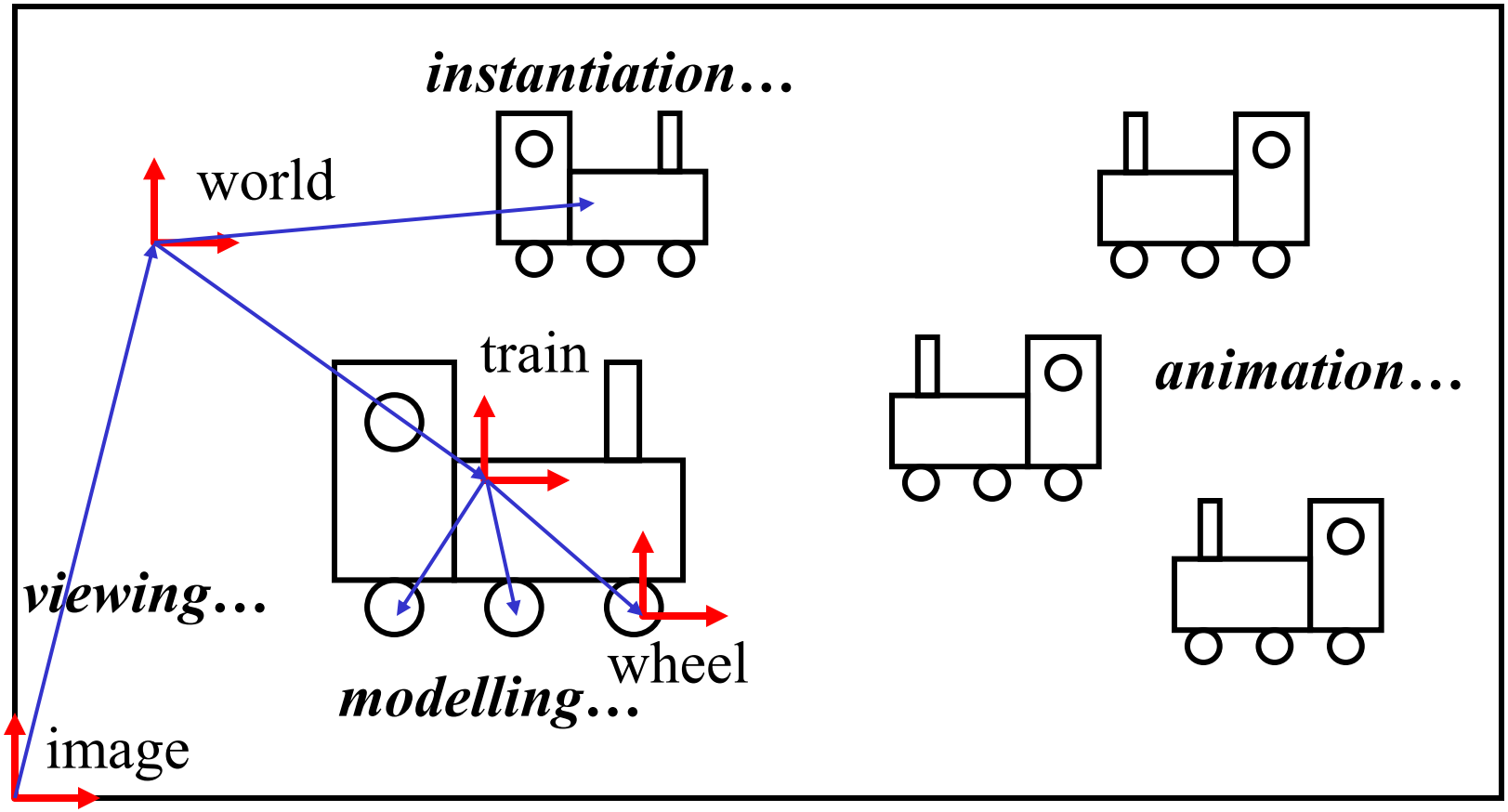
Transformations

Unit 2- Lecture 1

Overview

- Why transformations?
- Basic transformations:
  - translation, rotation, scaling
- Combining transformations
  - homogenous coordinates, transform. Matrices
- First 2D, next 3D

Transformations



Why transformation?

- Model of objects

  world coordinates: *km, mm, etc.*

  Hierarchical models::

  human = torso + arm + arm + head + leg + leg

  arm = upperarm + lowerarm + hand ...
- Viewing

  zoom in, move drawing, etc.
- Animation

Translation

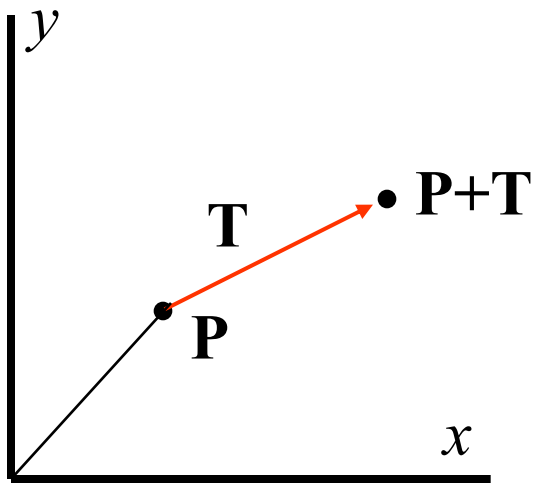
Translate over vector (t_x, t_y)

$$x' = x + t_x, \quad y' = y + t_y$$

or

$$\mathbf{P}' = \mathbf{P} + \mathbf{T}, \text{ with}$$

$$\mathbf{P}' = \begin{pmatrix} x' \\ y' \end{pmatrix}, \mathbf{P} = \begin{pmatrix} x \\ y \end{pmatrix} \text{ and } \mathbf{T} = \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$



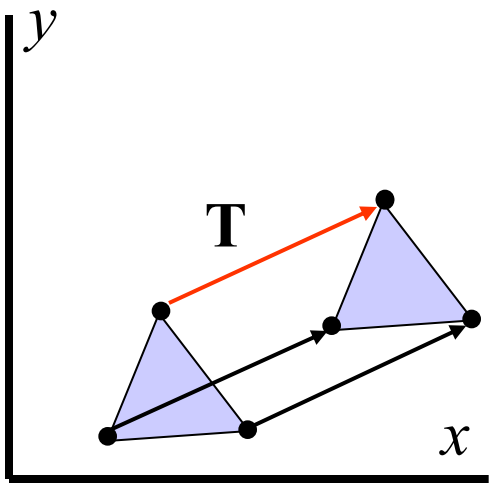
H&B 7-1:220-222

Translation polygon

Translate polygon:

Apply the same operation on all points.

Works always, for all transformations of objects defined as a set of points.



H&B 7-1:220-222

Rotation

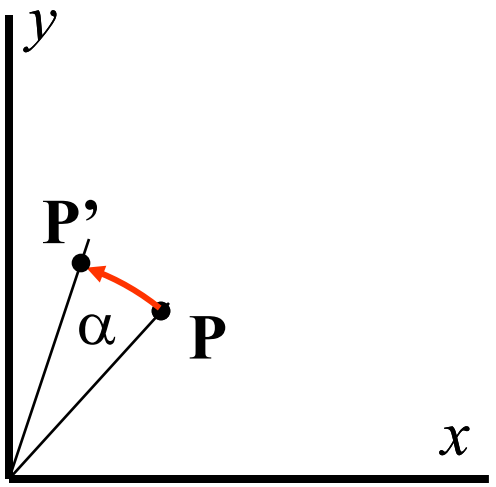
Rotate over an angle α:

$$x' = x\cos\alpha - y\sin\alpha$$

$$y' = x\sin\alpha + y\cos\alpha$$

Or

$$\mathbf{P}' = \mathbf{R}\mathbf{P}, \text{ with}$$



$$\mathbf{P}' = \begin{pmatrix} x' \\ y' \end{pmatrix}, \mathbf{R} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \text{ and } \mathbf{P} = \begin{pmatrix} x \\ y \end{pmatrix}$$

