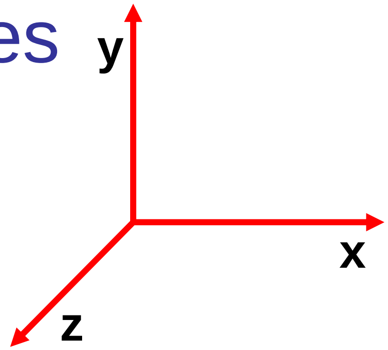
Computer Graphics

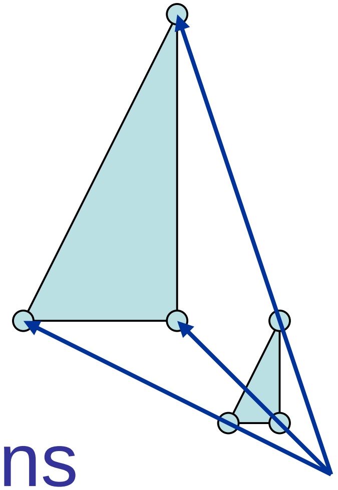
Geometric Transformations

Overview

- coordinate systems
  - scalar values, points, vectors, matrices
  - right-handed and left-handed coordinate systems
  - mathematical foundations



- transformations
  - mathematical descriptions of geometric changes, 2D and 3D
  - matrices: efficient representation
  - matrices for common transformations



The Basics

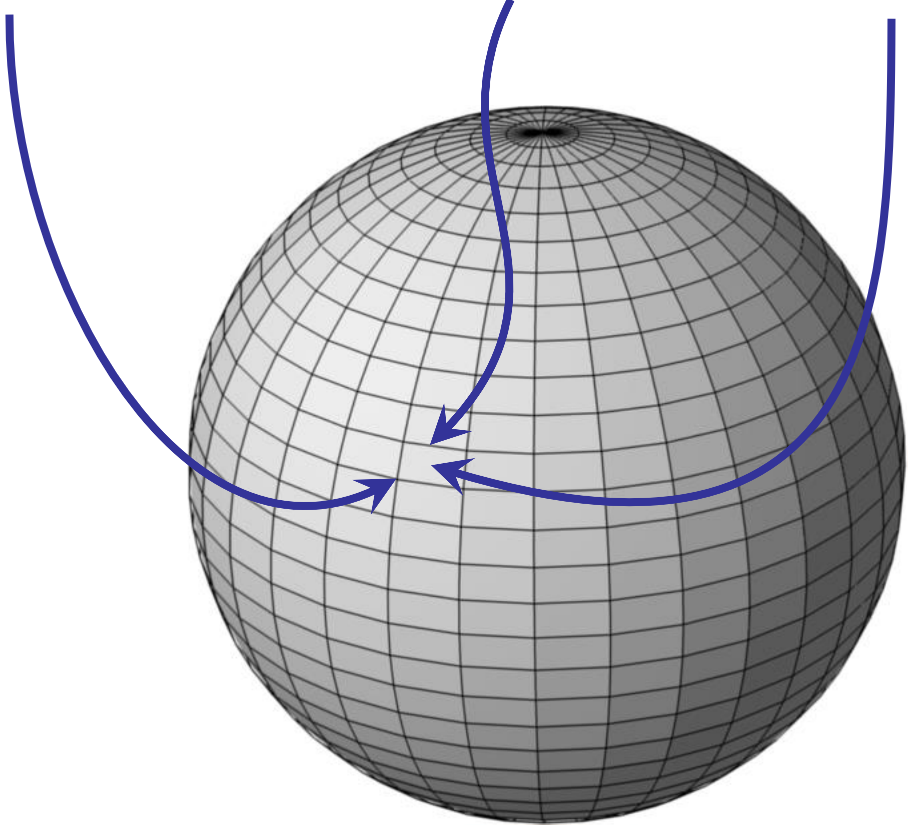
Introduction and Motivation

Mathematical Tools

Coordinate Systems

Introduction and Motivation

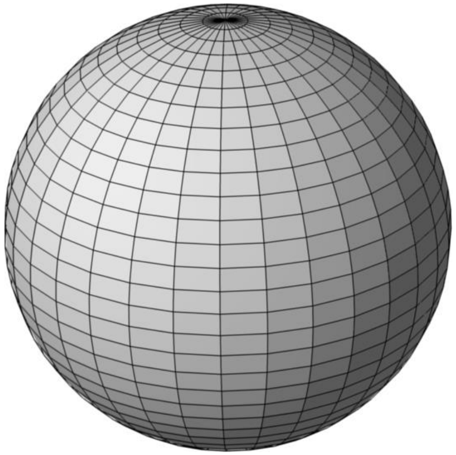
- 3D models (usually) represented as **points/vertices**, **edges**, **polygons**



Introduction and Motivation

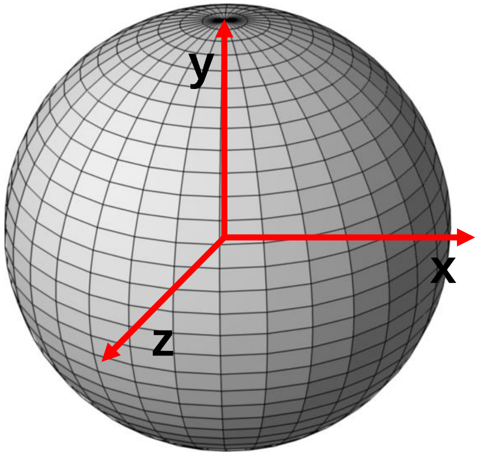
- 3D models (usually) represented as **points/vertices**, **edges**, **polygons** …

- … in 3D coordinate systems:
  - **object** coordinate systems
  - **world** coordinate system
  - **camera** coordinate system
  - **screen** coordinate system



