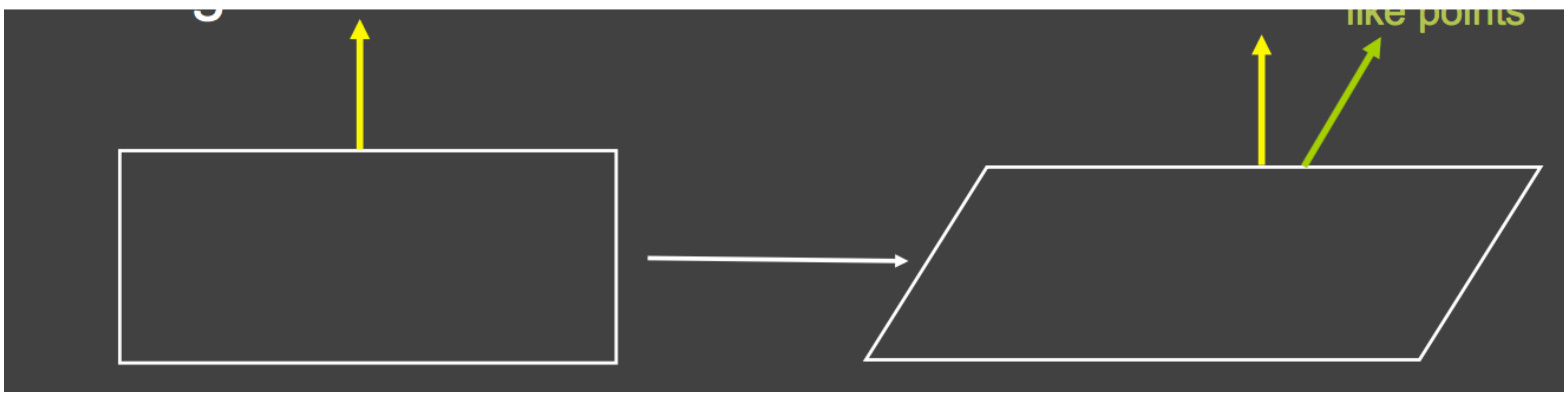
$$\overrightarrow{x_p} = \overrightarrow{y} \wedge \overrightarrow{DDV} \qquad \overrightarrow{y_p} = \overrightarrow{z_p} \wedge \overrightarrow{x_p} = \overrightarrow{DDV} \wedge (\overrightarrow{y} \wedge \overrightarrow{DDV}) \qquad \overrightarrow{z_p} = \overrightarrow{DDV}$$

$$R = \begin{bmatrix} |\overrightarrow{y} \wedge \overrightarrow{DDV}| & |\overrightarrow{DDV} \wedge (\overrightarrow{y} \wedge \overrightarrow{DDV})| & |\overrightarrow{DDV}| & \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Vectors are normalized

Normals

- Important for many tests in graphics like lighting
- Does not transform like points



Normals

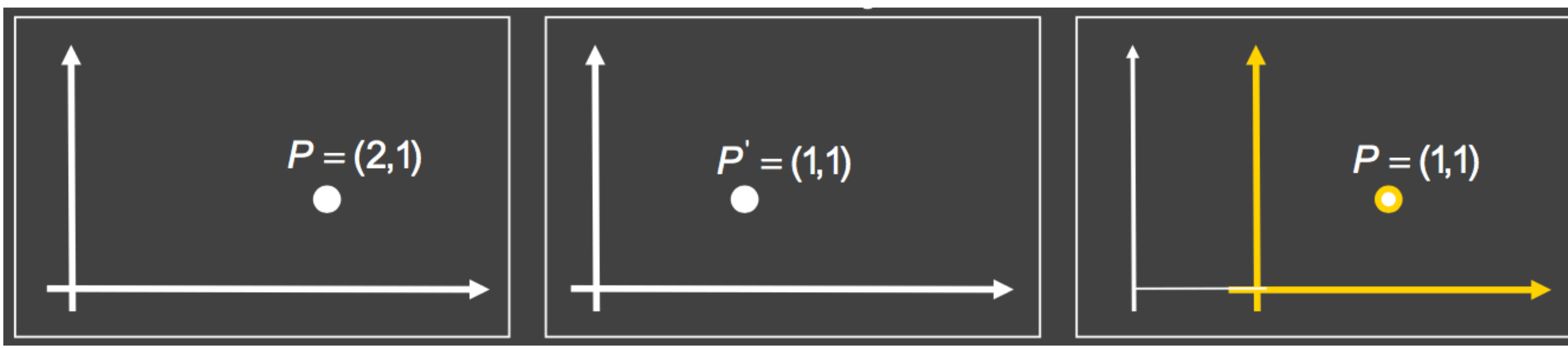
- M, transformation matrix of points
- What is the transformation matrix of normals ?

$$N' = (M^{-1})^T . N$$

Change of Coordinate System

Coordinate Frames

- Moving a point in a coordinate system
- Moving the coordinate system



Coordinate Frames

- Transformation of points in a coordinate frame is the inverse of the transformation from a frame to the other one (leaving points unchanged)