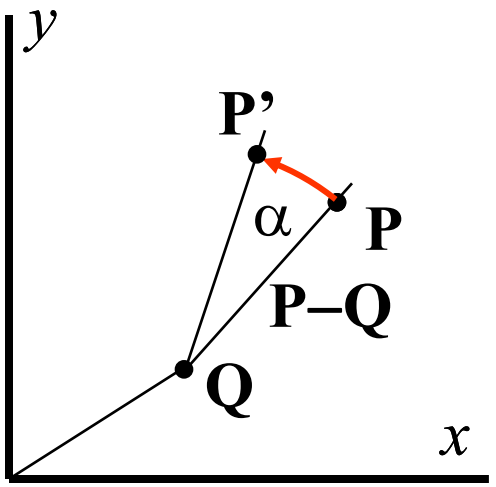
H&B 7-1:222-223

Rotation around a point **Q**

Rotate around origin:

$$P_x' = P_x \cos\alpha - P_y \sin\alpha$$

$$P_y' = P_x \sin\alpha + P_y \cos\alpha$$



Rotate around **Q** over an angle α:

$$P_x' = Q_x + (P_x - Q_x)\cos\alpha - (P_y - Q_y)\sin\alpha$$

$$P_y' = Q_y + (P_x - Q_x)\sin\alpha + (P_y - Q_y)\cos\alpha$$

H&B 7-1:222-223

Scaling

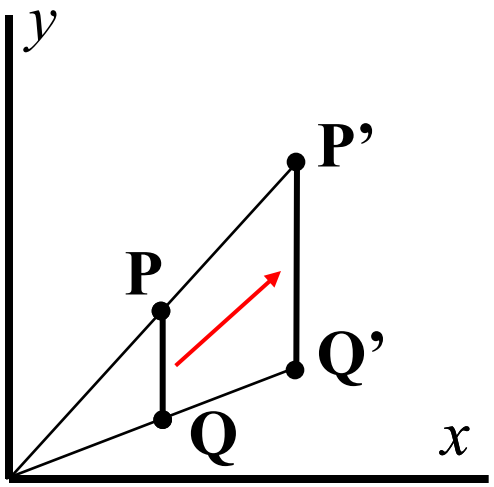
Schale with factor s_x and s_y:

$$x' = s_x\, x,\quad y' = s_y\, y$$

or

$$\mathbf{P}' = \mathbf{S}\mathbf{P},\ \text{with}$$

$$\mathbf{P}' = \begin{pmatrix} x' \\ y' \end{pmatrix}, \mathbf{S} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \text{and}\ \mathbf{P} = \begin{pmatrix} x \\ y \end{pmatrix}$$



H&B 7-1:224-225

Scaling with respect to a point **F**

Scale with factors s_x and s_y:

$$P_x' = s_x P_x, \quad P_y' = s_y P_y$$

With respect to **F**:

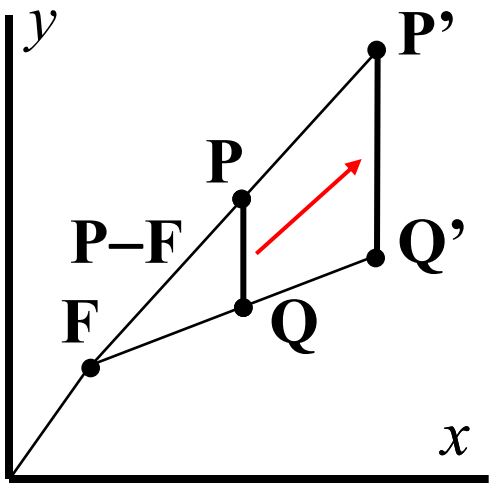
$$P_x' - F_x = s_x (P_x - F_x),$$

$$P_y' - F_y = s_y (P_y - F_y)$$

or

$$P_x' = F_x + s_x (P_x - F_x),$$

$$P_y' = F_y + s_y (P_y - F_y)$$



H&B 7-1:224-225

Transformations

- Translate with **V**:

  $\mathbf{T} = \mathbf{P} + \mathbf{V}$

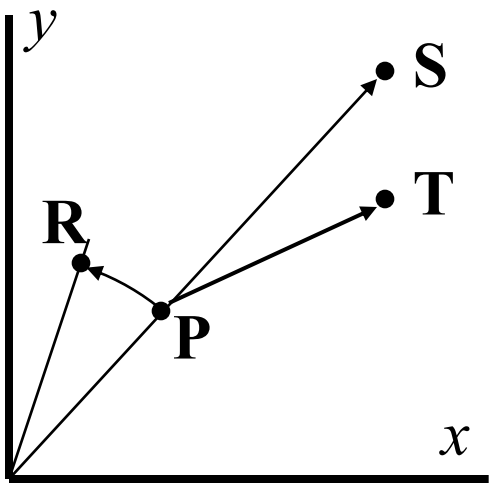
- Schale with factor $s_x = s_y = s$:

  $\mathbf{S} = s\mathbf{P}$

- Rotate over angle α:

  $R'_x = \cos\alpha\, P_x - \sin\alpha\, P_y$

  $R'_y = \sin\alpha\, P_x + \cos\alpha\, P_y$



H&B 7-2:225-228

Transformations…

- Messy!
- Transformations with respect to points: even more messy!
- How to combine transformations?

H&B 7-2:225-228

Homogeneous coordinates 1

- Uniform representation of translation, rotation, scaling
- Uniforme representation of points and vectors
- Compact representation of sequence of transformations

H&B 7-2:225-228

Homogeneous coordinaten 2

- Add extra coordinate:

  $\mathbf{P} = (p_x, p_y, p_h)$ or

  $\mathbf{x} = (x, y, h)$
- Cartesian coordinates: divide by h

  $\mathbf{x} = (x/h, y/h)$
- Points: $h = 1$ (for the time being…),

  vectors: $h = 0$

H&B 7-2:225-228

Translation matrix

Translation:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

or

$$\mathbf{P}' = \mathbf{T}(t_x, t_y)\mathbf{P}$$

H&B 7-2:225-228

Rotation matrix

Rotation:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

or

$$\mathbf{P}' = \mathbf{R}(\theta)\mathbf{P}$$

H&B 7-2:225-228

Scaling matrix

Scaling :

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

or

$$\mathbf{P}' = \mathbf{S}(s_x, s_y)\mathbf{P}$$

H&B 7-2:225-228

Inverse transformations

Translation:

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

Rotation:

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

Scaling:

$$\mathbf{S}^{-1}(s_x, s_y) = \mathbf{S}\left(\frac{1}{s_x}, \frac{1}{s_y}\right)$$

H&B 7-3:228

Combining transformations 1

$\mathbf{P}' = \mathbf{M}_1\mathbf{P}$ first transformation...

$\mathbf{P}'' = \mathbf{M}_2\mathbf{P}'$ second transformation...

Combined:

$$\mathbf{P}'' = \mathbf{M}_2(\mathbf{M}_1\mathbf{P})$$
$$= \mathbf{M}_2\mathbf{M}_1\mathbf{P}$$
$$= \mathbf{M}\mathbf{P} \quad \text{with } \mathbf{M} = \mathbf{M}_2\mathbf{M}_1$$

H&B 7-4:228-229

Combining transformations 2

$\mathbf{P}' = \mathbf{T}(t_{1x}, t_{1y})\mathbf{P}$ first translation

$\mathbf{P}'' = \mathbf{T}(t_{2x}, t_{2y})\mathbf{P}'$ second translation

Combined:

$$\mathbf{P}'' = \mathbf{T}(t_{2x}, t_{2y})\mathbf{T}(t_{1x}, t_{1y})\mathbf{P}$$

$$= \begin{pmatrix} 1 & 0 & t_{2x} \\ 0 & 1 & t_{2y} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & t_{1x} \\ 0 & 1 & t_{1y} \\ 0 & 0 & 1 \end{pmatrix} \mathbf{P} = \begin{pmatrix} 1 & 0 & t_{1x} + t_{2x} \\ 0 & 1 & t_{1y} + t_{2y} \\ 0 & 0 & 1 \end{pmatrix} \mathbf{P}$$

$$= \mathbf{T}(t_{1x} + t_{2x}, t_{1x} + t_{2y})\mathbf{P}$$

H&B 7-4:229

Combining transformations 3

Composite translations:

$$\mathbf{T}(t_{2x}, t_{2y})\mathbf{T}(t_{1x}, t_{1y}) = \mathbf{T}(t_{1x} + t_{2x}, t_{1x} + t_{2y})$$

