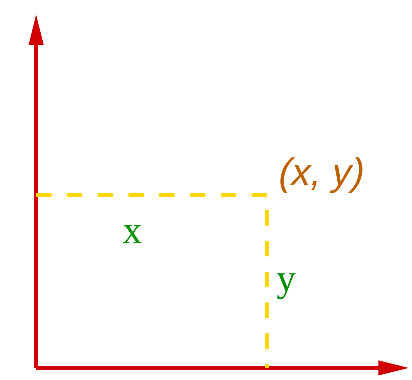
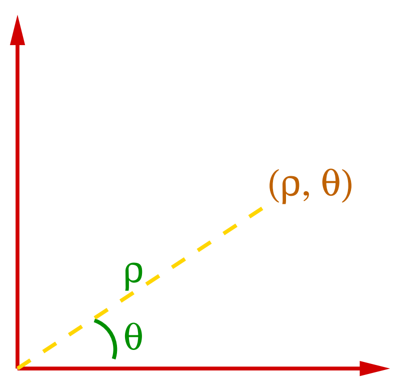
Coordinate Systems

Point Representation in two dimensions

- ☆ Cartesian Coordinates: (x, y)
- ☆ Polar Coordinates: (ρ, θ)



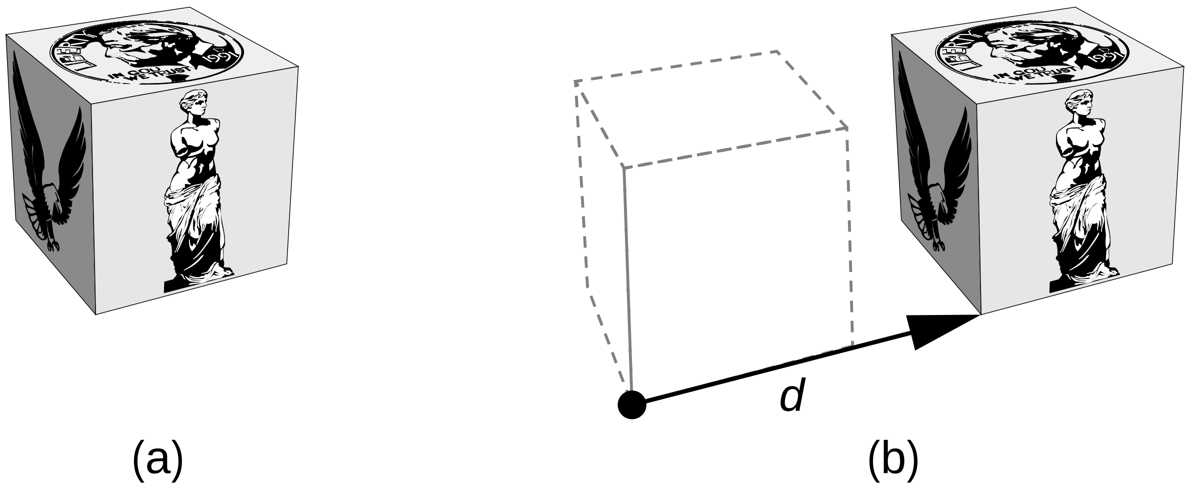
Cartesian Coordinates

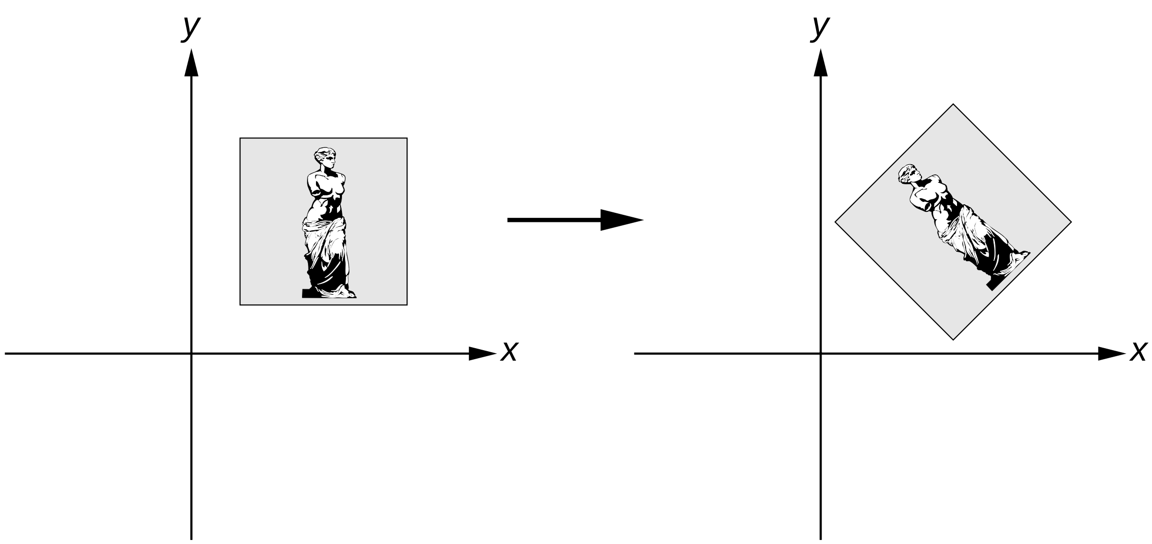


Polar Coordinates

$$p = \begin{bmatrix} x \\ y \end{bmatrix}$$

Transformations





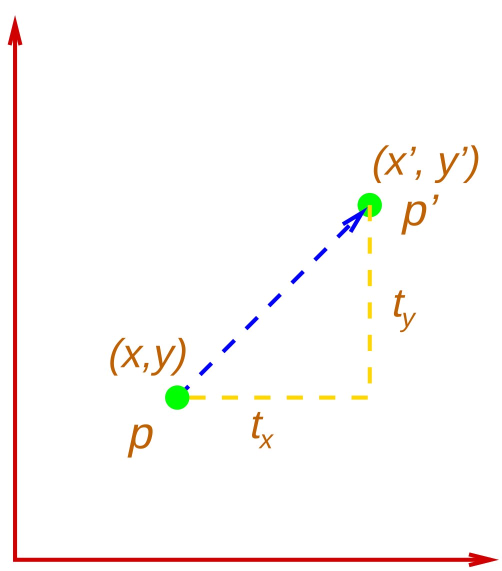
Translation in 2D

Moving the object along a line from one location to another.