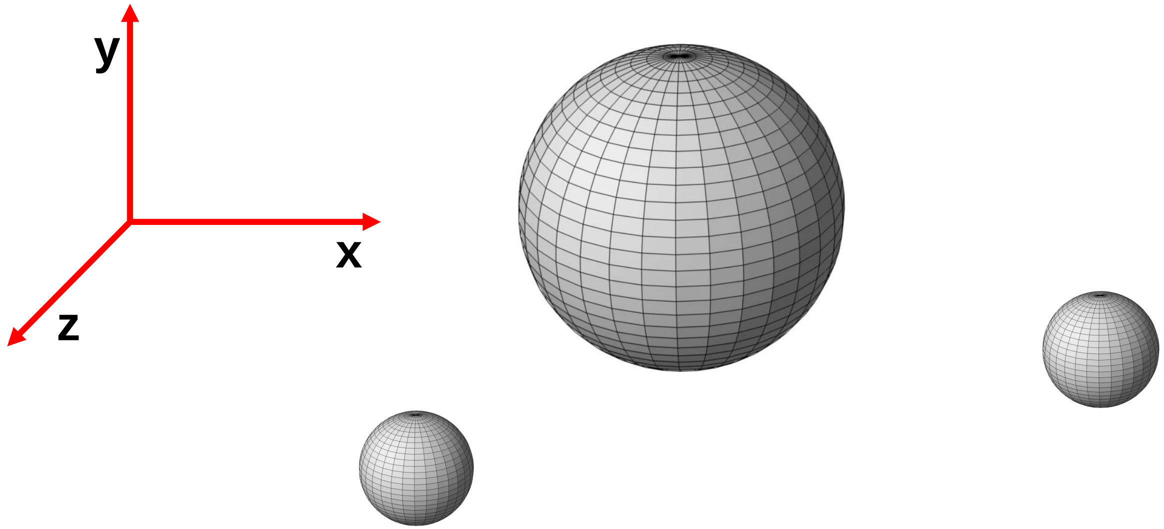
Introduction and Motivation

- … in 3D coordinate systems:
  - **object** coordinate systems



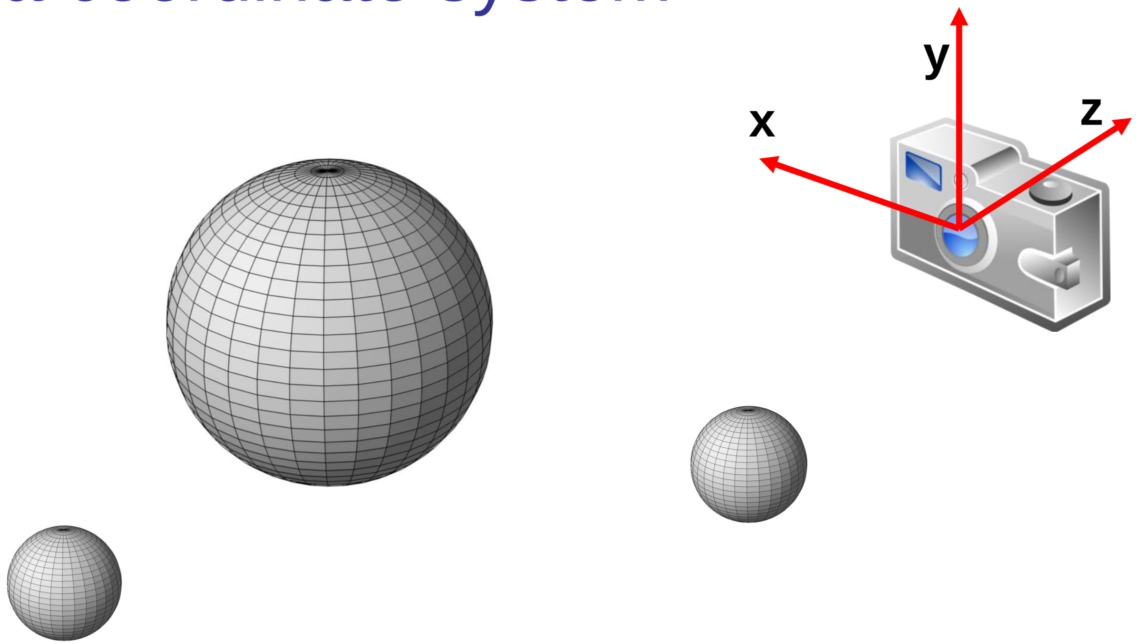
Introduction and Motivation

- … in 3D coordinate systems:
  - **world** coordinate system



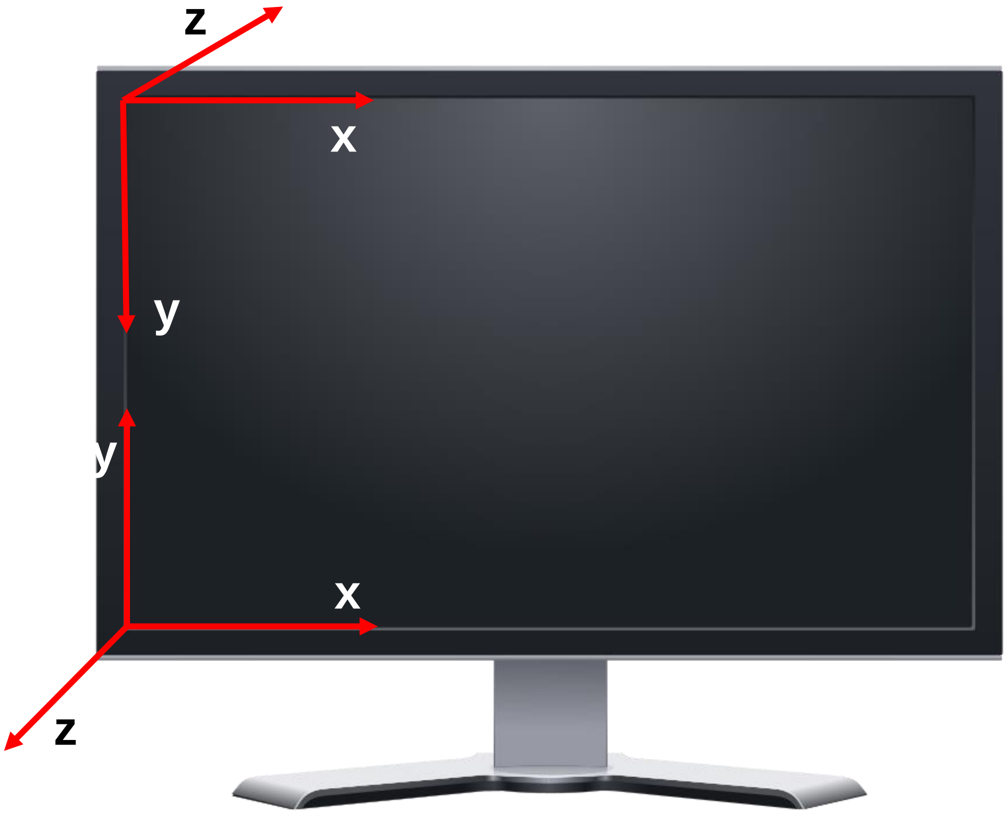
Introduction and Motivation

- … in 3D coordinate systems:
  - **camera** coordinate system



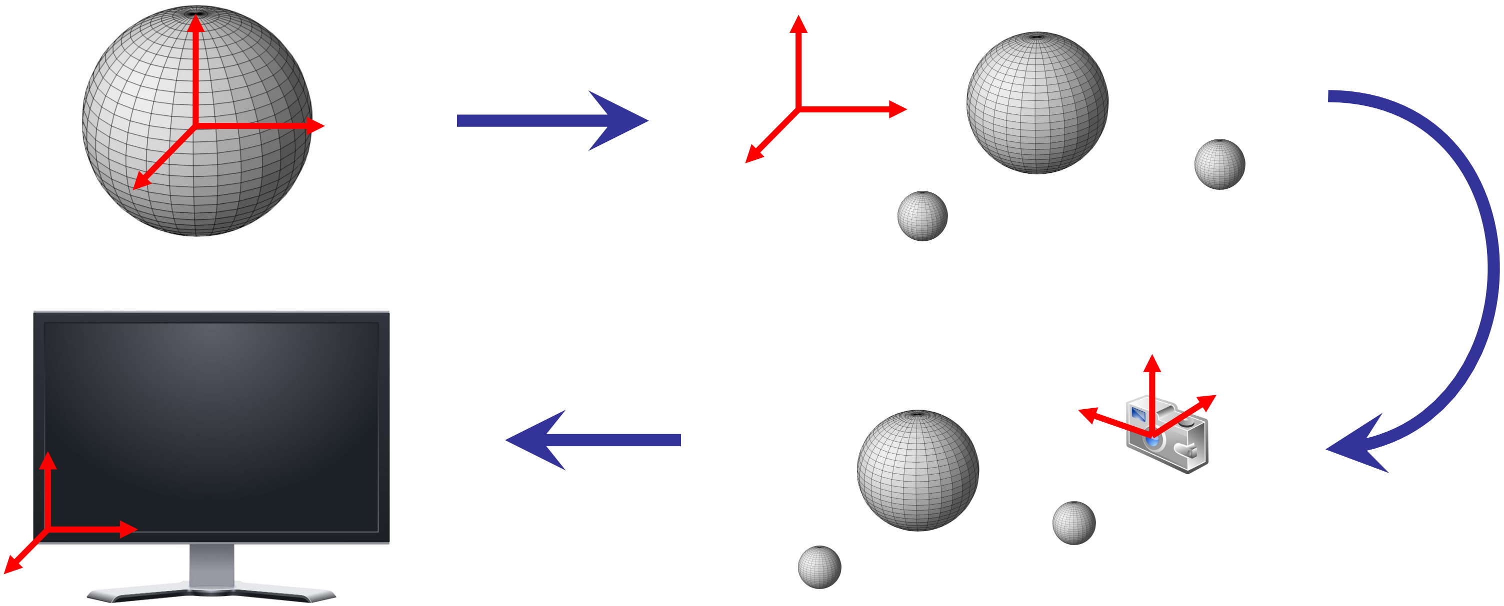
Introduction and Motivation

- … in 3D coordinate systems:
  - **screen** coordinate system



Introduction and Motivation

- transformations necessary



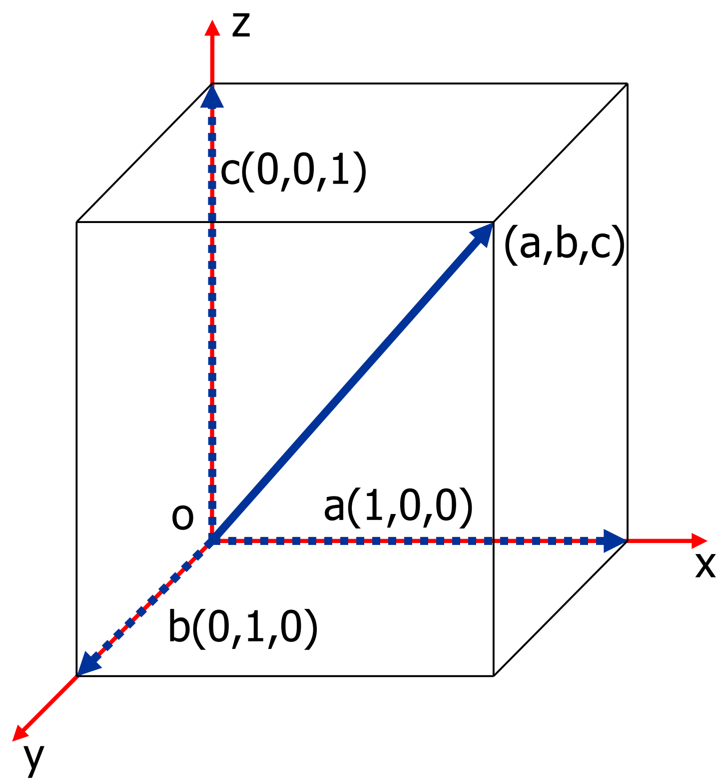
- foundation: geometry and linear algebra

Scalars, Points, Vectors, Matrices

- scalar values: real (rational) numbers: 7.39
- points: position in nD space: $(2.53, -1.78)^T$
- vectors: directions in nD space
  - have no position in space
  - is linear combination of basis vectors of the vector space, e.g., $(a, b, c)^T = a(1, 0, 0)^T + b(0, 1, 0)^T + c(0, 0, 1)^T$
- matrices $(n \times m)$: transformations between vector spaces

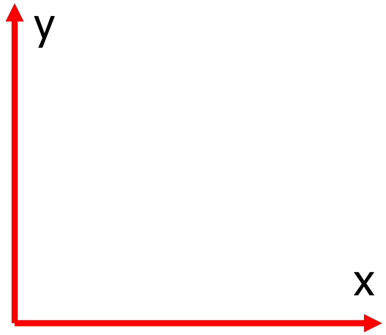
Coordinate System and Coordinates

- coordinate system: set $(o, e_1, e_2, ..., e_n)$ consisting of point $o \in A^n$ and basis $(e_1, e_2, ..., e_n)$ of vector space A^n
- position vector $v = (\overrightarrow{op})$ for each $p \in A^n$
- coordinates: scalar components of v with respect to $(e_1, e_2, ..., e_n)$ $v = x_1 e_1 + x_2 e_2 + ... + x_n e_n$
- o: point of origin

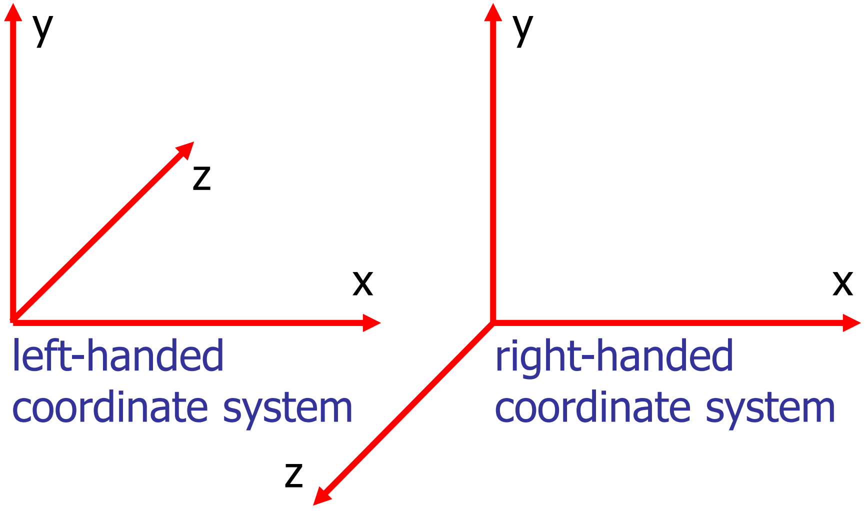


Coordinate Systems in CG: 2D & 3D

- two-dimensional



- three-dimensional
  - 2 mirrored systems
  - cannot be matched by rotation
  - we use right-handed