Composite rotations:

$$\mathbf{R}(\theta_2)R(\theta_1) = \mathbf{R}(\theta_1 + \theta_2)$$

Composite scaling:

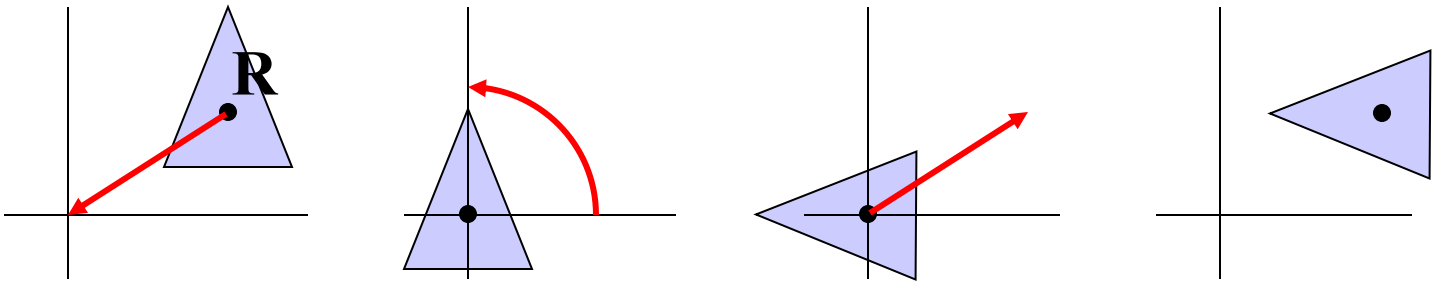
$$\mathbf{S}(s_{2x}, s_{2y})\mathbf{S}(s_{1x}, s_{1y}) = \mathbf{S}(s_{1x}s_{2x}, s_{1y}s_{2y})$$

H&B 7-4:229

Rotation around a point 1

Rotate over angle θ around point **R**:

1) Translate such that **R** coincides with origin;
2) Rotate over angle θ around origin;
3) Translate back.



1) 2) 3)

H&B 7-4:229-230

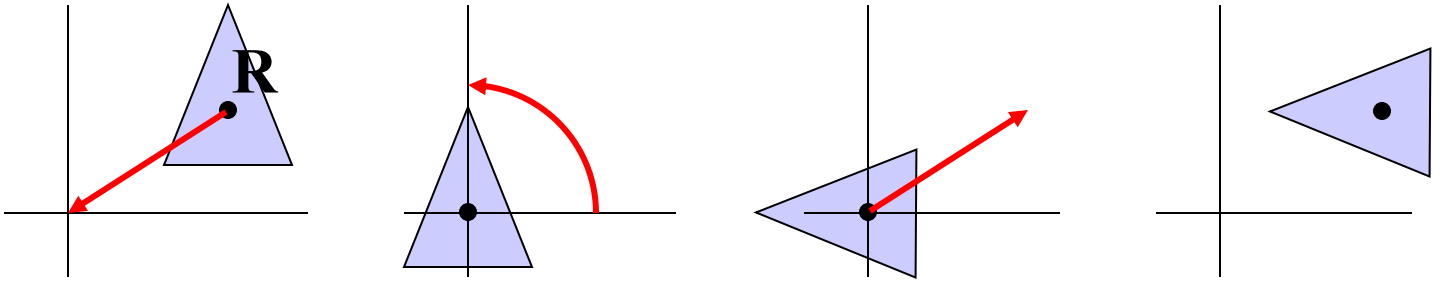
Rotation around a point 2

Rotate over angle θ around point $\mathbf{R}$:

1) $\mathbf{P}' = \mathbf{T}(-R_x, -R_y)\mathbf{P}$

2) $\mathbf{P}'' = \mathbf{R}(\theta)\mathbf{P}'$

3) $\mathbf{P}''' = \mathbf{T}(R_x, R_y)\mathbf{P}''$



1) 2) 3)

H&B 7-4:229-230

Rotation around point 3

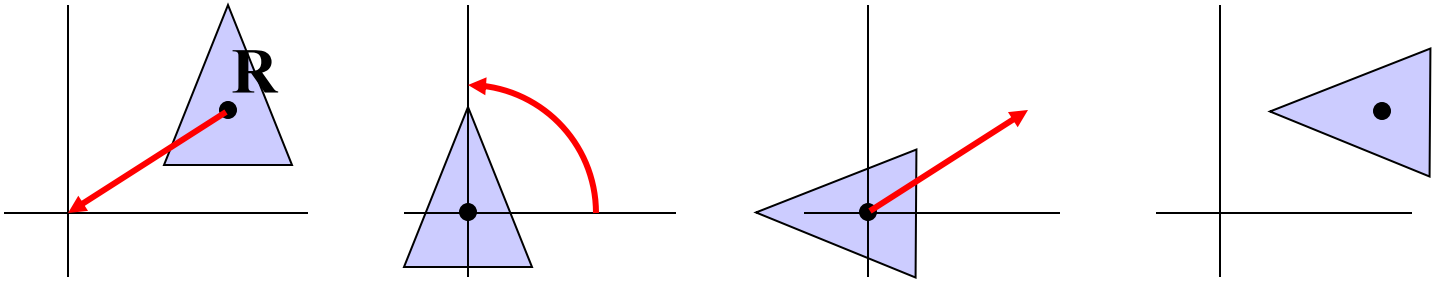
1) $\mathbf{P}' = \mathbf{T}(-R_x, -R_y)\mathbf{P}$

2) $\mathbf{P}'' = \mathbf{R}(\theta)\mathbf{P}' = \mathbf{R}(\theta)\mathbf{T}(-R_x, -R_y)\mathbf{P}$

3) $\mathbf{P}''' = \mathbf{T}(R_x, R_y)\mathbf{P}''$

$= \mathbf{T}(R_x, R_y)\mathbf{R}(\theta)\mathbf{P}'$

$= \mathbf{T}(R_x, R_y)\mathbf{R}(\theta)\mathbf{T}(-R_x, -R_y)\mathbf{P}$



1) 2) 3)

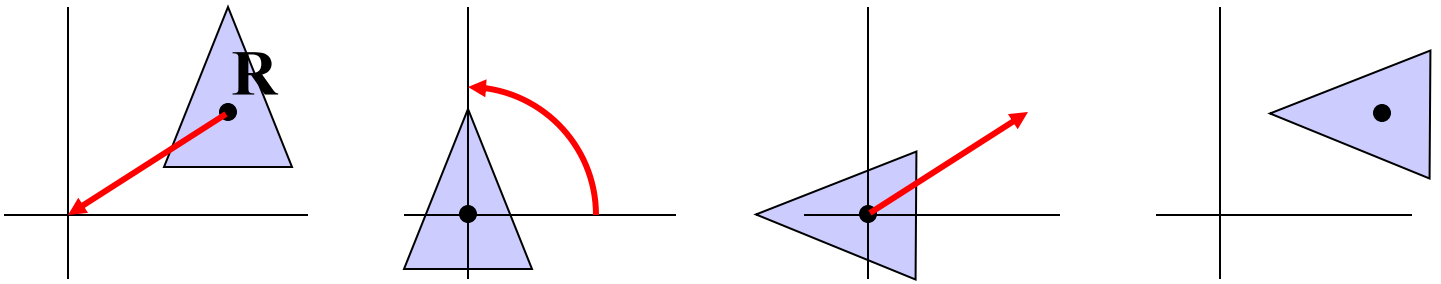
H&B 7-4:229-230

Rotation around point 4

$$1\text{-}3)\ \mathbf{P}''' = \mathbf{T}(R_x, R_y)\mathbf{R}(\theta)\mathbf{T}(-R_x, -R_y)\mathbf{P}$$

or

$$\mathbf{P}''' = \begin{pmatrix} \cos\theta & -\sin\theta & R_x(1-\cos\theta) + R_y \sin\theta \\ \sin\theta & \cos\theta & R_y(1-\cos\theta) - R_x \sin\theta \\ 0 & 0 & 1 \end{pmatrix} \mathbf{P}$$