$$p = (x, y), t = (t_x, t_y),\ p' = (x', y')$$

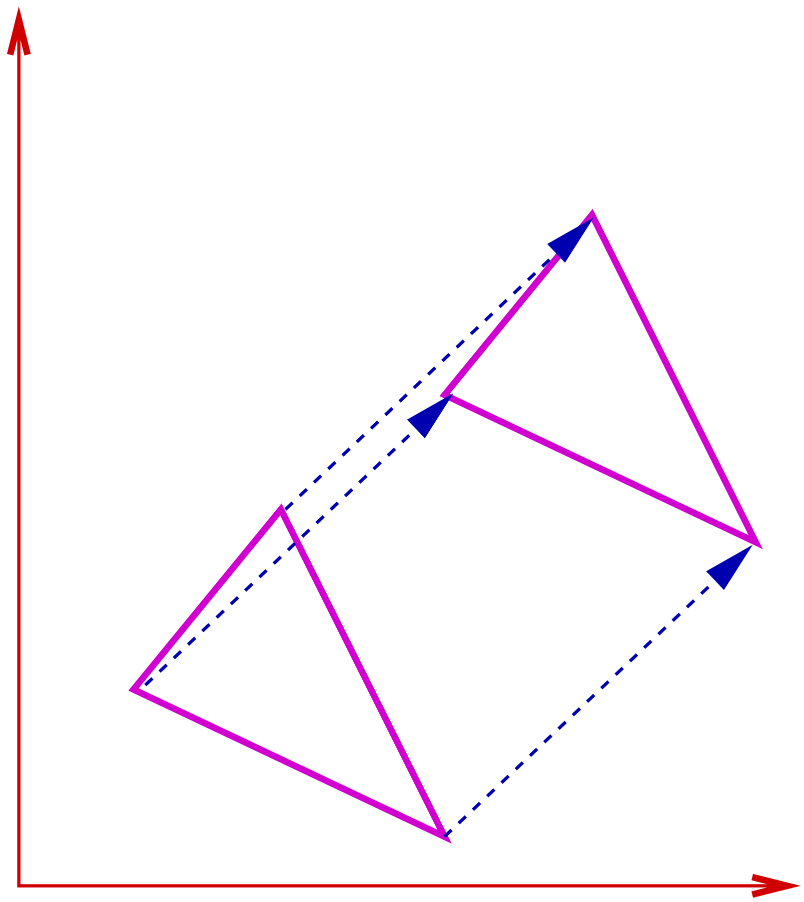
$$x' = x + t_x, \quad y' = y + t_y,\ p' = p + t.$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} t_x \\ t_y \end{bmatrix}}_{T(t)} + \begin{bmatrix} x \\ y \end{bmatrix}$$



Translation in 2D

Translating a rigid body:



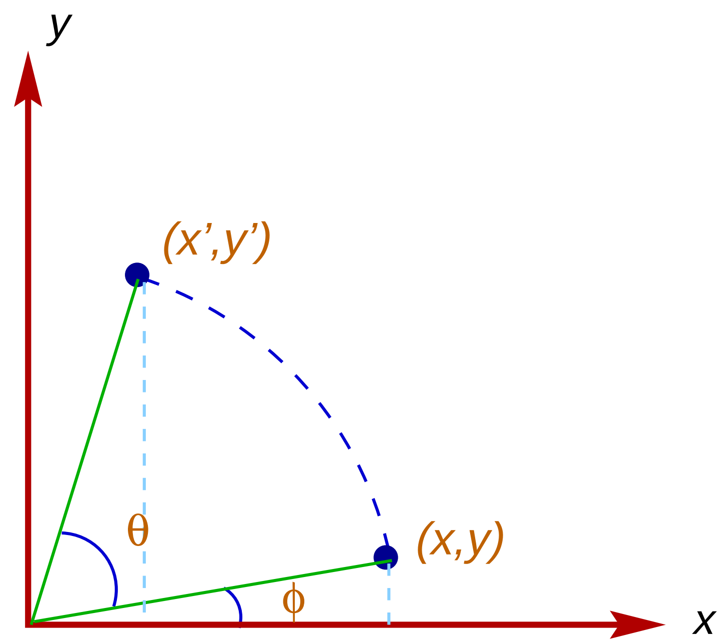
Translate every point of Δ by t.

$$\Delta' = \{p + t \mid p \in \Delta\}.$$

Enough to translate the vertices.

Rotation in 2D

Rotation by θ with respect to origin.



$$x = r\cos\phi, \quad y = r\sin\phi.$$

$$\begin{aligned} x' &= r\cos(\theta+\phi) \\ &= r\cos\theta\cos\phi - r\sin\theta\sin\phi \\ &= x\cos\theta - y\sin\theta. \end{aligned}$$

$$\begin{aligned} y' &= r\sin(\theta+\phi) \\ &= x\sin\theta + y\cos\theta. \end{aligned}$$

Rotation in 2D

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}}_{\mathbf{R}(\theta)} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

$$p' = R \cdot p$$

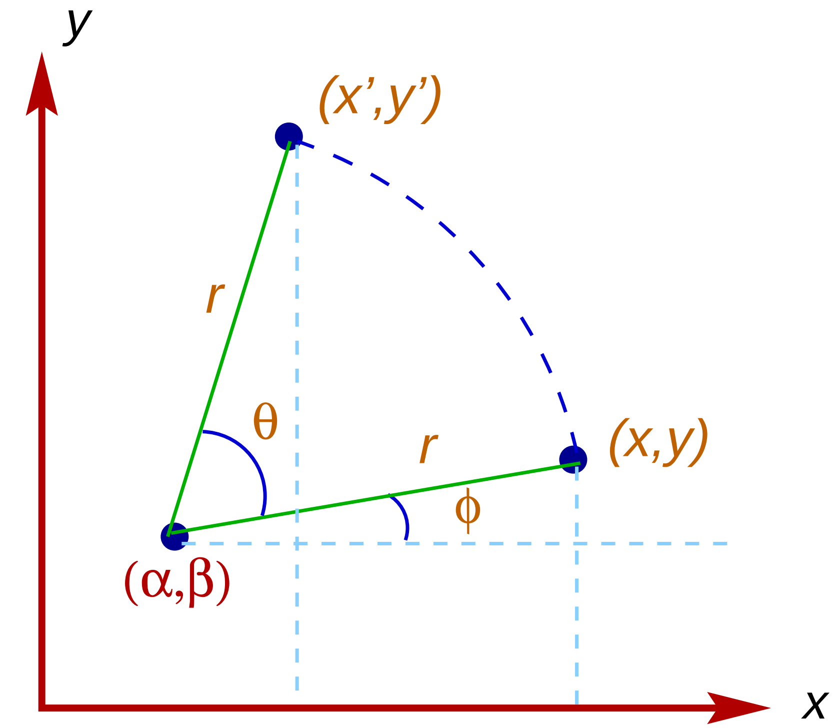
Suppose $R(\theta) = [R_1(\theta)\ R_2(\theta)]$.