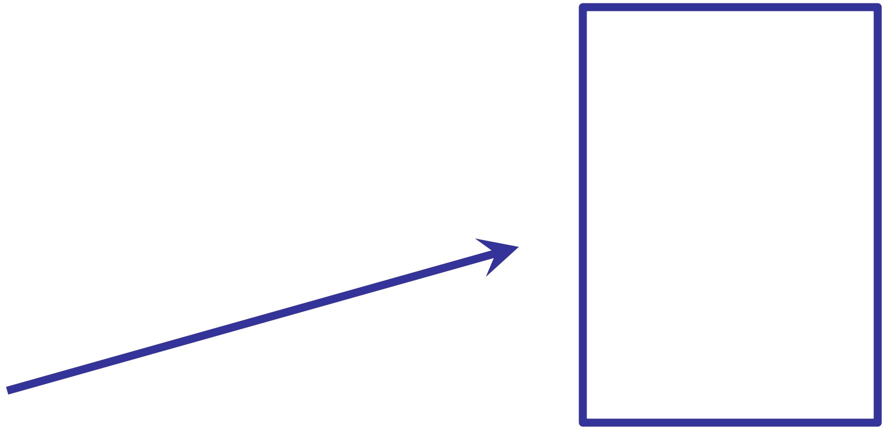
Transformations

in 2D:

Translation, Scaling, Rotation

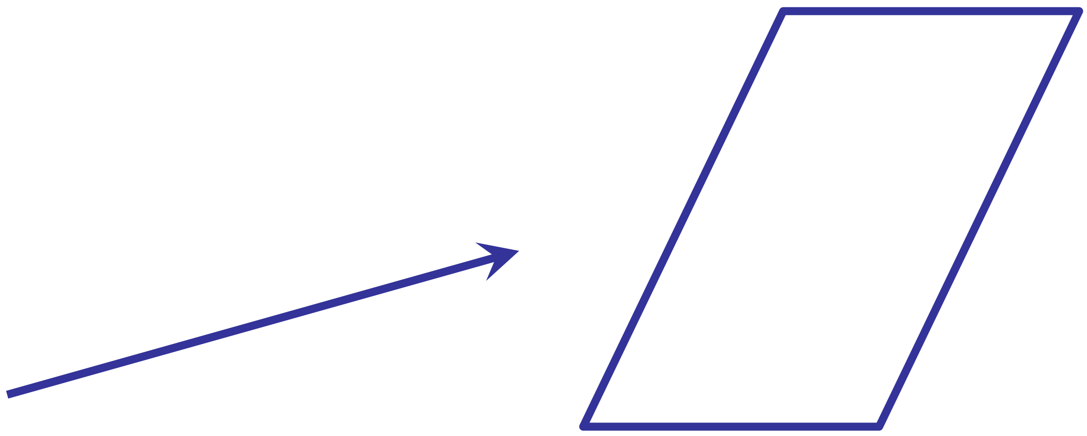
Transformations in 2D

- goal: represent changes and movements of objects in the vector space
- common transformations:
  - translation
  - rotation
  - scaling
  - mirroring
  - shearing
  - combinations of these



Transformations in 2D

- goal: represent changes and movements of objects in the vector space
- common transformations:
  - translation
  - rotation
  - scaling
  - mirroring
  - shearing
  - combinations of these

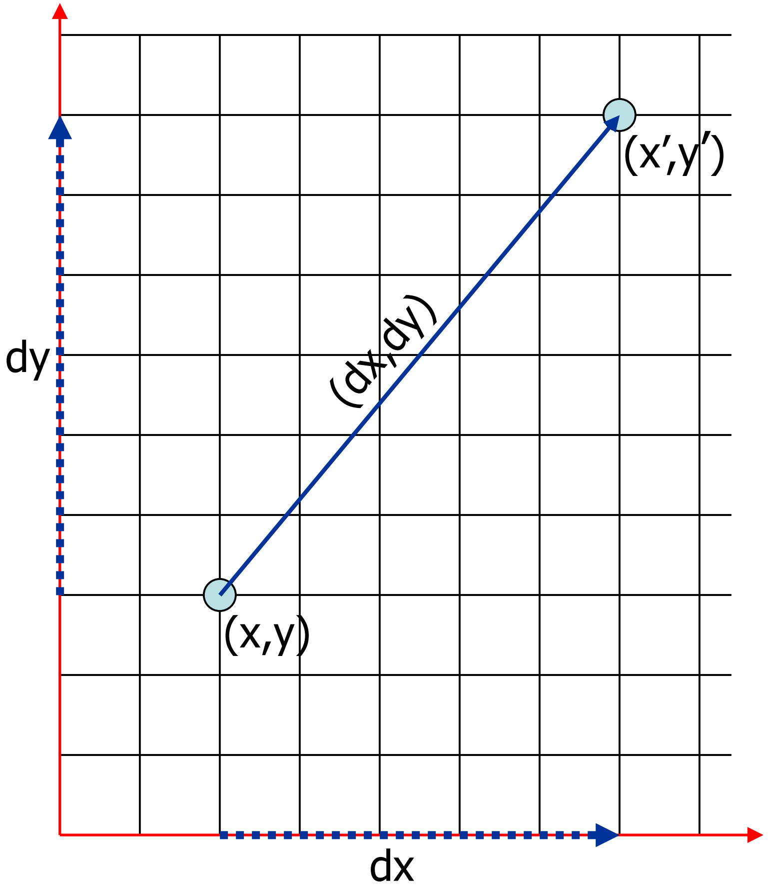


2D Translation

- move point $(x, y)^T$ on a straight line to $(x', y')^T$
- represent translation by a translation vector that is added

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} dx \\ dy \end{pmatrix}$$

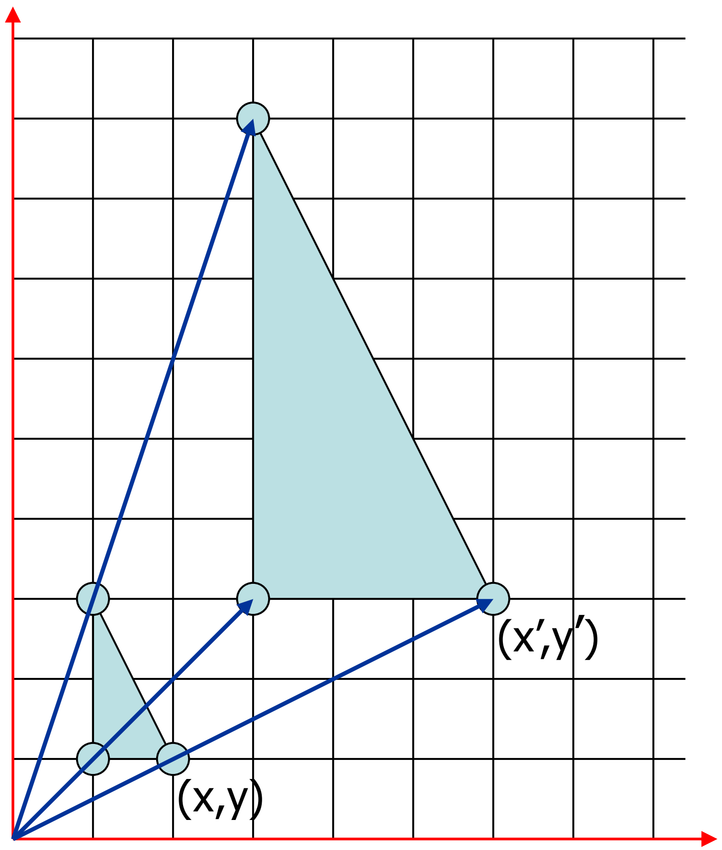
- vector: movement from one point to another



2D Uniform Scaling

- center of scaling is o
- scaling uniformly in all directions
- stretching of $(x, y)^T$'s position vector by scalar factor α to get $(x', y')^T$
- mathematically: multiplication with α

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \alpha \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \alpha x \\ \alpha y \end{pmatrix}$$

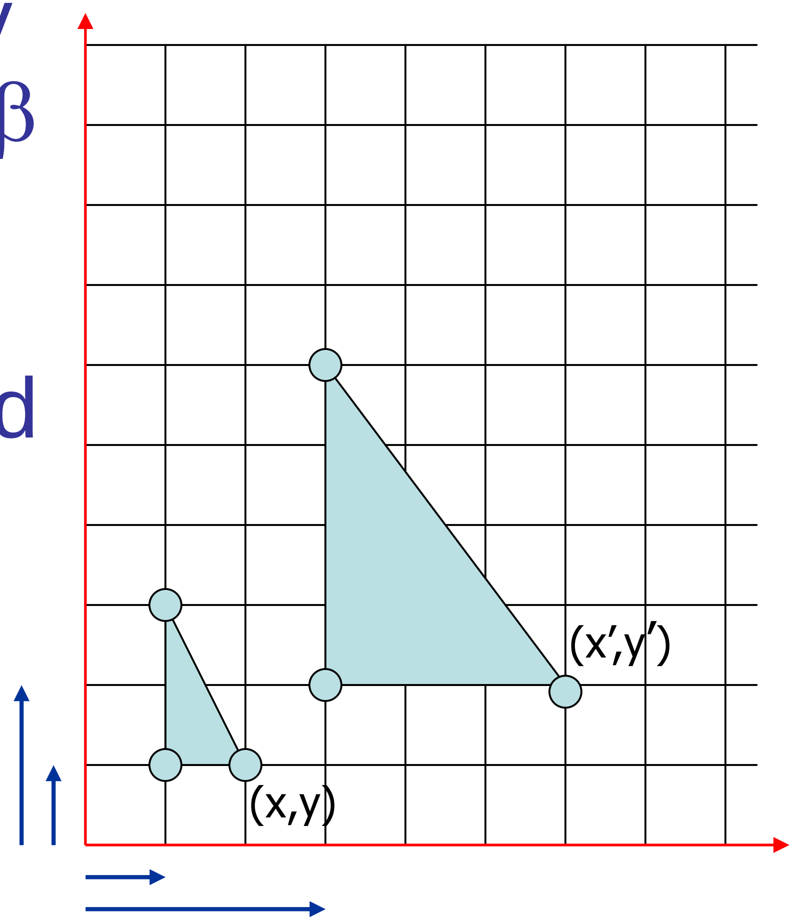


2D Non-Uniform Scaling

- center of scaling is o
- scaling in *x*-direction by α and in y-direction by β (scaling vector $(\alpha, \beta)^T$)
- mathematically: multiplication with α and β according to axis