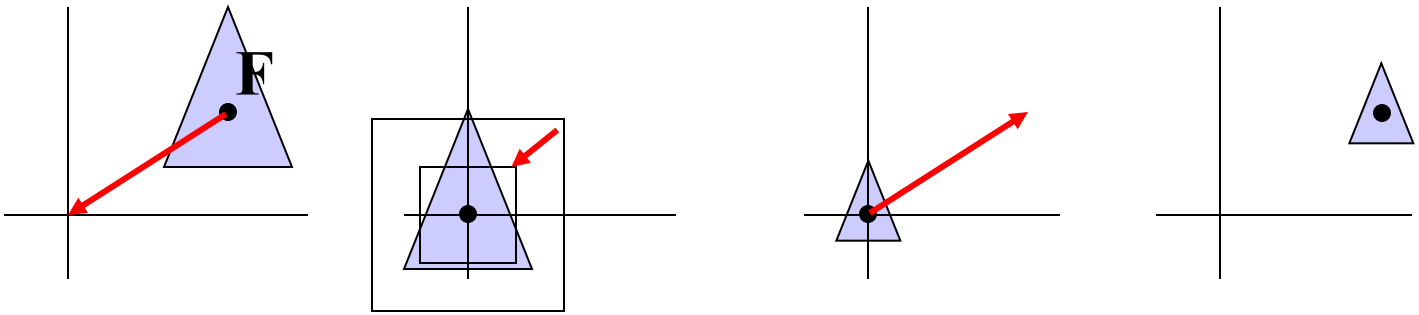
1) 2) 3)

H&B 7-4:229-230

Scaling w.r.t. point 1

Scale with factors s_x and s_x w.r.t. point **F**:

1) Translate such that **F** coincides with origin;
2) Schale w.r.t. origin;
3) Translate back again.



1) 2) 3)

H&B 7-4:230-231

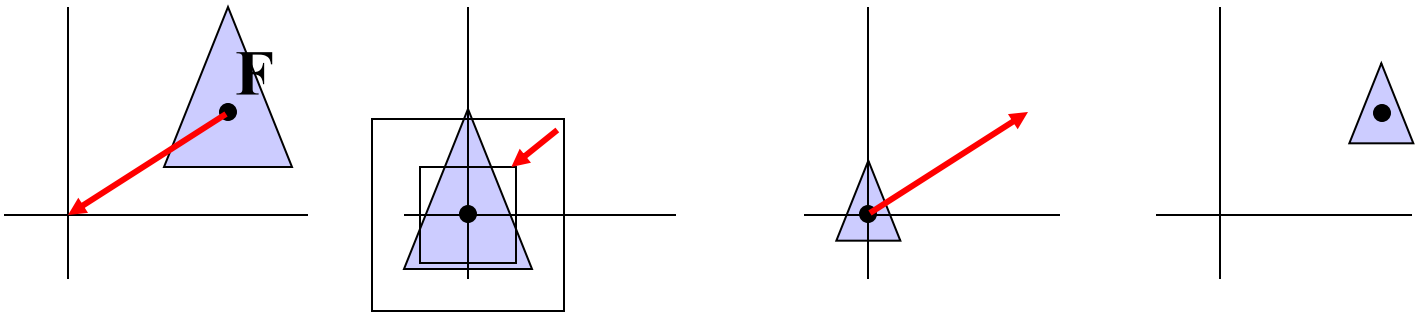
Scaling w.r.t.point 2

Schale w.r.t.point **F** :

1) $\mathbf{P}' = \mathbf{T}(-F_x, -F_y)\mathbf{P}$

2) $\mathbf{P}'' = \mathbf{S}(s_x, s_y)\mathbf{P}'$

3) $\mathbf{P}''' = \mathbf{T}(F_x, F_y)\mathbf{P}''$



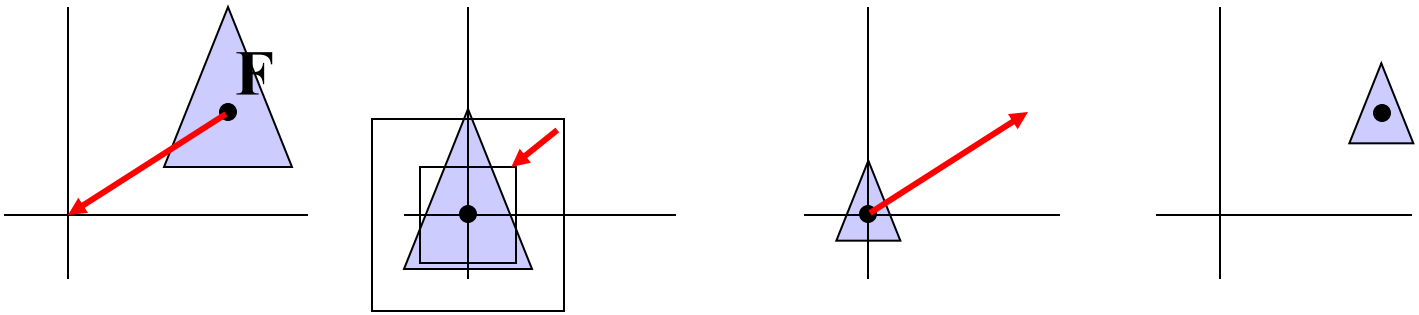
H&B 7-4:230-231

Scaling w.r.t.point 3

$$1\text{-}3)\ \mathbf{P}''' = \mathbf{T}(F_x, F_y)\mathbf{S}(s_x, s_y)\mathbf{T}(-F_x, -F_y)\mathbf{P}$$

or

$$\mathbf{P}''' = \begin{pmatrix} s_x & 0 & F_x(1-s_x) \\ 0 & s_y & F_y(1-s_y) \\ 0 & 0 & 1 \end{pmatrix} \mathbf{P}$$

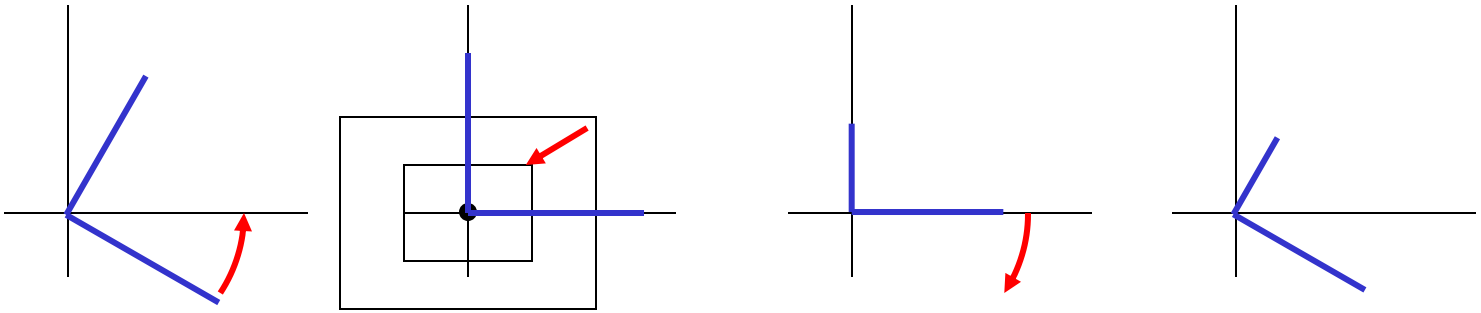


H&B 7-4:230-231

Scale in other directions 1

Scale with factors s_1 and s_2 w.r.t. rotated frame:

1) Rotate such that frame coincides with standard xy - frame;
2) Scale w.r.t. origin;
3) Rotate back again.