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

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

Rotation in 2D

Rotation w.r.t. (α, β):



$$x = \alpha + r\cos\phi \quad \Rightarrow \quad r\cos\phi = x - \alpha$$

$$y = \beta + r\sin\phi \quad \Rightarrow \quad r\sin\phi = y - \beta$$

$$\begin{aligned} x' &= \alpha + r\cos(\theta + \phi) \\ &= \alpha + r\cos\theta\cos\phi - r\sin\theta\sin\phi \\ &= \alpha + (x - \alpha)\cos\theta - (y - \beta)\sin\theta \end{aligned}$$

$$y' = \beta + (x - \alpha)\sin\theta + (y - \beta)\cos\theta.$$

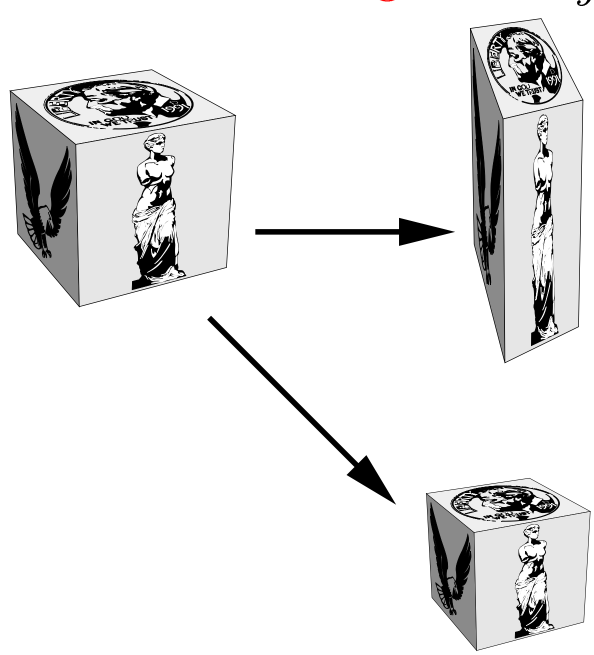
Scaling in 2D

Scale by factor (s_x, s_y)

$$x' = x \cdot s_x \qquad y' = y \cdot s_y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}}_{\mathbf{S(x,y)}} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Uniform scaling $s_x = s_y$



Matrix Representation of Transforms

$$P' = AP + B$$

Translation by (t_x, t_y):

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Rotation by θ:

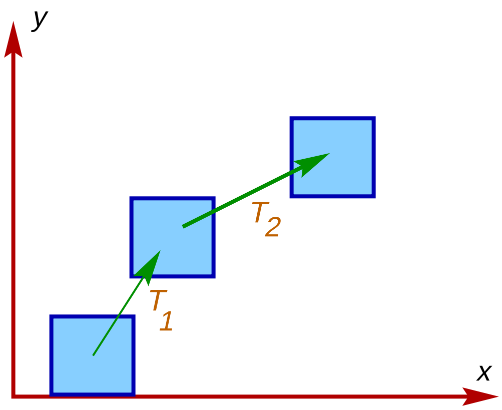
$$A = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Scaling by (s_x, s_y):

$$A = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

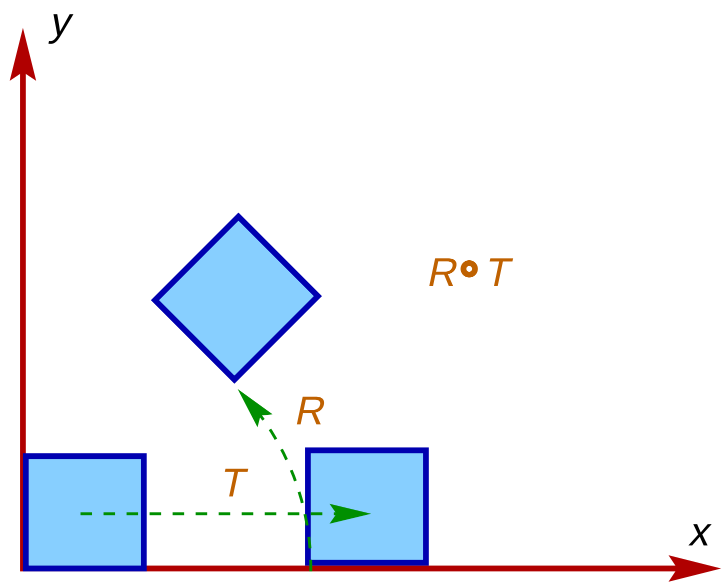
Composite Transformation

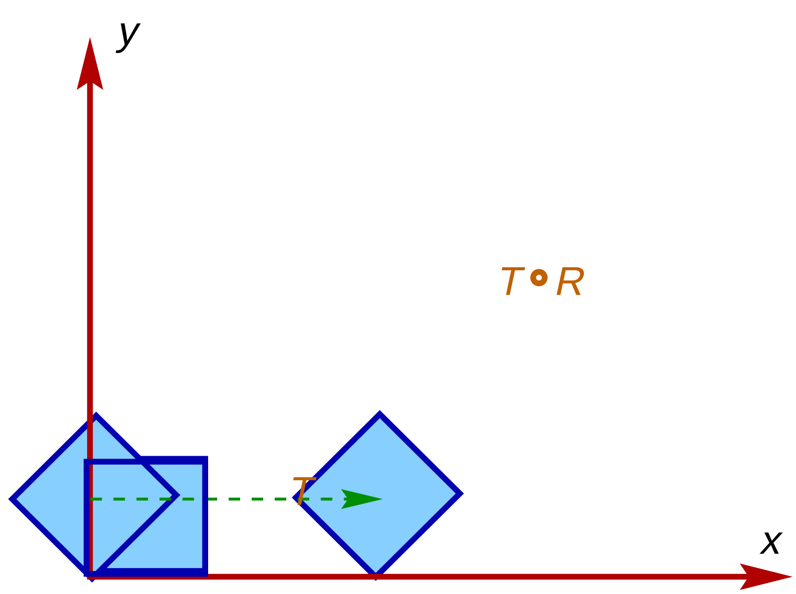
$$T_1 = (A_1, B_1)$$
$$T_2 = (A_2, B_2)$$
$$T_2 \cdot T_1(P) = A_2 \cdot (A_1 P + B_1) + B_2$$



Noncommutativity of Transforms

T: Translate $(5, 0)$; R: Rotate by $\pi/4$





Homogeneous Coordinates

Homogeneous coordinates unify translation, rotation, and scaling in one transformation matrix.

Transformation matrix is 3×3 matrix.