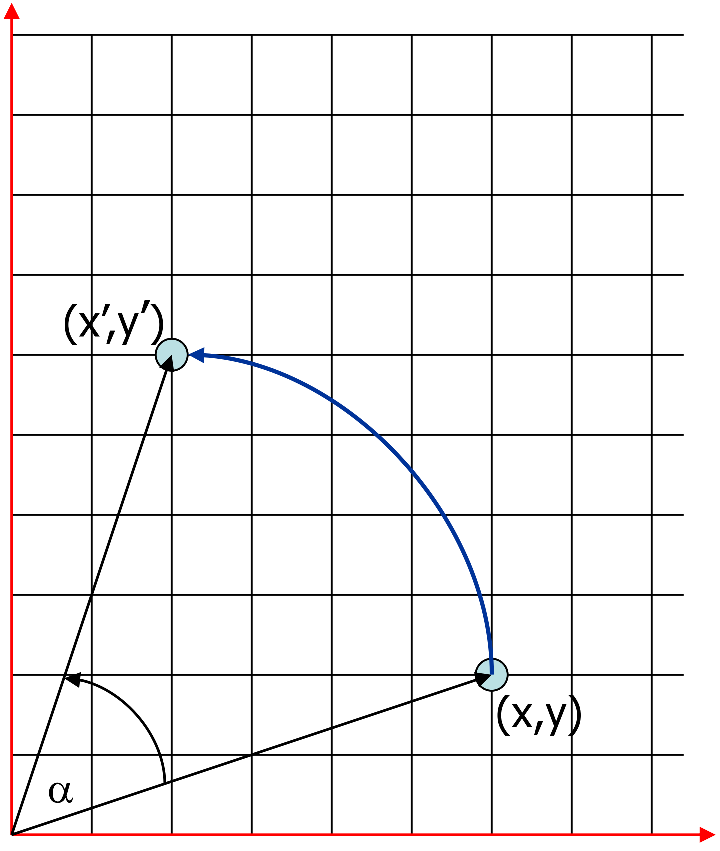
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \alpha x \\ \beta y \end{pmatrix}$$

- application: mirroring



2D Rotation

- center of rotation is o
- point $(x, y)^T$ is rotated by an angle α around o to obtain $(x', y')^T$
- positive angles α mean counter-clockwise rotation

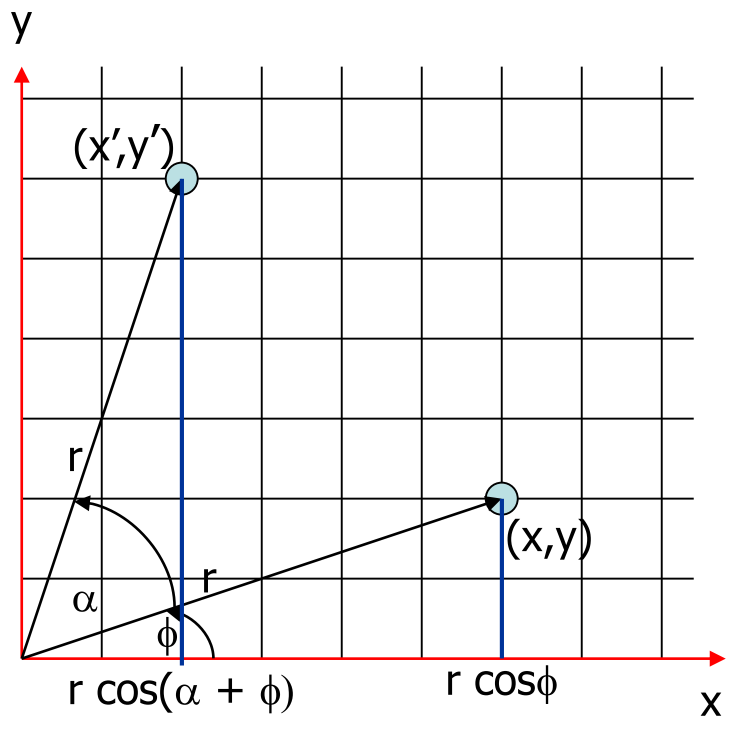


2D Rotation – Derivation of Formula

- distances $(x,y)^T - o$ & $(x',y')^T - o$ are both r
- $x = r\cos\phi$ and $y = r\sin\phi$
- $x' = r\cos(\alpha+\phi)$ and $y' = r\sin(\alpha+\phi)$
- addition theorems
- $x' = r\cos\alpha\cos\phi - r\sin\alpha\sin\phi$

  $\mathbf{= x\cos\alpha - y\sin\alpha}$
- $y' = r\sin\alpha\cos\phi + r\cos\alpha\sin\phi$

  $\mathbf{= x\sin\alpha + y\cos\alpha}$



2D Rotation

- $x' = x \cos\alpha - y \sin\alpha$
  $y' = x \sin\alpha + y \cos\alpha$
- can be expressed as matrix multiplication

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

- rotations around negative angles result in clockwise motion
- use symmetry of triginometric functions: $\cos(-\alpha)=\cos(\alpha)$ and $\sin(-\alpha)=-\sin(\alpha)$

Transformations in 2D: Results

- translation: addition of translation vector
  scaling: multiplication of factor(s)
  rotation: matrix multiplication
- problems:
  - non-uniform treatment of transformations
  - no way to combine N transformation into one
- idea: all transformations as matrix multiplications!
- only scaling and translation to do

Transformations

in 2D:

Matrix Formulation

Homogeneous Coordinates

Scaling using 2D Matrix

- general 2D matrix multiplication format

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

- scaling formula

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \alpha x \\ \beta y \end{pmatrix}$$ (possibly with $\alpha = \beta$)

- scaling as matrix multiplication

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Translation using 2D Matrix

- general matrix multiplication format

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

- tanslation formula

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} dx \\ dy \end{pmatrix}$$

- translation as matrix multiplication

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} ? & ? \\ ? & ? \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$ not possible in 2×2 matrix! ☹

Transformations in 2D: What now?

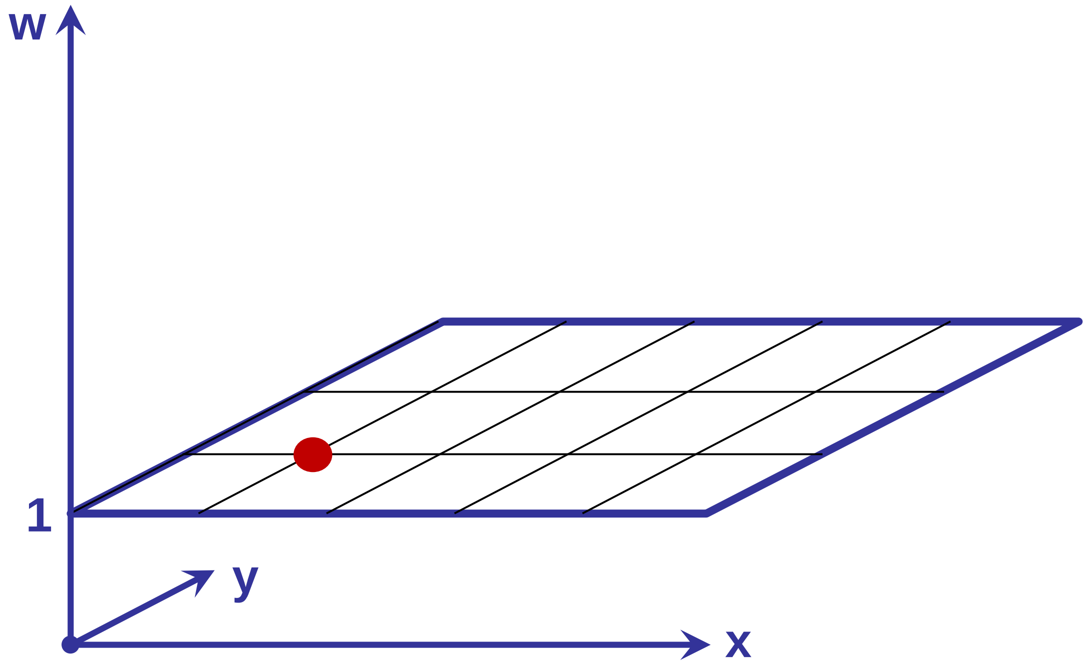
- translation: addition of translation vector
  scaling: matrix multiplication
  rotation: matrix multiplication
- only scaling and rotation possible as matrix multiplication
- possible solutions
  - don't use translations?
  - use both matrices and vector adding?
  - mathematical magic:
    homogeneous coordinates!

Homogeneous Coordinates in 2D

- add an additional dimension to our vector space A^n: $n \rightarrow n+1$
- $(x, y)^T$ represented as $(wx, wy, w)^T$, $w \neq 0$
- normalized using $w = 1 \rightarrow (x, y, 1)^T$
- homogeneous coordinates are not to be confused with "regular" 3D coordinates!