H&B 7-4:230-231

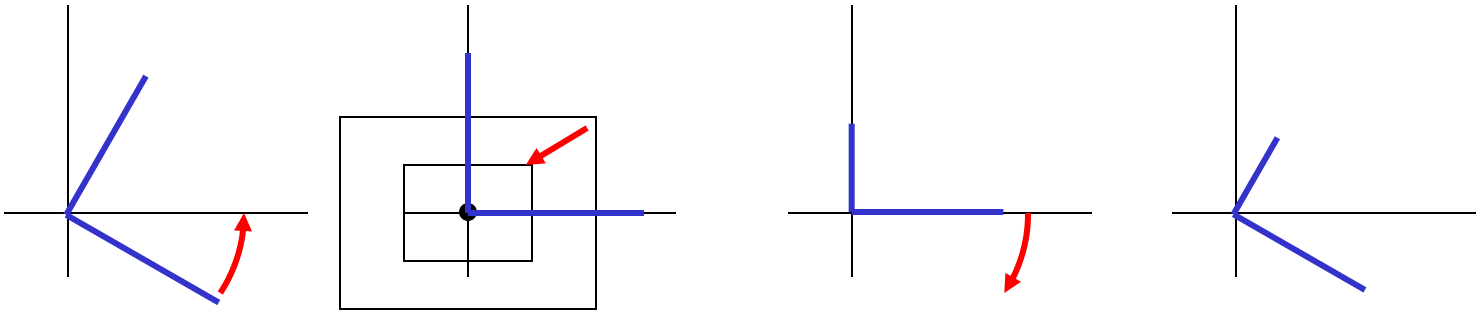
Scale in other directions 2

Scale in otherdirection:

1) $\mathbf{P}' = \mathbf{R}(\theta)\mathbf{P}$

2) $\mathbf{P}'' = \mathbf{S}(s_1, s_2)\mathbf{P}'$

3) $\mathbf{P}''' = \mathbf{R}(-\theta)\mathbf{P}''$



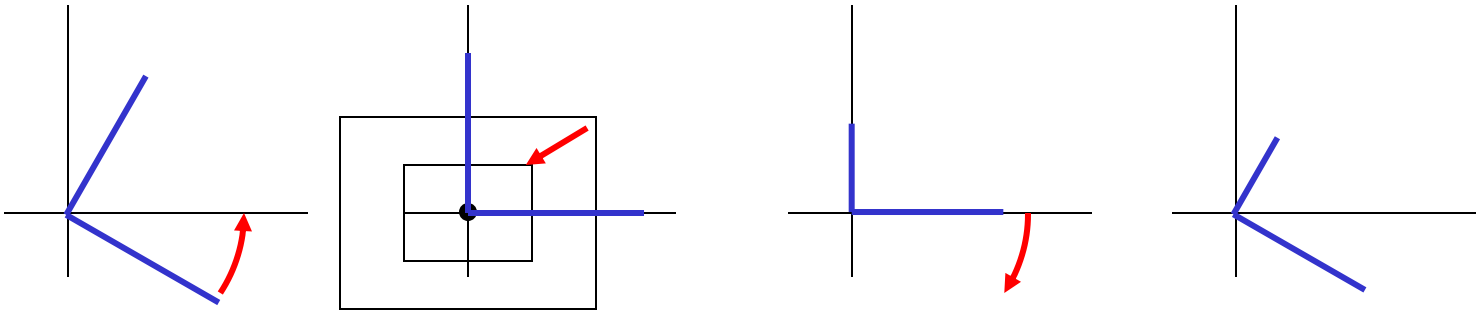
H&B 7-4:230-231

Scale in other directions 3

1-3) $\mathbf{P}''' = \mathbf{R}(-\theta)\mathbf{S}(s_1, s_2)\mathbf{R}(\theta)\mathbf{P}$

or

$$\mathbf{P}''' = \begin{pmatrix} s_1\cos^2\theta + s_2\sin^2\theta & (s_2 - s_1)\cos\theta\sin\theta & 0 \\ (s_2 - s_1)\cos\theta\sin\theta & s_1\sin^2\theta + s_2\cos^2\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}\mathbf{P}$$

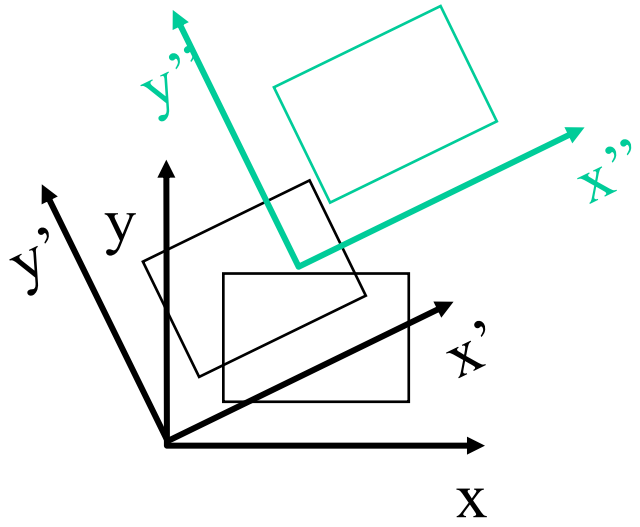


1) 2) 3)

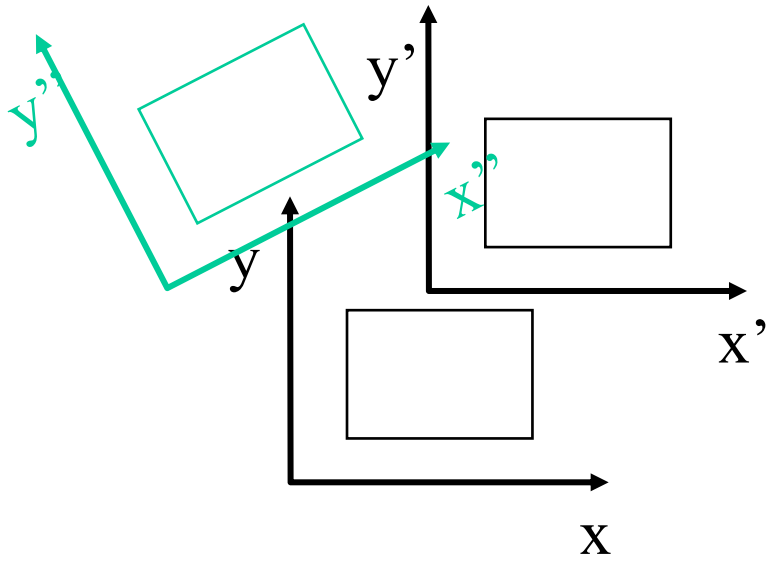
H&B 7-4:230-231

Order of transformations 1

Rotation, translation…



Translation, rotation…



Matrix multiplication does **not** commute.

The order of transformations makes a difference!

H&B 7-4:232

Order of transformations 2

- Pre-multiplication:

  $\mathbf{P'} = \mathbf{M}_n \mathbf{M}_{n-1} \ldots \mathbf{M}_2 \mathbf{M}_1 \mathbf{P}$

  Transformation $\mathbf{M}_n$ in global coordinates
- Post-multiplication:

  $\mathbf{P'} = \mathbf{M}_1 \mathbf{M}_2 \ldots \mathbf{M}_{n-1} \mathbf{M}_n \mathbf{P}$

  Transformation $\mathbf{M}_n$ in local coordinates: the coordinate system after application of $\mathbf{M}_1 \mathbf{M}_2 \ldots \mathbf{M}_{n-1}$

H&B 7-4:232

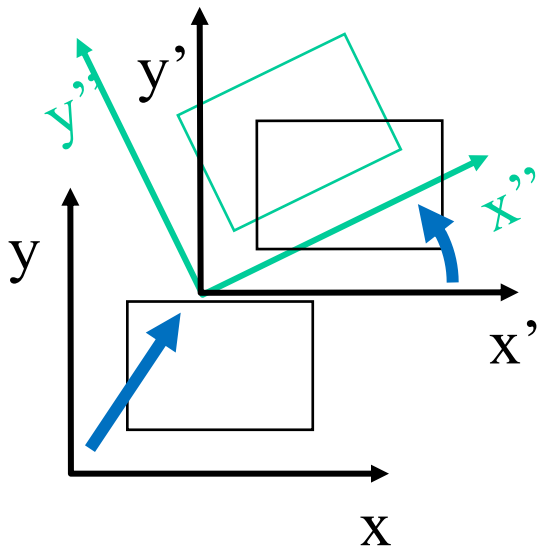
Order of transformations 3

OpenGL: **`glRotate, glScale`**, etc.:

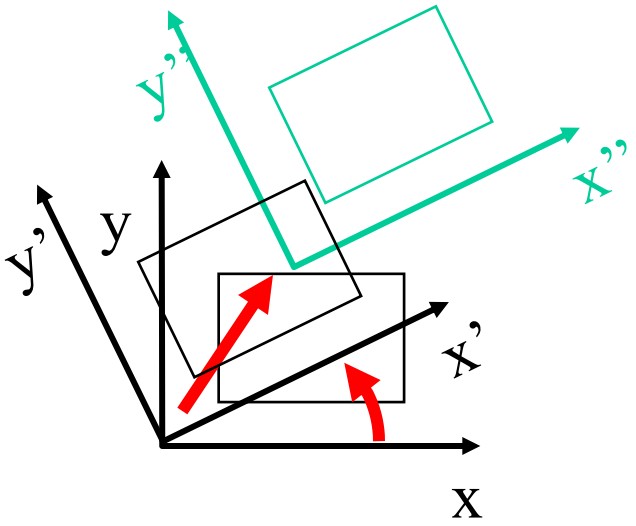
- Post-multiplication of current transformation matrix
- Always transformation in local coordinates
- Global coordinate version: read in reverse order

Order of transformations 4

***Local** transformations*:



***Global** transformations:*



***Local** trafo*

interpretation

```
glTranslate(…);
glRotate(…);
```

***Global** trafo*

interpretation

H&B 7-4:232

Matrices in general

$$\mathbf{P}' = \mathbf{M}\mathbf{P}, \text{ or}$$

rotation and scaling

translation

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} rs_{xx} & rs_{xy} & trs_x \\ rs_{yx} & rs_{yy} & trs_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \text{ or}$$

$$x' = rs_{xx}x + rs_{xy}y + trs_x$$

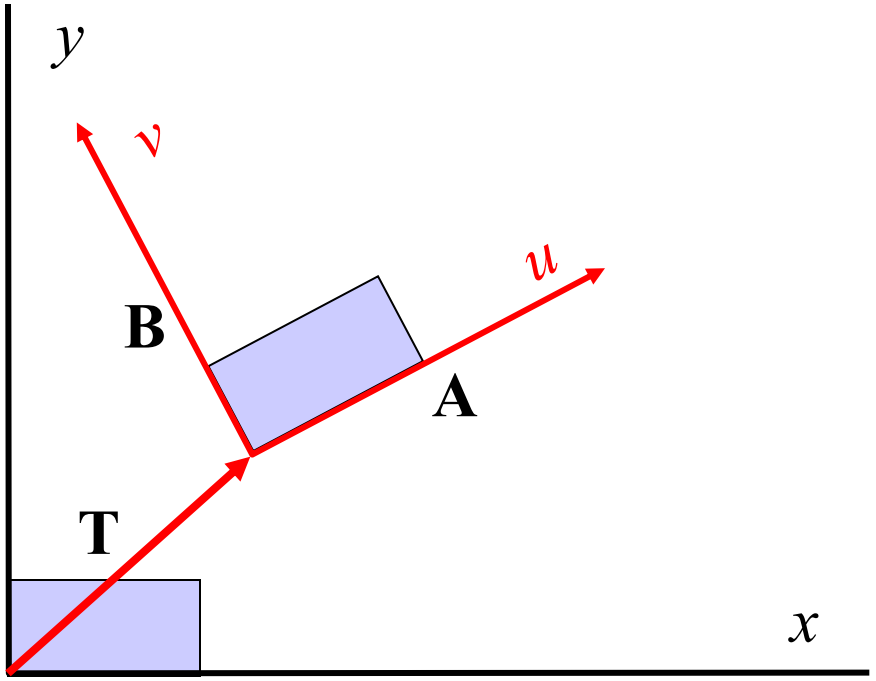
$$y' = rs_{yx}x + rs_{yy}y + trs_y$$

H&B 7-4:233-235

Direct construction of matrix

If you know the target frame:
Construct matrix directly.

Define shape in nice local *u,v* coordinates, use matrix transformation to put it in *x,y* space.



H&B 7-4:233-235

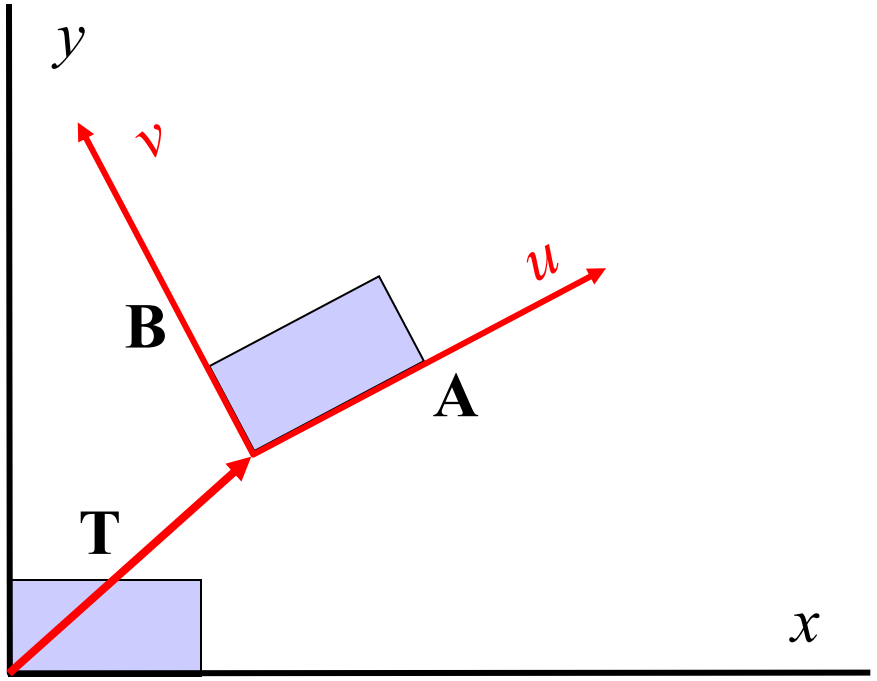
Direct construction of matrix

If you know the target frame:
Construct matrix directly.

$$\mathbf{P'} = \mathbf{A}u + \mathbf{B}v + \mathbf{T}\text{, or}$$

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{B} & \mathbf{T} \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}\text{, or}$$

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} A_x & B_x & T_x \\ A_y & B_y & T_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$



H&B 7-4:233-235

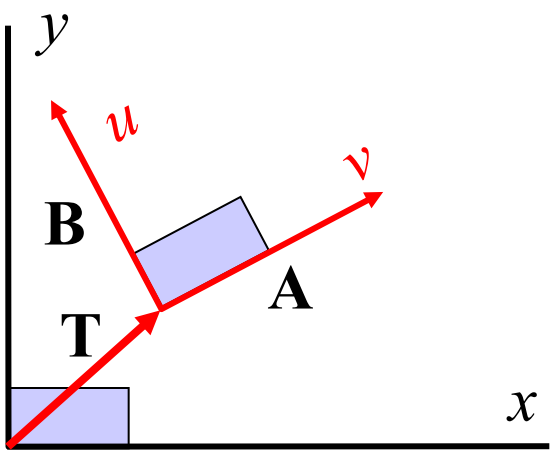
Rigid body transformation

$\mathbf{P}' = \mathbf{M}\mathbf{P}$, of

only rotation

translation

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} r_{xx} & r_{xy} & tr_x \\ r_{yx} & r_{yy} & tr_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$



$\begin{pmatrix} r_{xx} & r_{xy} \\ r_{yx} & r_{yy} \end{pmatrix}$: orthonormal submatrix

$$\mathbf{A} = \begin{pmatrix} r_{xx} \\ r_{yx} \end{pmatrix} \text{ and } \mathbf{B} = \begin{pmatrix} r_{xy} \\ r_{yy} \end{pmatrix}, \ |\mathbf{A}| = 1, |\mathbf{B}| = 1, \mathbf{A} \cdot \mathbf{B} = 0$$