Translation

$$\begin{aligned} p &= (x, y, w) \rightarrow \left(\frac{x}{w}, \frac{y}{w}\right) \\ t &= (t_x, t_y) \\ p' &= p + t = \left(\frac{x}{w} + t_x, \frac{y}{w} + t_y\right) \\ p' &= (x + t_x \cdot w,\ y + t_y \cdot w,\ w) \end{aligned}$$

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T(t_x, t_y)}} \cdot \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$$T(t_x, t_y)^{-1} = T(-t_x, -t_y)$$

Homogeneous Coordinates

Rotation

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{R}(\theta)} \cdot \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$$R(\theta)^{-1} = R(-\theta)$$

Scaling

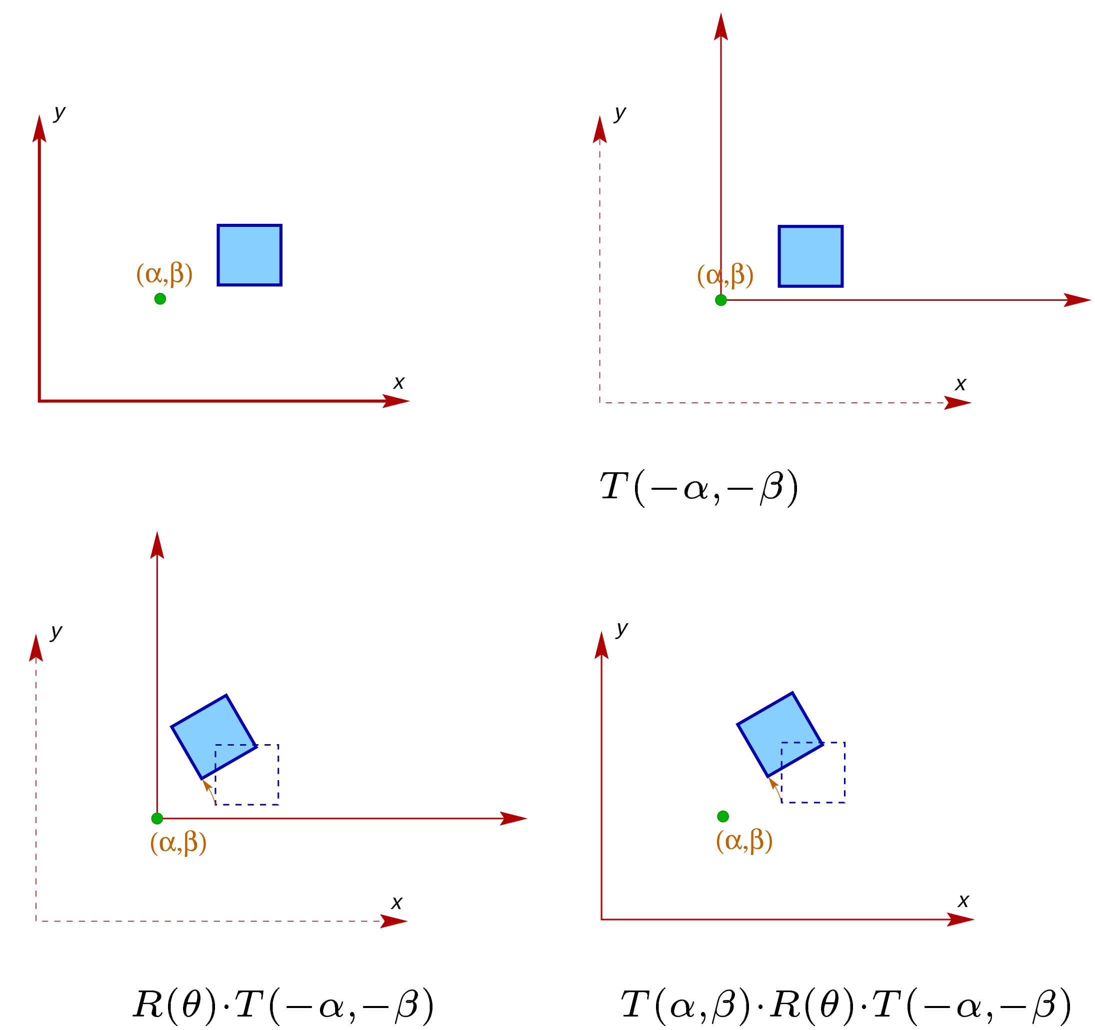
$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \underbrace{\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{S}(\mathbf{s_x},\mathbf{s_y})} \cdot \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$$S(s_x, s_y)^{-1} = S(1/s_x, 1/s_y)$$

2D Rotation revisited

Rotation w.r.t a point q (α, β) by angle θ

Regard as a composite transform.



2D Rotation

- ☆ Translate by $T(-\alpha, -\beta)$ so that q becomes the origin.
- ☆ Rotate by θ with respect to the origin; $R(\theta)$.
- ☆ Translate by $T(\alpha, \beta)$ to bring q back to its original position.

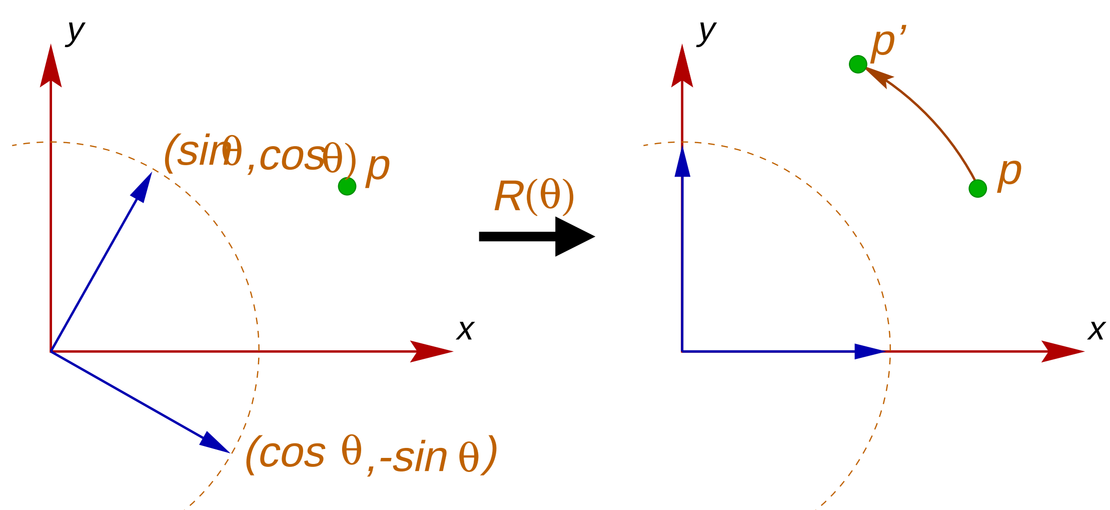
$$R(\theta, \alpha, \beta) = T(\alpha, \beta) \cdot R(\theta) \cdot T(-\alpha, -\beta).$$

2D Rotation: Few Remarks

- ☆ Rotation involves either trigonometric functions or $\sqrt{\ }$
  Use Power series to approximate rotation for small values of θ

- ☆ Consider the upper-left 2×2 matrix of $R(\theta)$ and regard the rows as vectors in the plane

$$\begin{aligned} v_1 &= (\cos\theta, -\sin\theta), \\ v_2 &= (\sin\theta, \cos\theta). \end{aligned}$$