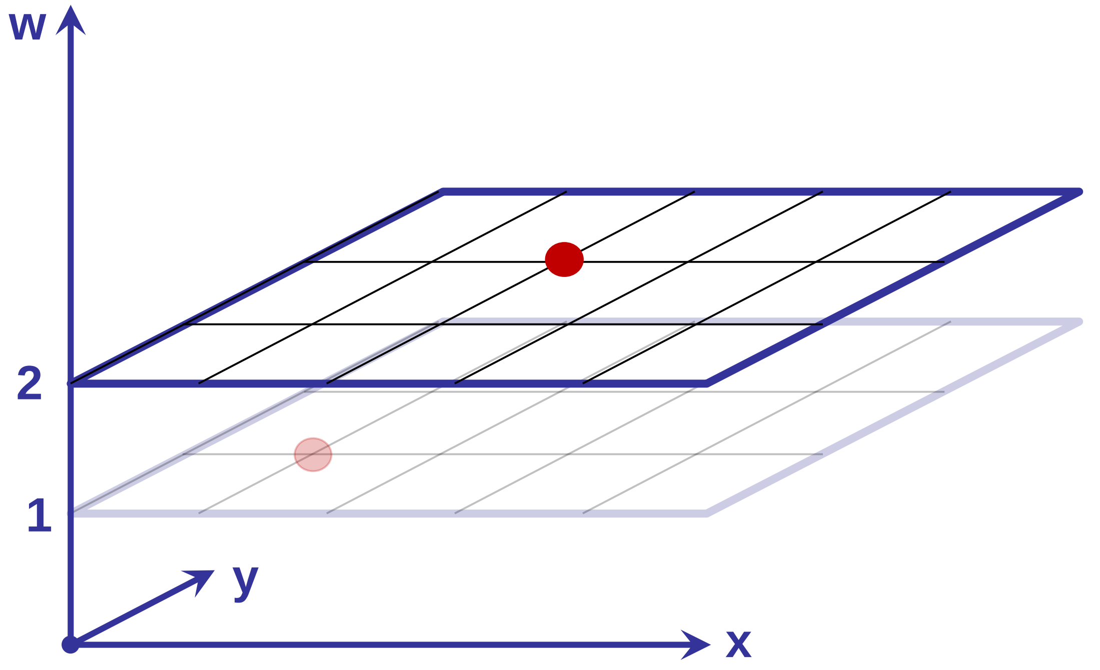
Homogeneous Coordinates in 2D

- each point in A^n is equivalent to a line in homogeneous space A^{n+1} that originates in the origin o but without including o



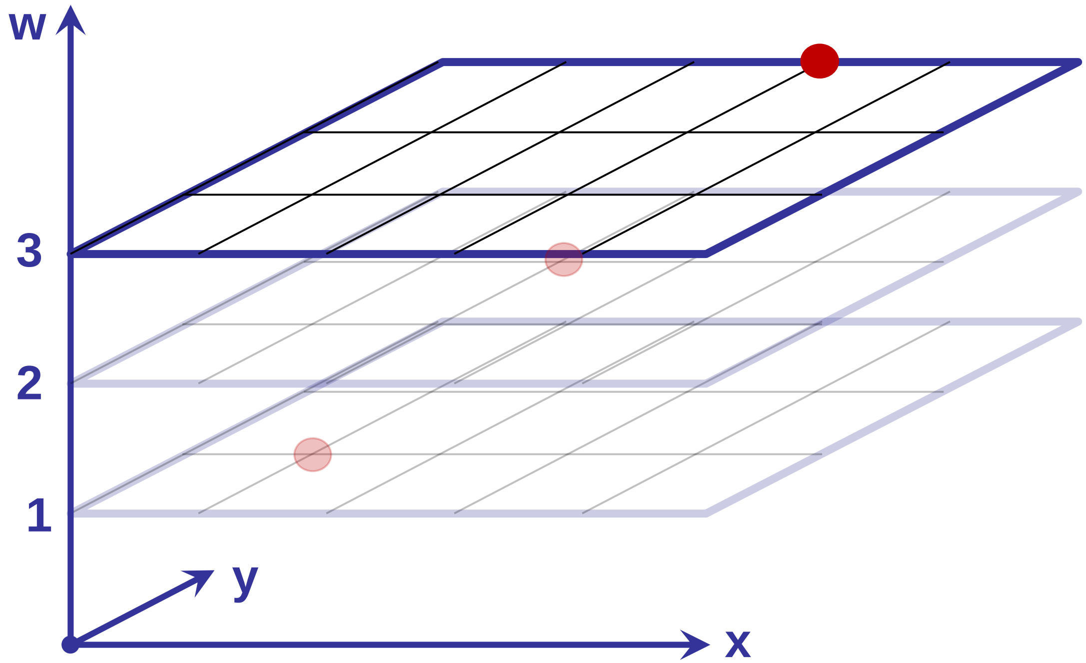
Homogeneous Coordinates in 2D

- each point in A^n is equivalent to a line in homogeneous space A^{n+1} that originates in the origin o but without including o



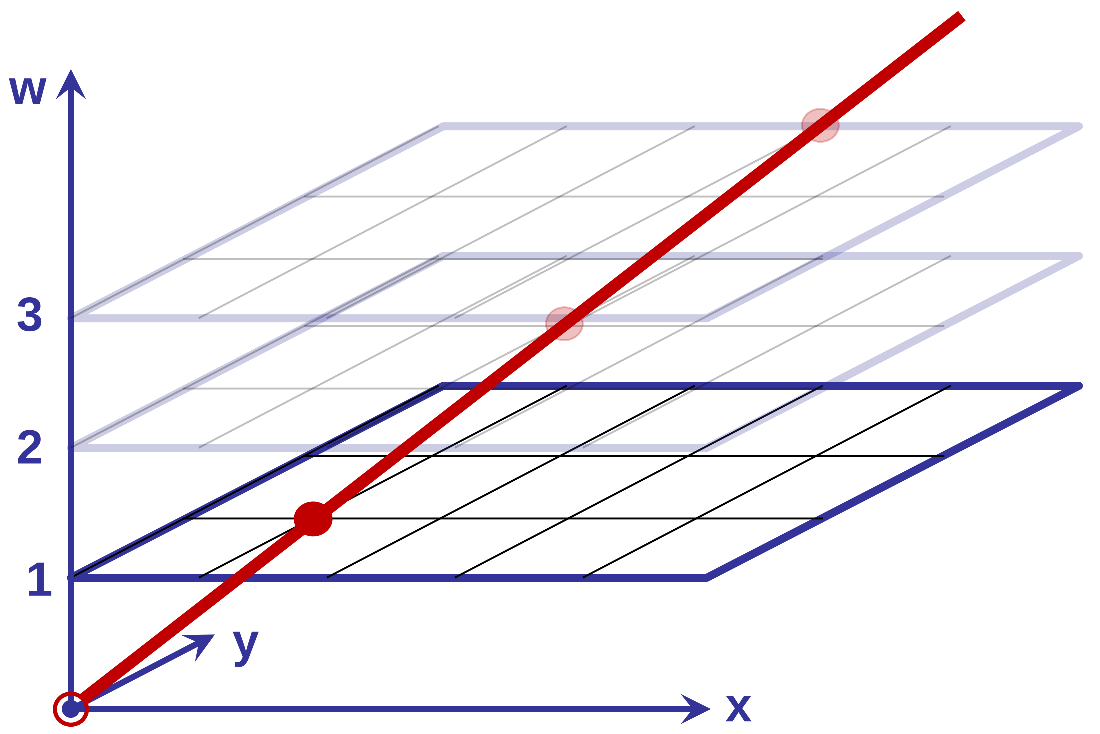
Homogeneous Coordinates in 2D

- each point in A^n is equivalent to a line in homogeneous space A^{n+1} that originates in the origin o but without including o



Homogeneous Coordinates in 2D

- each point in A^n is equivalent to a line in homogeneous space A^{n+1} that originates in the origin o but without including o



Homogeneous Coordinates in 2D

- advantages of homogeneous coordinates
  - uniform treatment of transformations
  - all transformations can be represented
  - combined transformations as one matrix
- procedure: matrix-vector multiplication

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} x \\ y \\ w \end{pmatrix}$$

- goal: derive transformation matrices

Translation in Homogeneous Coords

- general matrix multiplication format

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} x \\ y \\ w \end{pmatrix}$$

- translation formula

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} dx \\ dy \end{pmatrix}$$

- translation as matrix multiplication

$$\begin{pmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Scaling in Homogeneous Coords

- scaling as matrix multiplication

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

- scaling as homogeneous matrix multiplication

$$\begin{pmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Rotation in Homogeneous Coords

- rotation as matrix multiplication

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

- rotation as homogeneous matrix multiplication

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

Homogeneous Coordinates in 2D

- general transformation matrix

$$\begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix}$$

scaling

rotation

translation

- what about normalizing w’ when done?
  – in all basic transformations we get w’ = 1
  – no normalization necessary

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Transformations

in 2D:

Inverse Transformations

Concatenating Transformations

Inverse Transformations

- how to reverse basic transformations?
- translation: using negative vector
  $T^{-1}(dx, dy) = T(-dx, -dy)$
- scaling: using inverted factor $^1/_\alpha$
  $S^{-1}(\alpha) = S(^1/_\alpha)$
- rotation: using negative angle
  $R^{-1}(\alpha) = R(-\alpha)$

  also: rotation matrices are orthonormal
  $R^{-1} = R^T$

Concatenating Transformations

- two translations

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & dx_2 \\ 0 & 1 & dy_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & dx_1 \\ 0 & 1 & dy_1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \Rightarrow$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & dx_1 + dx_2 \\ 0 & 1 & dy_1 + dy_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- translations are additive
- result is a translation by the sum of both vectors

Concatenating Transformations

- two scalings

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha_2 & 0 & 0 \\ 0 & \beta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha_1 & 0 & 0 \\ 0 & \beta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \Rightarrow$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha_1\alpha_2 & 0 & 0 \\ 0 & \beta_1\beta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- scaling is multiplicative
- result is a scaling
  by the product of both factors

Concatenating Transformations

- two rotations

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\alpha_2 & -\sin\alpha_2 & 0 \\ \sin\alpha_2 & \cos\alpha_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\alpha_1 & -\sin\alpha_1 & 0 \\ \sin\alpha_1 & \cos\alpha_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \Rightarrow$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\alpha_1 + \alpha_2) & -\sin(\alpha_1 + \alpha_2) & 0 \\ \sin(\alpha_1 + \alpha_2) & \cos(\alpha_1 + \alpha_2) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