H&B 7-4:233-235

Other 2D transformations

- Reflection
- Shear

Can also be combined

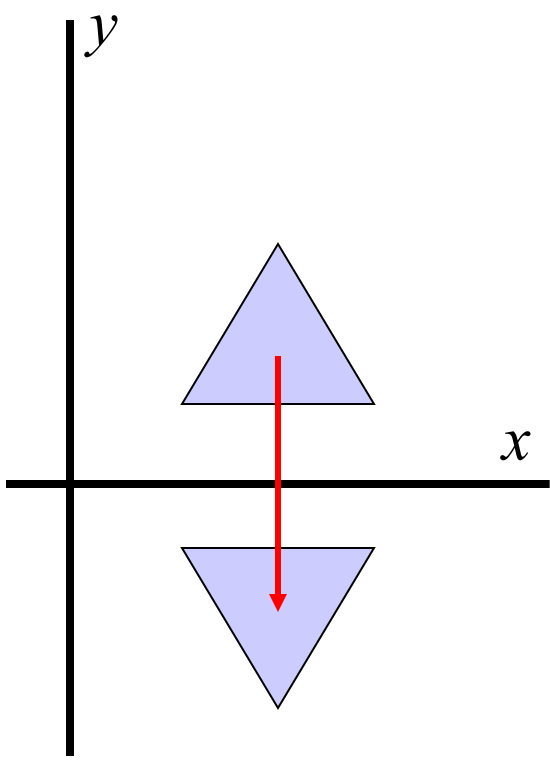
Reflection over axis

Reflext over x-axis:

$$x' = x, \quad y' = -y$$

or

$$\mathbf{P'} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{P}$$



H&B 7-4:240-242

Reflect over origin

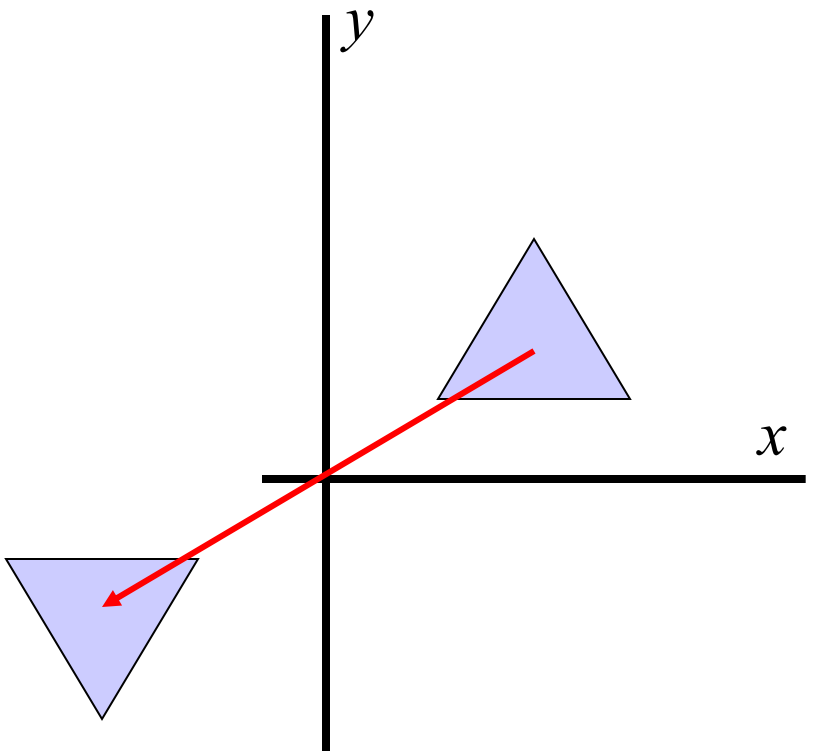
Reflect over origin:

$$x' = -x, \quad y' = -y$$

or

$$\mathbf{P}' = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{P}$$

Same as $\mathbf{P}' = \mathbf{R}(180)\mathbf{P}$



H&B 7-4:240-242

Shear

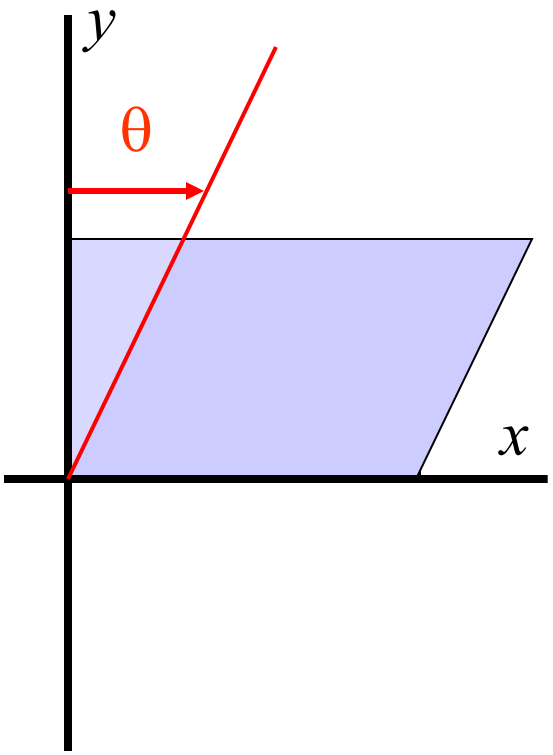
Shear the y-as:

$$x' = x + fy, \quad y' = y$$

or

$$\mathbf{P'} = \begin{pmatrix} 1 & f & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{P}$$

with $f = \tan\theta$

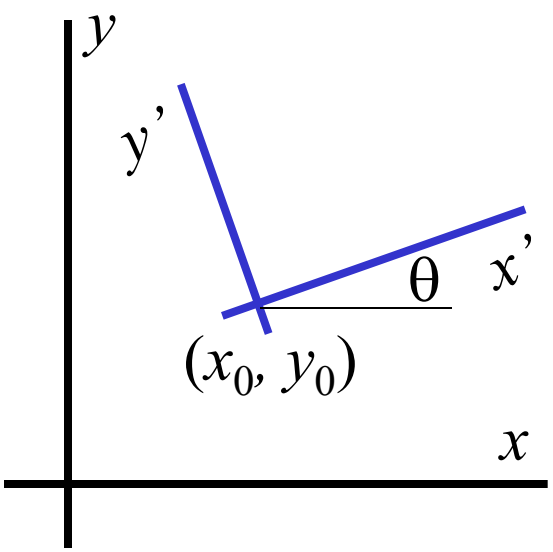


H&B 7-4:242-243

Transformations coordinates

Given (x,y)-coordinates,
Find (x',y')-coordinates.

Reverse route as
object transformaties.

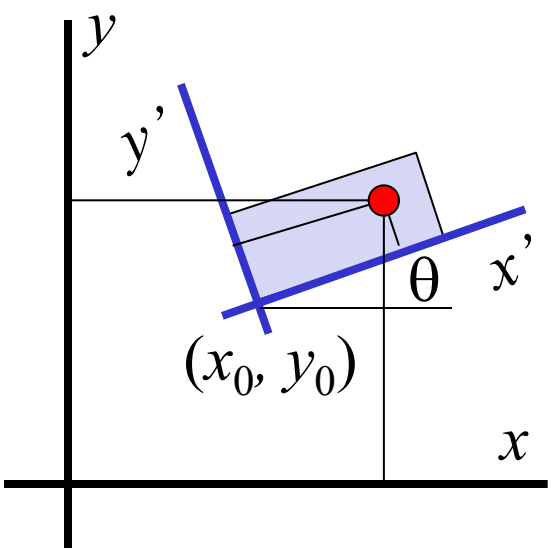


H&B 7-8:246-248

Transformations coordinates

Given (x,y)-coordinates,
Find (x',y')-coordinates.

Example: user points at (x,y), what's the position in local coordinates?



H&B 7-8:246-248

Transformations coordinates

Given **X**: (x,y)-coordinates,

Find **X'**: (x',y')-coordinates.