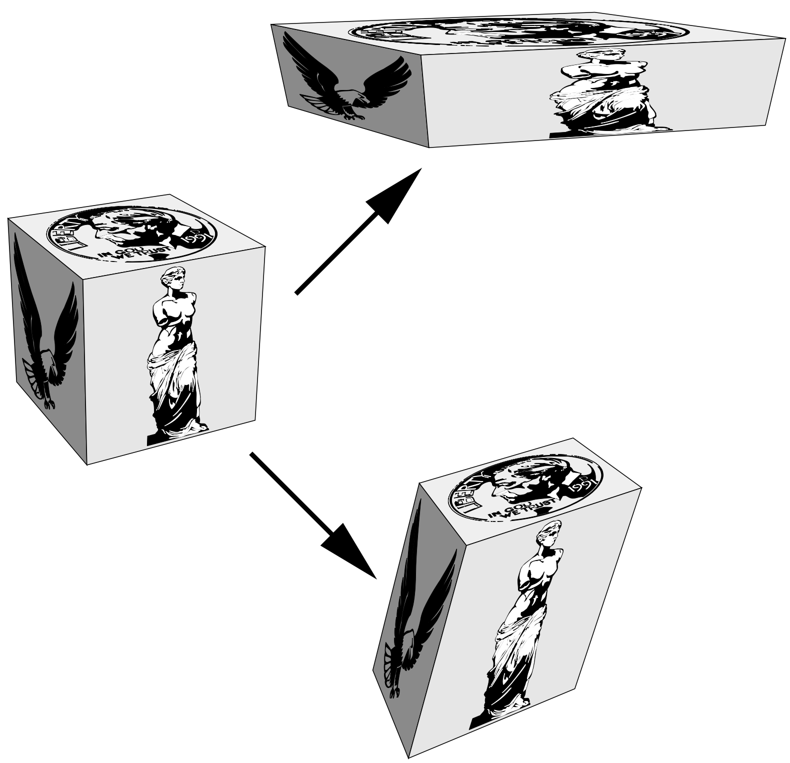


Euclidean Transform

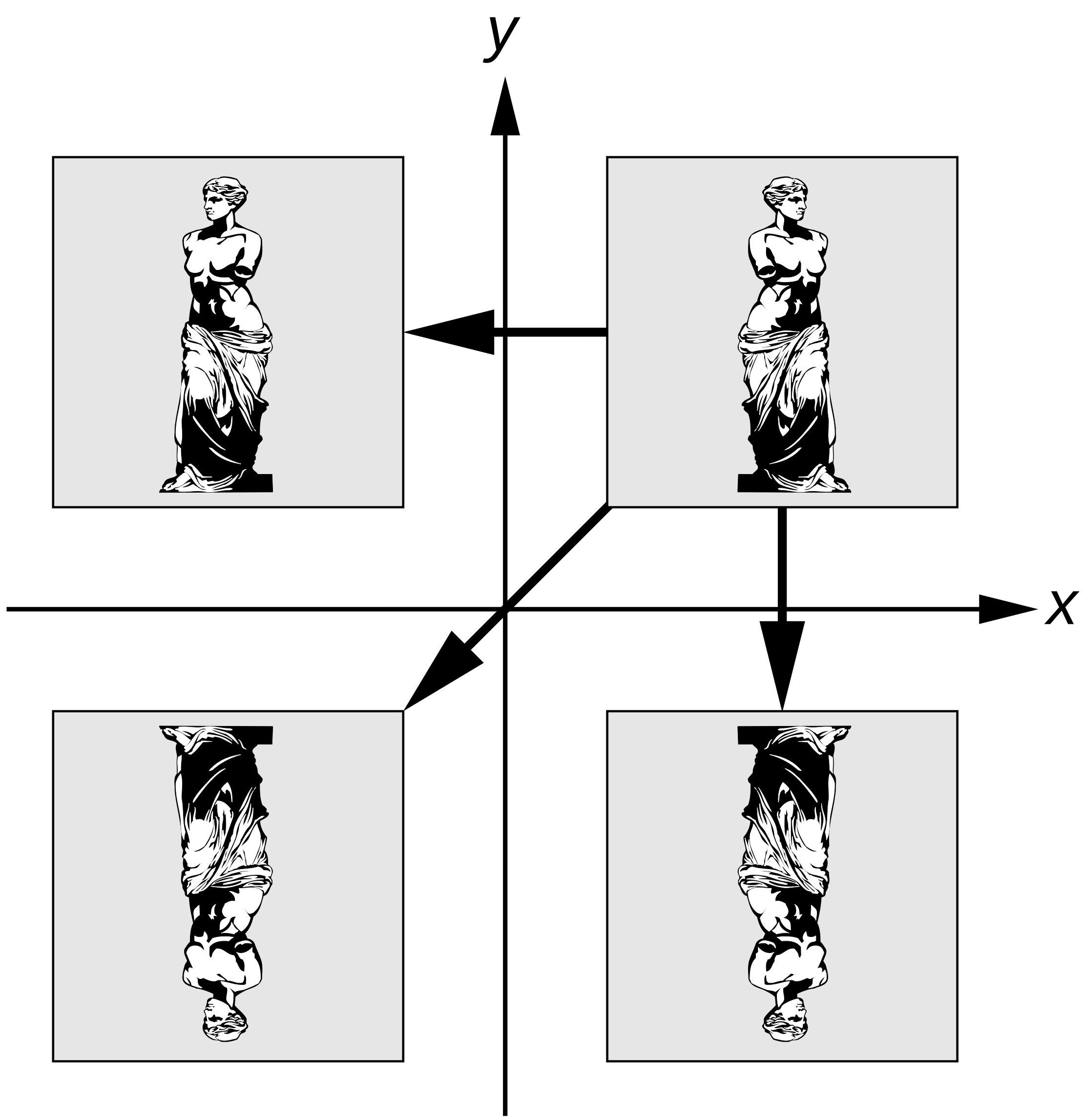
Euclidean transformation

- ☆ Rigid Body transformations
- ☆ Translation and Rotation
- ☆ Angles and distances are unchanged

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



Reflection



w.r.t x-axis

$S(1, -1)$

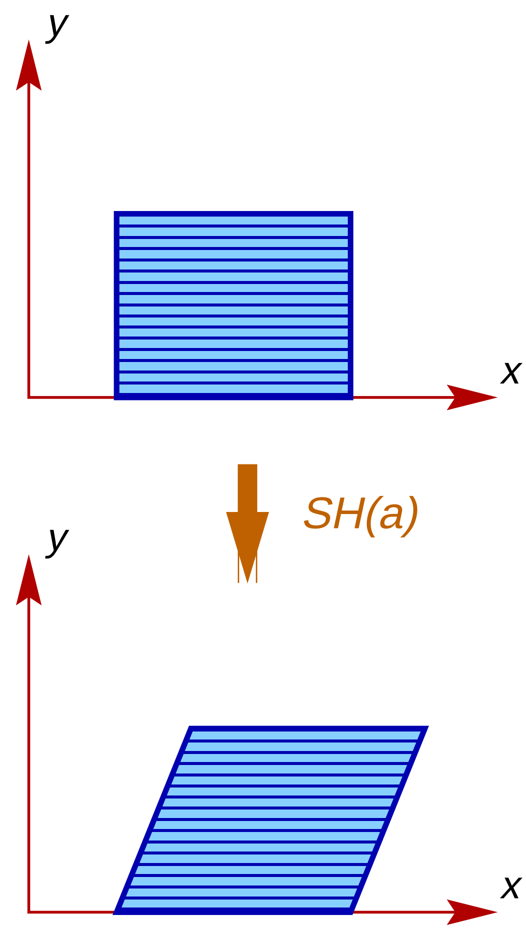
w.r.t y-axis

$S(-1, 1)$

What about w.r.t the line $x = -y$?