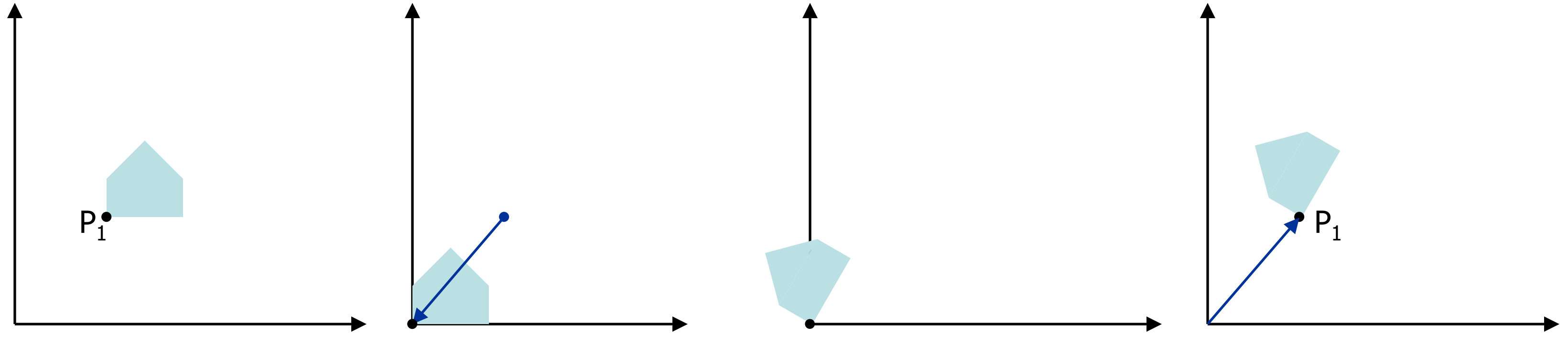
- rotation is additive (addition theorems)
- result is a rotation around the sum of both angles

Concatenating Transformations

- in general, concatenating transformations by multiplying their homogenous matrices
- given n transformations T_1, T_2, ..., T_n
- all are to be concatenated so that T_1 is processed first, T_2 second, ..., and T_n last
- $P' = T_n \cdot \ldots \cdot T_2 \cdot T_1 \cdot P$
- only for column vectors (points)!
  row vectors (points): $P' = P \cdot T_1 \cdot T_2 \cdot \ldots \cdot T_n$
- we use column vectors (points)

Using Concatenated Transformations

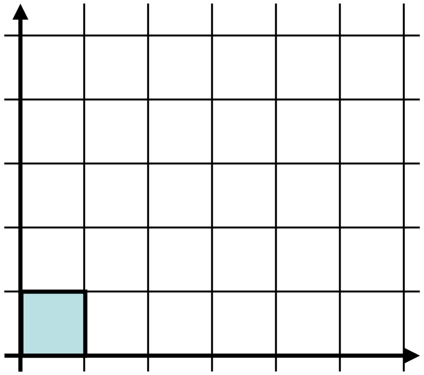
- example: rotation around arbitrary point P_1
- three steps:
  - translation into origin
  - rotation
  - inverse translation to bring back to P_1
- $P' = T_{+} \cdot R \cdot T_{-} \cdot P$



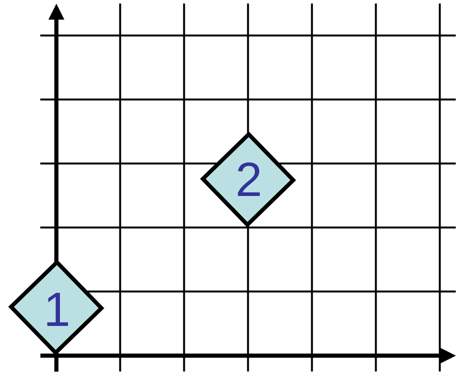
Using Concatenated Transformations

- all matrices for one concatenated transformation can be multiplied into one (as preprocessing)
- but: matrix multiplication is not commutative! the order matters! (as with normal transformations, too)
- $T_n \ldots T_2 T_1 P \neq T_1 T_2 \ldots T_n P \neq T_2 T_n \ldots T_1 P$ in most cases (depending on T_i)
- some transformations are commutative (just translating; just scaling; just rotating)

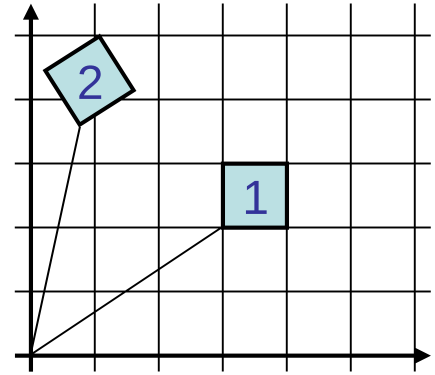
Concatenating Transformations



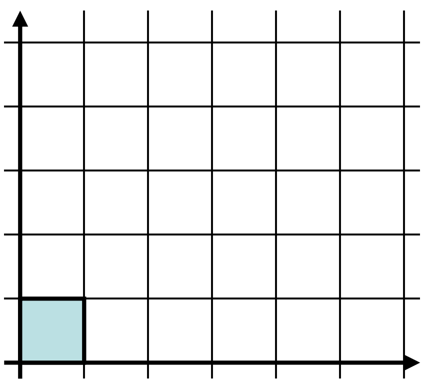
Rotation and Translation



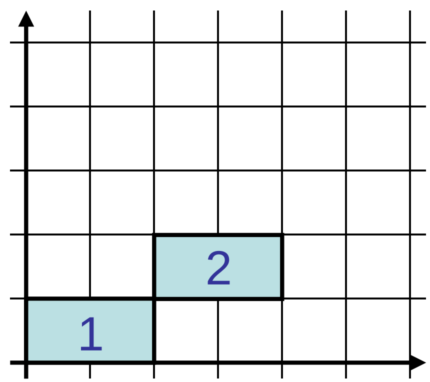
P’ = T·R·P



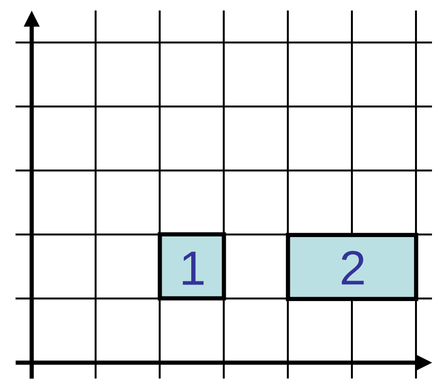
P’ = R·T·P



Scaling and Translation

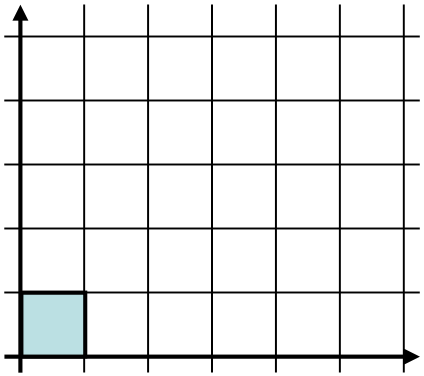


P’ = T·S·P

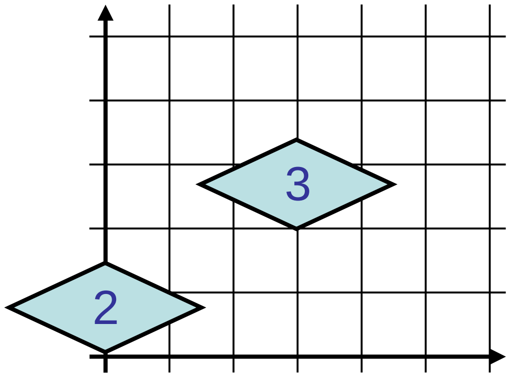


P’ = S·T·P

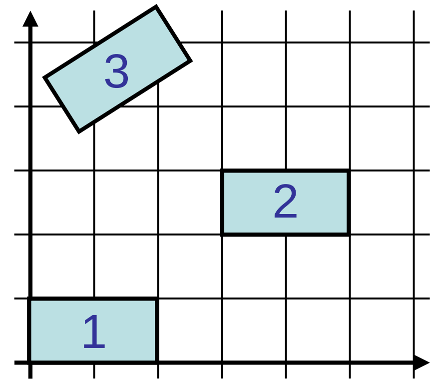
Concatenating Transformations



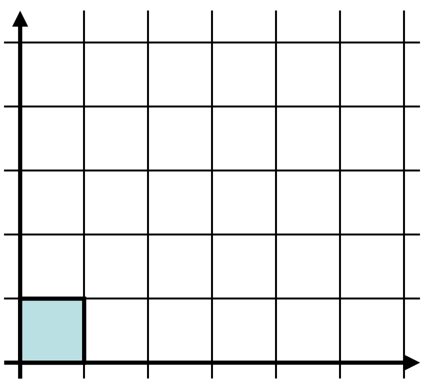
Rotation, Scaling, Translation



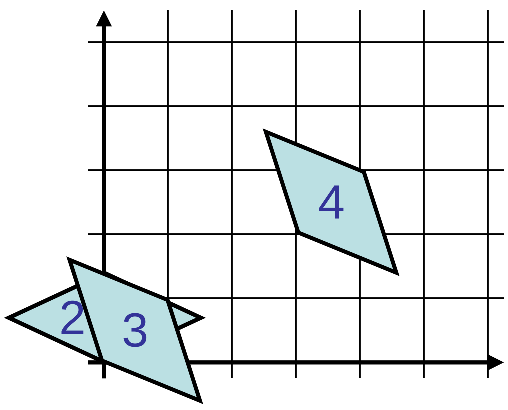
$P' = T \cdot S \cdot R \cdot P$



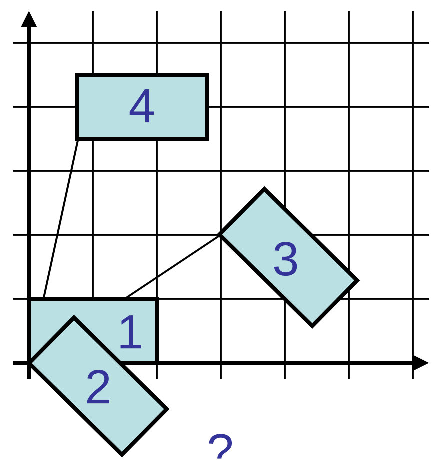
$P' = R \cdot T \cdot S \cdot P$



Rot., Scal., neg. Rot., Trans.



$P' = T \cdot R_{-} \cdot S \cdot R_{+} \cdot P$



?