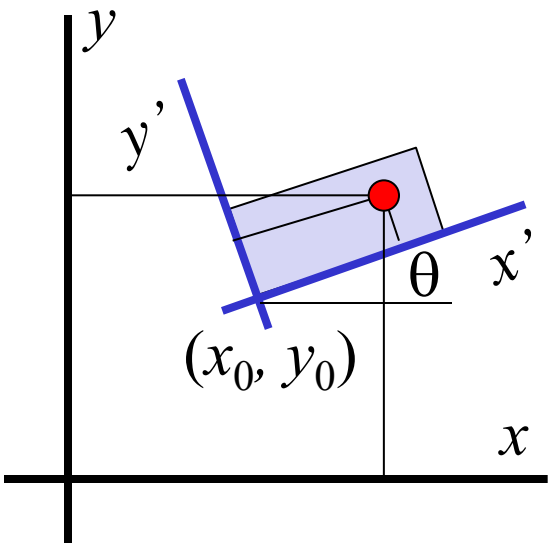
Standard:

X=MX' (object trafo:

from local to global)

Here:

X'=M^{-1}X (from global to local)



H&B 7-8:246-248

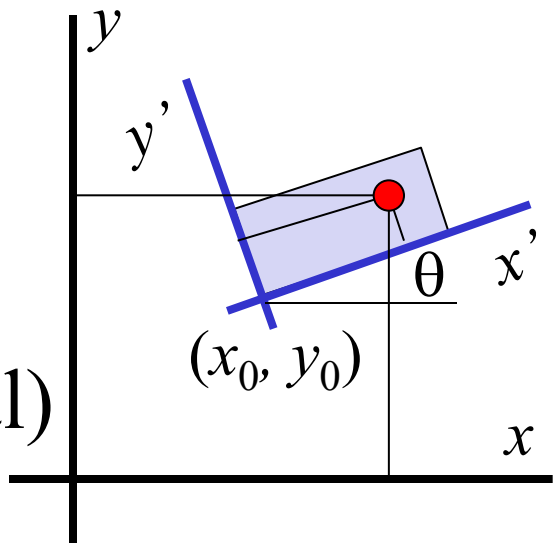
Transformations coordinates

Given **X**: (x,y)-coordinates,

Find **X'**: (x',y')-coordinates.

Here:

$\mathbf{X'=M^{-1}X}$ (from global to local)



Approach 1:

- Determine "standard matrix" **M** (from local to global coordinates) and invert

H&B 7-8:246-248

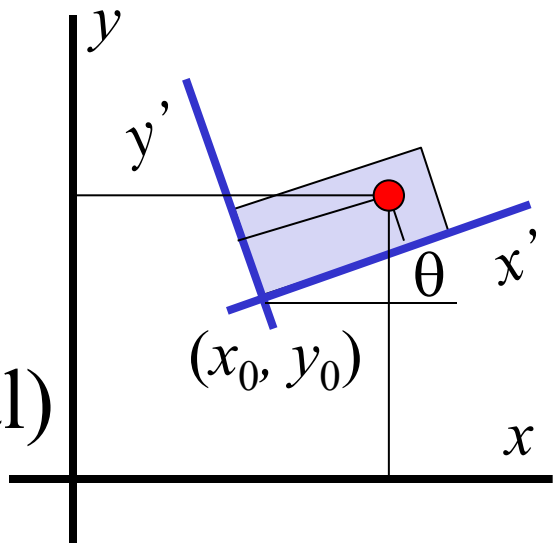
Transformations coordinates

Given **X**: (x,y)-coordinates,

Find **X'**: (x',y')-coordinates.

Here:

$\mathbf{X'=M^{-1}X}$ (from global to local)



Approach 2:

- construct transformation that maps local frame to global (*reverse of usual*).

H&B 7-8:246-248

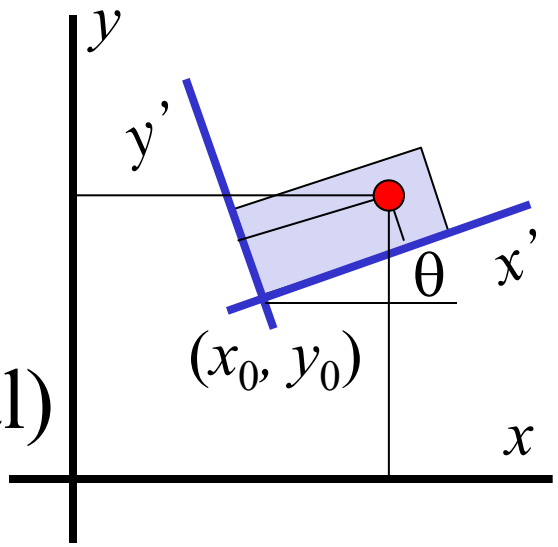
Transformations coordinates

Given **X**: (x,y)-coordinates,

Find **X'**: (x',y')-coordinates.

Here:

$\mathbf{X'=M^{-1}X}$ (from global to local)



Approach 2:

1. Translate (x_0, y_0) to origin;
2. Rotate x'-axis to x-axis.

H&B 7-8:246-248

Transformations coordinates

Given **X**: (x,y)-coordinates,

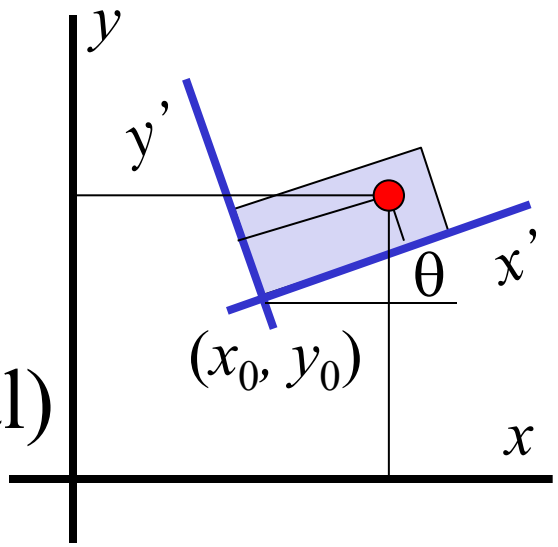
Find **X'**: (x',y')-coordinates.

Here:

$\mathbf{X'=M^{-1}X}$ (from global to local)

Approach 2:

$\mathbf{M^{-1} = R}(-\theta)\mathbf{T}(-x_0, -y_0)$



H&B 7-8:246-248

OpenGL 2D transformations 1

Internally:

- Coordinates are four-element row vectors
- Transformations are 4×4 matrices

2D trafo's: Ignore z-coordinates, set $z = 0$.

H&B 7-9:248-253

OpenGL 2D transformations 2

OpenGL maintains two matrices:

- **`GL_PROJECTION`**
- **`GL_MODELVIEW`**

Transformations are applied to the current matrix, to be selected with:

- **`glMatrixMode(GL_PROJECTION)`** or
- **`glMatrixMode(GL_MODELVIEW)`**

H&B 7-9:248-253

OpenGL 2D transformations 3

Initializing the matrix to **I**:

- `glLoadIdentity();`

Replace the matrix with **M**:

- `GLfloat M[16]; fill(M);`
- `glLoadMatrix*(M);`

Matrices are specified in *column-major* order:

$$\mathbf{M} = \begin{bmatrix} m[0] & m[4] & m[8] & m[12] \\ m[1] & m[5] & m[9] & m[13] \\ m[2] & m[6] & m[10] & m[14] \\ m[3] & m[7] & m[11] & m[15] \end{bmatrix}$$

Multiply current matrix with **M**:

- `glMultMatrix*(M);`

H&B 7-9:248-253

OpenGL 2D transformations 4

Basic transformation functions: generate matrix and post-multiply this with current matrix.

Translate over [tx, ty, tz]:

```
glTranslate*(tx, ty, tz);
```

Rotate over theta degrees (!) around axis [vx, vy, vz]:

```
glRotate*(theta, vx, vy, vz);
```

Scale axes with factors sx, sy, sz:

```
glScale*(sx, sy, sz);
```

H&B 7-9:248-253

OpenGL 2D Transformations 5

OpenGL maintains *stacks* of transformation matrices.

Two operations:

- **`glPushMatrix()`:**

  Make copy of current matrix and put that on top of the stack;
- **`glPopMatrix()`:**

  Remove top element of the stack.

Handy for dealing with hierarchical models

Handy for “undoing” transformations

H&B 9-8:324-327

OpenGL 2D Transformations 6

Standard:

```
glRotate(10, 1, 2, 0);
glScale(2, 1, 0.5);
glTranslate(1, 2, 3);

glutWireCube(1);

glTranslate(-1, -2, -3);
glScale(0.5, 1, 0.5);
glRotate(-10, 1, 2, 0);
```

Undo transformation

Using the stack:

```
glPushMatrix();
glRotate(10, 1, 2, 0);
glScale(2, 1, 0.5);
glTranslate(1, 2, 3);

glutWireCube(1);

glPopMatrix();
```

Shorter, more robust

H&B 9-8:324-327

2D transformations summarized

- Transformations: modeling, viewing, animation;
- Several kinds of transformations;
- Homogeneous coordinates;
- Combine transformations using matrix multiplication.

Up to 3D!