Shear Transform

$$SH_x(a) = \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$x' = x + ay, \quad y' = y.$$

Horizontal segment at height y is shifted in the x-direction by ay.

Not a Euclidean transform.

3D Transforms

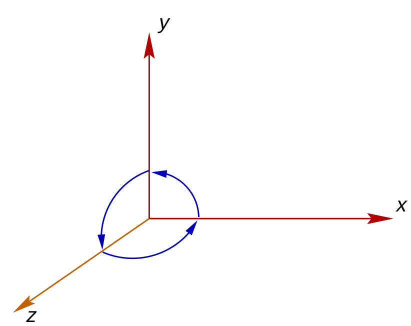
Homogeneous coordinates

$$(x, y, z, w) \Longrightarrow \left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w}\right).$$

Right Hand System:

Positive rotation defined as:

- ☆ Look toward origin from a positive axis,
- ☆ $\pi/2$ counter-clockwise rotation transforms one axis to another.



$x : y \to z, \quad y : z \to x, \quad z : x \to y.$

Some graphics systems use the left hand system.

Translation

Translation by (t_x, t_y, t_z)

$$
\begin{aligned}
p &= (x, y, z, w), \quad t = (t_x, t_y, t_z) \\
p' &= (x', y', z', w') \\
&= (x + w \cdot t_x, y + w \cdot t_y, z + w \cdot t_z, w).
\end{aligned}
$$

$$
\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T(t_x, t_y, t_z)}} \cdot \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}
$$

$$
T(t_x, t_y, t_z)^{-1} = T(-t_x, -t_y, -t_z)
$$

Scaling

Scaling by (s_x, s_y, s_z)

$$\begin{aligned} p &= (x, y, z, w) \quad s = (s_x, s_y, s_z) \\ p' &= (x', y', z', w') = (x \cdot s_x, y \cdot s_y, z \cdot s_z, w). \end{aligned}$$

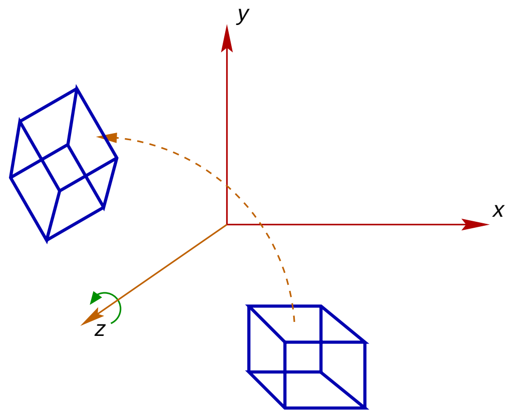
$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \underbrace{\begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{S(s_x, s_y, s_z)}} \cdot \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

$$S(s_x, s_y, s_z)^{-1} = S(1/s_x, 1/s_y, 1/s_z)$$

Rotation in 3D

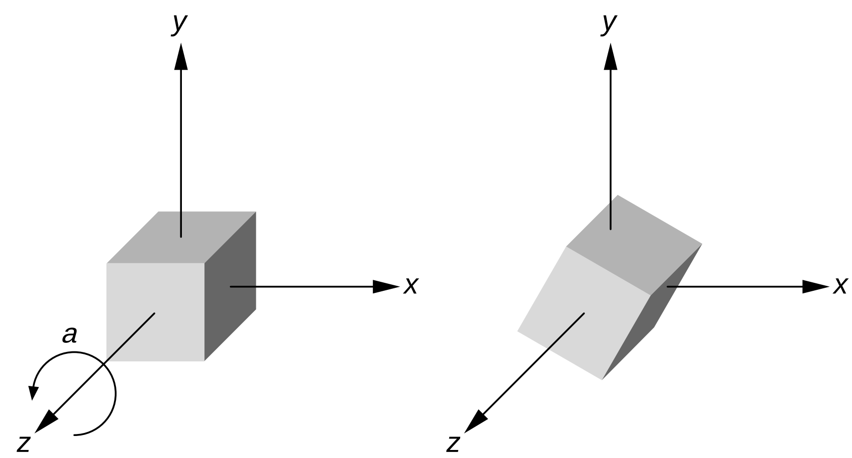
Rotation w.r.t. a line: Axis of rotation

Axis of rotation: z-axis



$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta$$
$$z' = z$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{R_z}(\theta)} \cdot \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

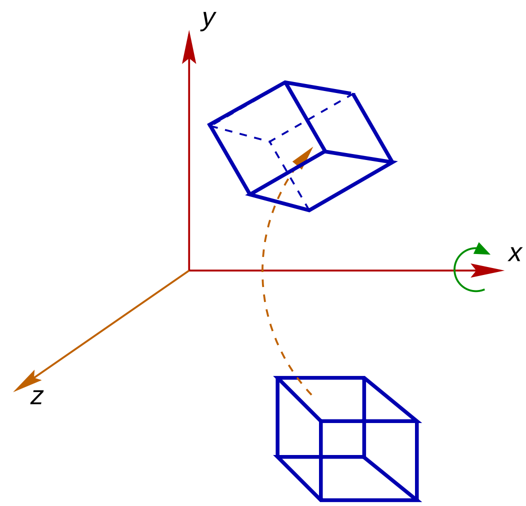


(a) (b)

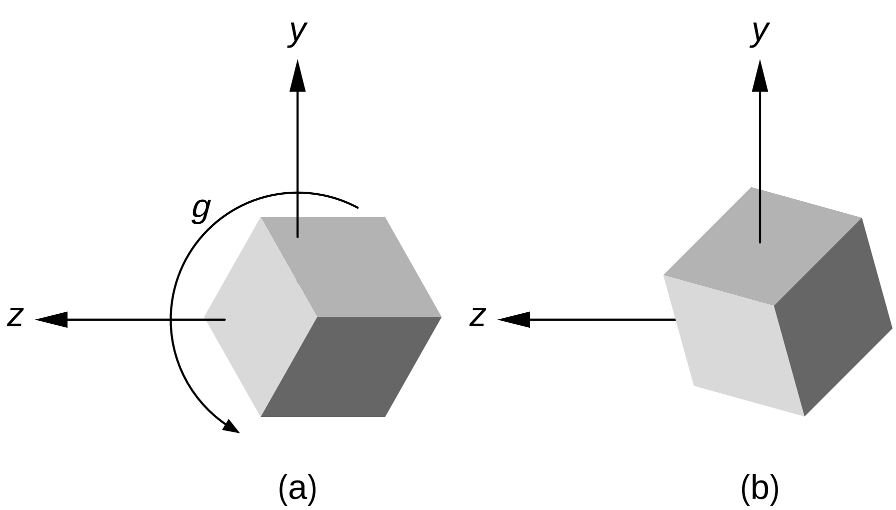
Rotation w.r.t. x-axis

Substitute $x \to y, y \to z, z \to x$.