$P' = R_{+} \cdot T \cdot R_{-} \cdot S \cdot P$

Concatenating Transformations

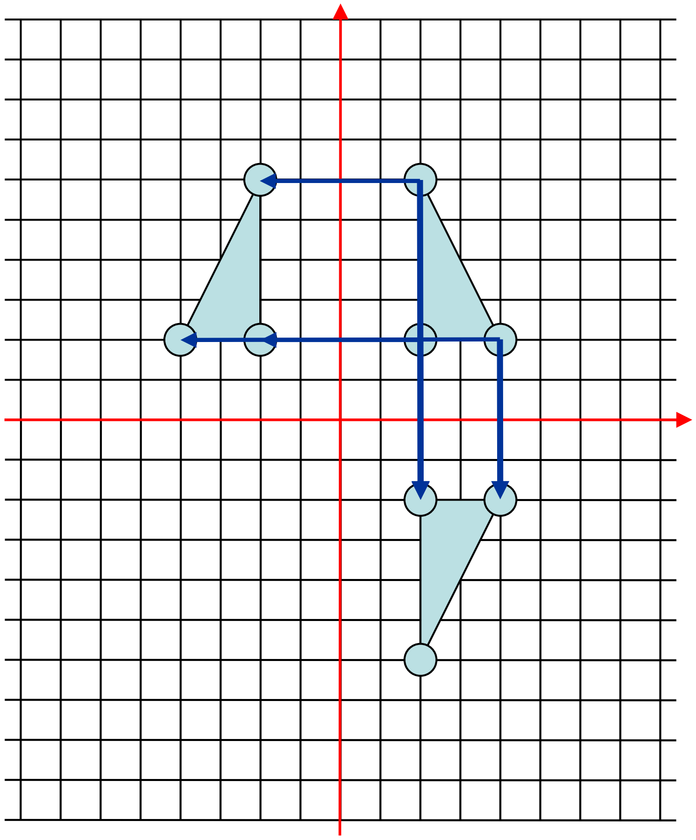
- efficiency example
  - model with 1,000,000 vertices
  - operations: scaling, rotation, translation
  - individual matrix multiplications:
    3 x (9M + 6A) x 1,000,000 = 27,000,000M
    18,000,000A
  - one concatenated matrix :
    (9M + 6A) x 1,000,000 = 9,000,000M
    6,000,000A

Other Transformations in 2D

- mirroring
- non-uniform scaling using -1 & 1 as factors

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

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



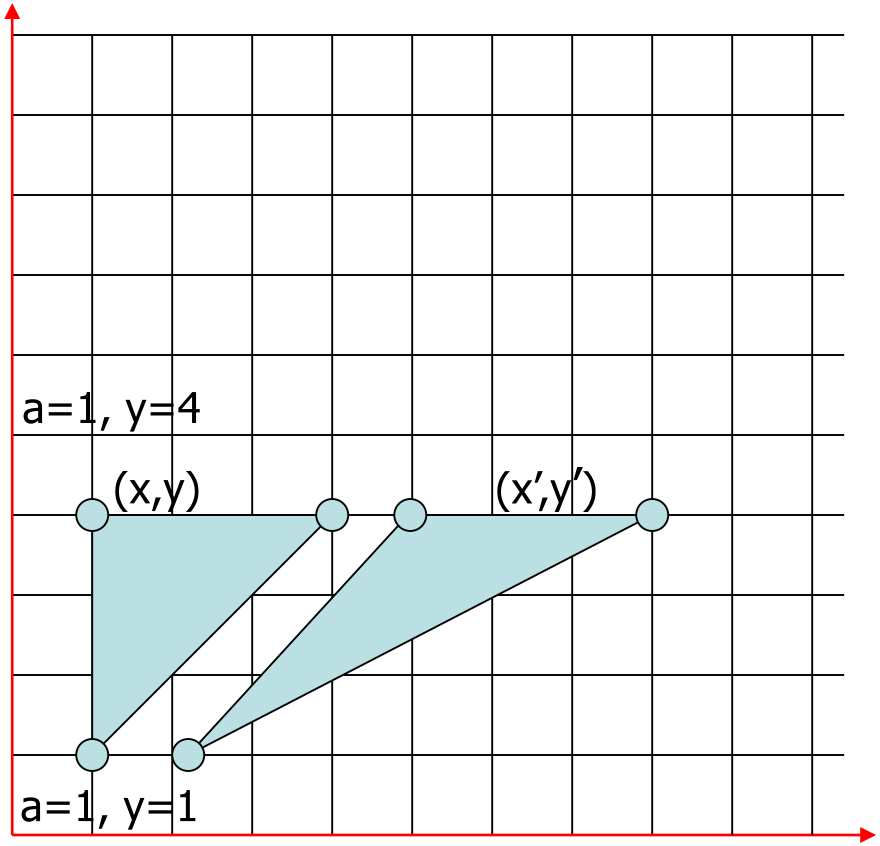
Other Transformations in 2D

- shearing
- in *x*-direction

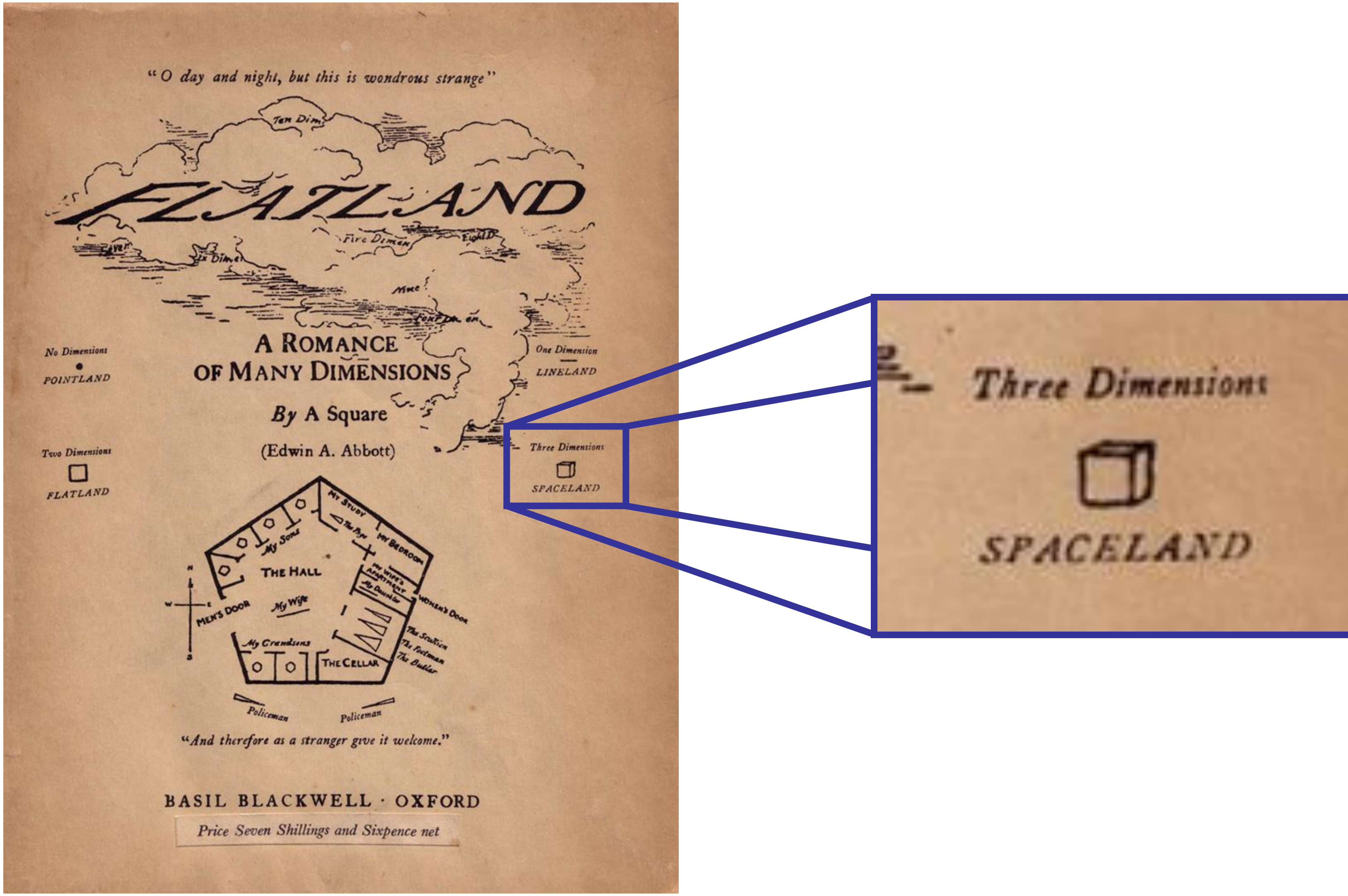
$$T = \begin{pmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} x + ay \\ y \\ 1 \end{pmatrix}$$

- in *y*-direction

$$T = \begin{pmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y + bx \\ 1 \end{pmatrix}$$



Escaping Flatland

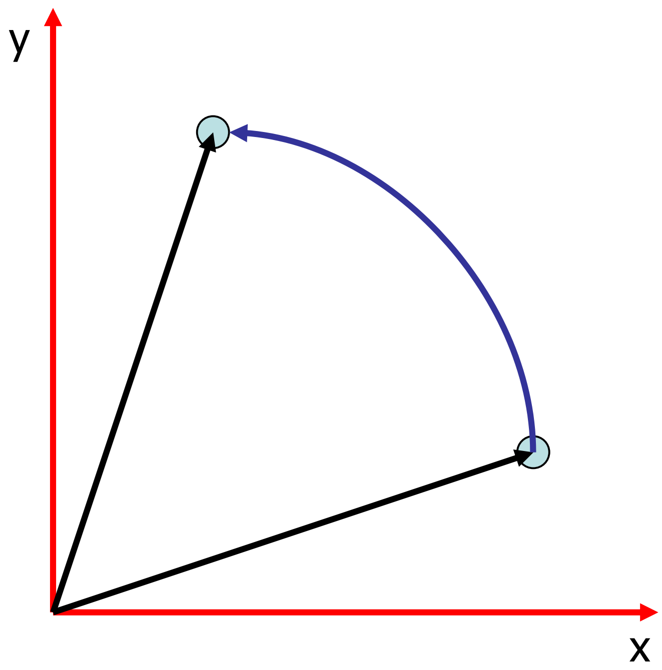


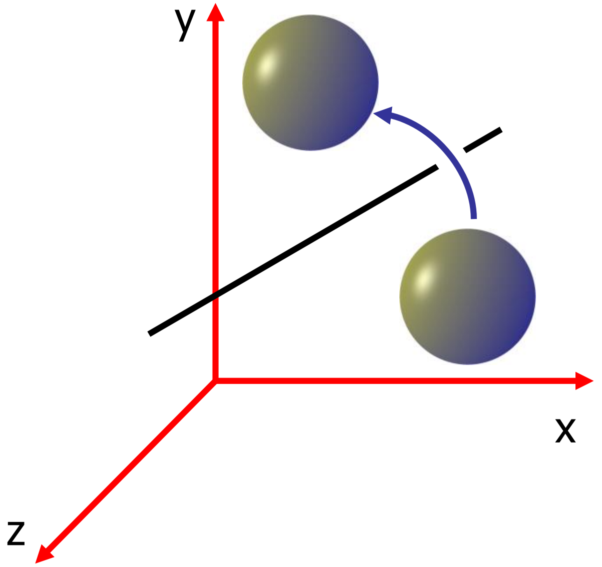
Computer Graphics

3D Geometric Transformations

Contents

- transformations *continued*
  - transitioning from 2D to 3D





Geometric Transformations in 3D

- same approach as in 2D
- also use homogeneous coordinates (for the same reasons)
- vectors/points from 3D to 4D