$$
\begin{aligned}
y' &= y\cos\theta - z\sin\theta \\
z' &= y\sin\theta + z\cos\theta \\
x' &= x
\end{aligned}
$$



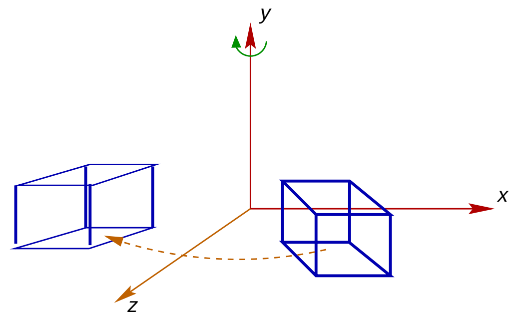
$$
\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{R_x}(\theta)} \cdot \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}
$$



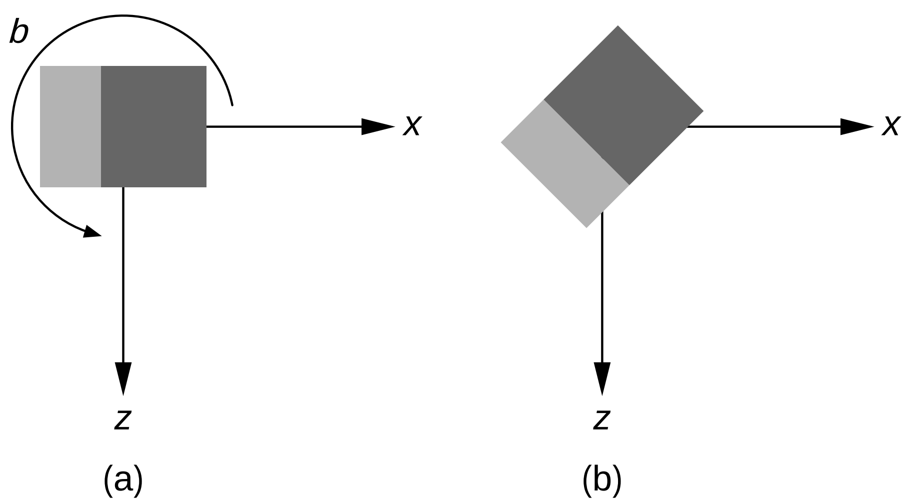
Rotation w.r.t. y-axis

Substitute $x \to y, y \to z, z \to x$ in the previous equations.

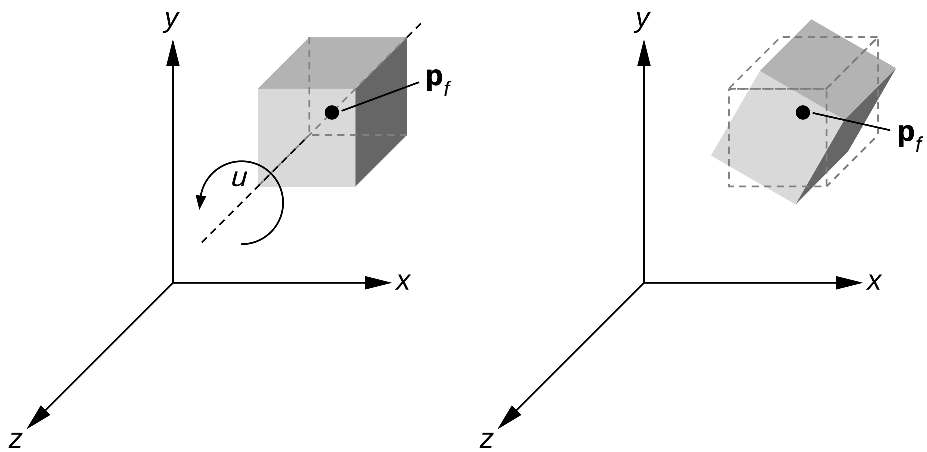
$$
\begin{aligned}
z' &= z\cos\theta - x\sin\theta \\
x' &= z\sin\theta + x\cos\theta \\
y' &= y
\end{aligned}
$$



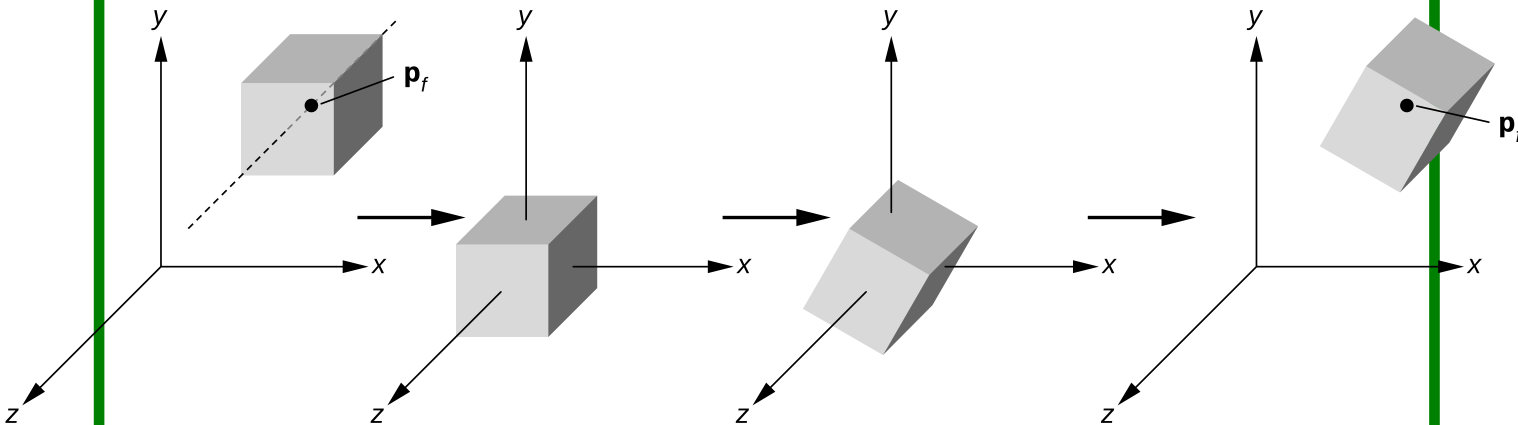
$$
\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{R_y}(\theta)} \cdot \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}
$$



Rotation wrt a Point



(a) (b)



$$R(\theta, p) = T(p) \cdot R_z(\theta) \cdot T(-p)$$

Transformations in OpenGL

☆ `glMatrixMode` (mode)

- Specifies the matrix that will be modified.
- GL_MODELVIEW , GL_PROJECTION, GL_TEXTURE.

☆ `glLoadIdentity` ()
Sets the current matrix to the 4×4 identity matrix.

☆ `glLoadMatrix`{fd} (type $*M$)
Sets the current matrix to M.

☆ `glMultMatrix`{fd} (type $*M$)
Multiplies the current matrix by M;
C: current matrix.

$$C = C \times M.$$

Transformations in OpenGL

- `glTranslate`{fd} $(x,\ y,\ z)$
  Translates an object by (x, y, z).

$$C = C \times T(x, y, z)$$

- `glScale`{fd} $(x,\ y,\ z)$
  Scales an object by (x, y, z).

$$C = C \times S(x, y, z)$$

- `glRotate`{fd} $(\alpha,\ x,\ y,\ z)$
  Rotates an object by angle α w.r.t. the ray emanating from the origin to the point (x, y, z).

$$C = C \times R(\alpha, x, y, z)$$

Order of multiplication

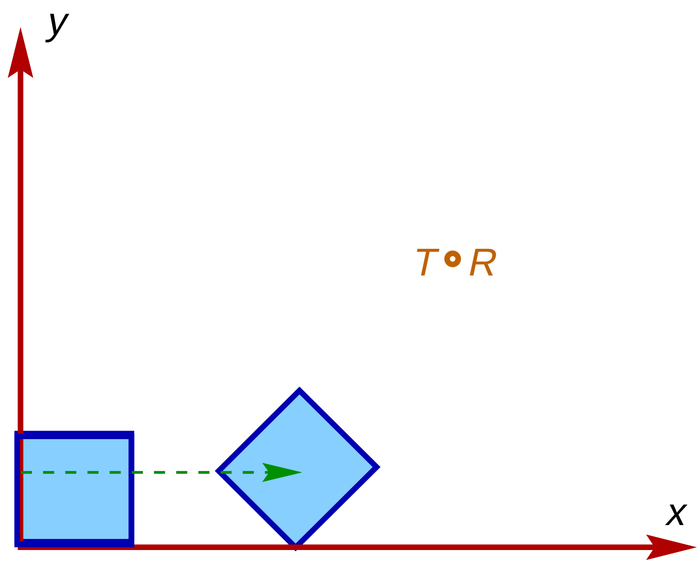
Last transformation is applied first!

`glLoadIdentity();`	$C = I$
`glMultMatrixf (L);`	$C = L$
`glMultMatrixf (M);`	$C = L \times M$
`glMultMatrixf (N);`	$C = L \times M \times N$

```
glLoadIdentity();
glMultMatrixf (L);
glMultMatrixf (M);
glMultMatrixf (N);
glBegin(GL_POINT)
    glVertex3f(v);
glEnd();
```

$$v' = C \times v = L \times M \times N \times v$$

Order of multiplication



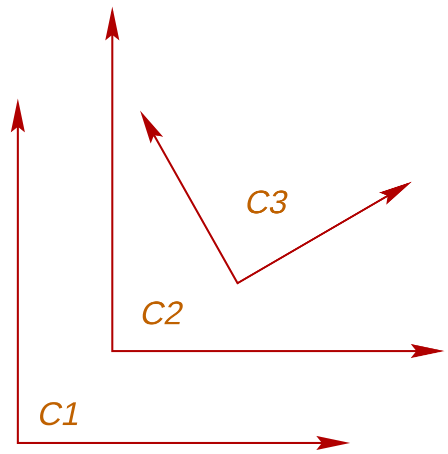
`glTranslatef` (T);
`glRotatef` (R);

Assume the object comes with its own coordinate system.

- ☆ Keep the object coordinates the same as world coordinates and transform the object.
- ☆ Transform the object coordinate system.

Multiple Coordinate systems

C_i: i-th coordinate system.
$P^{(i)}$: Coordinates of P in C_i
M_{ij}: Transformation $C_j \to C_i$.



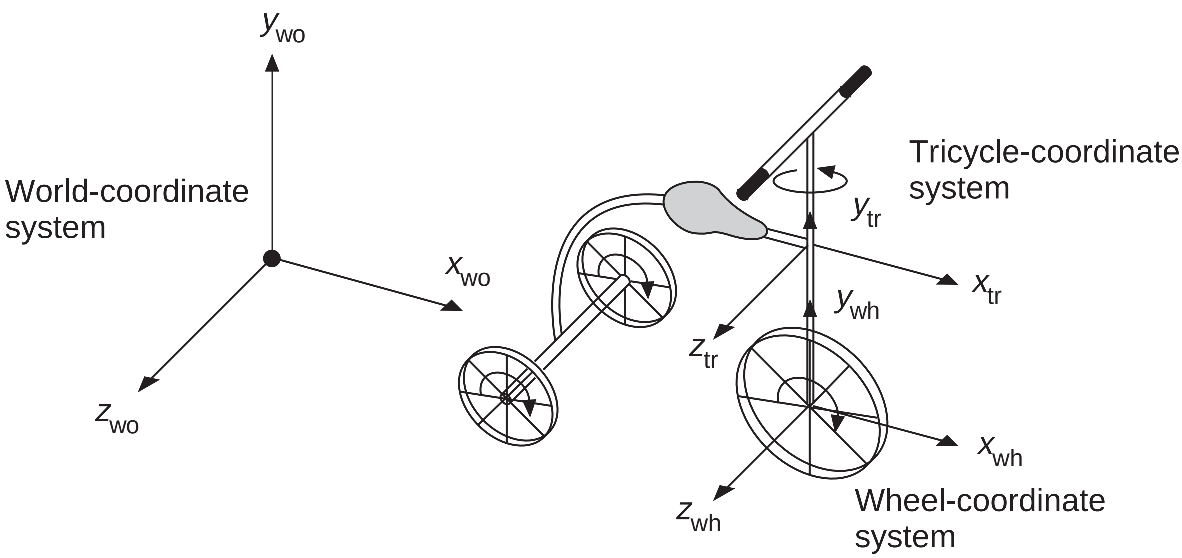
$$P^{(i)} = M_{ij} \cdot P^{(j)}.$$

$$\begin{aligned} M_{ik} &= M_{ij} \cdot M_{jk} \\ M_{ij} &= M_{ji}^{-1}. \end{aligned}$$

$Q^{(j)}$: A geometric transform in C_j.

$$Q^{(i)} = M_{ij} \cdot Q^{(j)} \cdot M_{ji} = M_{ij} \cdot Q^{(j)} \cdot M_{ij}^{-1}.$$

Multiple Coordinate Systems



Coordinate systems: World, tricycle, wheels

As the front wheel rotates around its z-axis:

- ☆ Both wheels rotate about the z-axes of wheels coordinate system.
- ☆ Tricycle moves as a whole.
- ☆ Both tricycle and wheels move w.r.t world coordinates.
- ☆ Wheel and tricycle coordinates are related by translation in x and z directions and rotation in y-direction.