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \to \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

- transformation matrices are now 4×4 (instead of 3×3)

Transformation Matrices in 3D

- translation
  - translation vector $(dx, dy, dz)^T$
- scaling
  - for uniform scaling $s_x = s_y = s_z$
  - otherwise individual factors may differ
  - mirroring using factors of -1 and 1 depending on the mirror plane

$$T = \begin{pmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$S = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Transformation Matrices in 3D

- rotation
  - rotation around 3 axes possible now
  - each has individual rotation matrix
  - rotation around positive angles in right-handed coordinate system
  - rotation axis stays unit vector in matrix

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

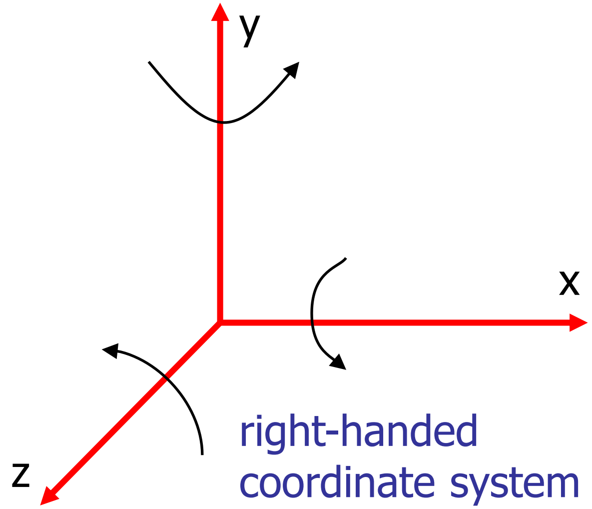
$$R_y = \begin{pmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_z = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

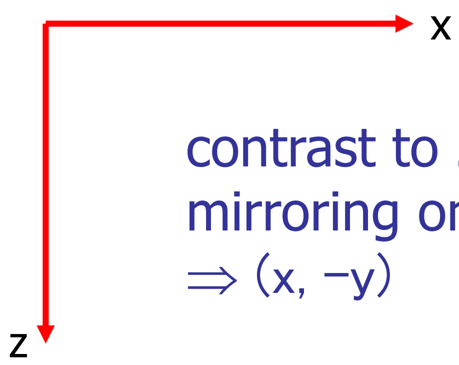
Transformation Matrices in 3D

- rotation
  - why are signs different from 2D case for R_y?

$$R_y = \begin{pmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



right-handed coordinate system



contrast to 2D case:
mirroring on *x*-axis
$\Rightarrow$ (x, −y)

Concatenating Transformations in 3D

- matrix multiplication just as in 2D
- general transformation matrix in 3D

$$T = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix}$$

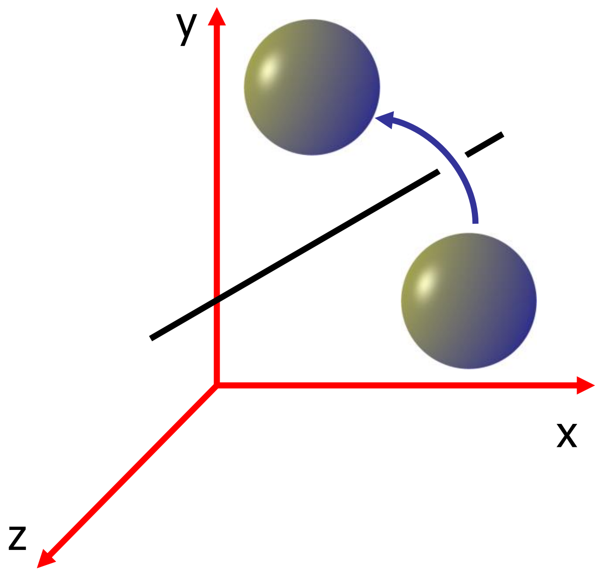
scaling

rotation

translation

Concatenating Transformations in 3D

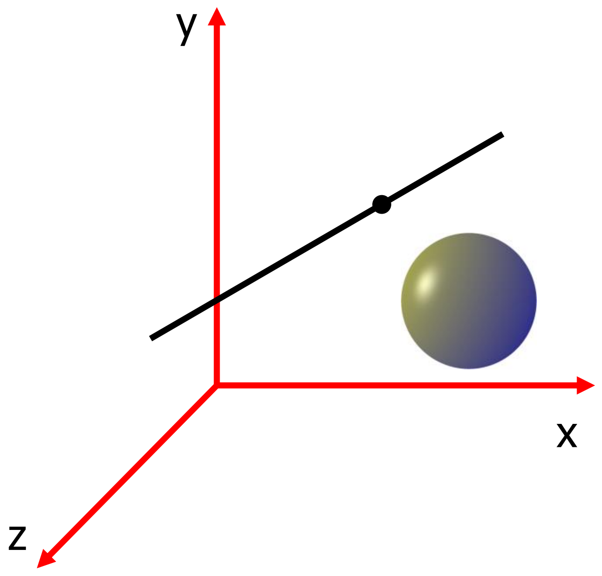
- rotation around arbitrary axis in 3D



Concatenating Transformations in 3D

- rotation around arbitrary axis in 3D

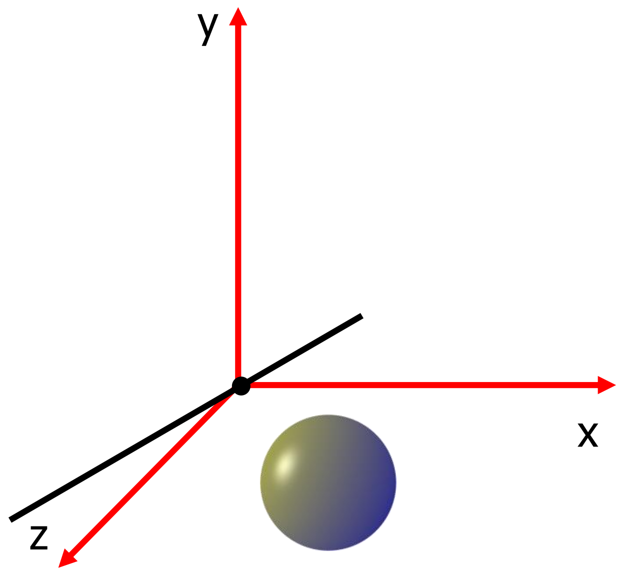
P' = **P**



Concatenating Transformations in 3D

- rotation around arbitrary axis in 3D

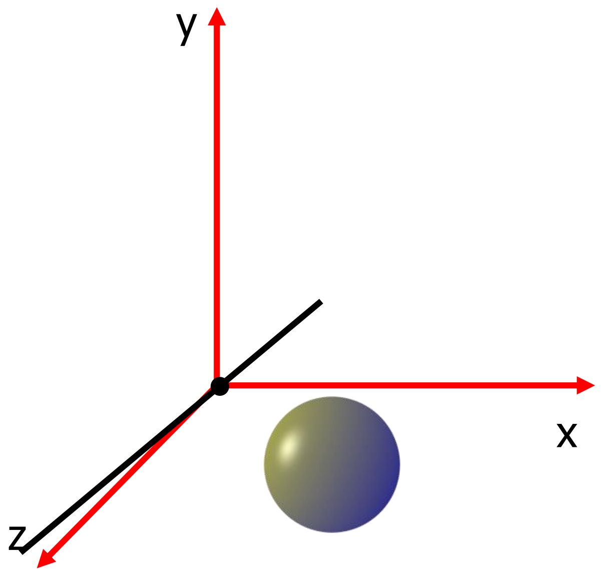
$$P' = \qquad\qquad T_{-} \bullet P$$



Concatenating Transformations in 3D

- rotation around arbitrary axis in 3D

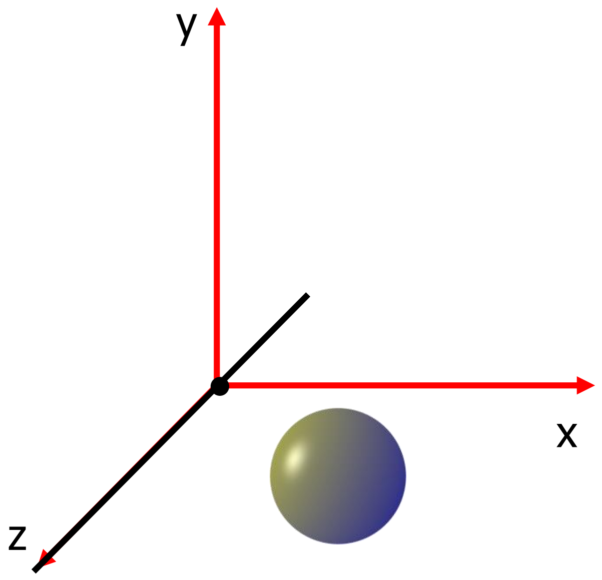
$$P' = \qquad R_y \bullet T_- \bullet P$$



Concatenating Transformations in 3D

- rotation around arbitrary axis in 3D

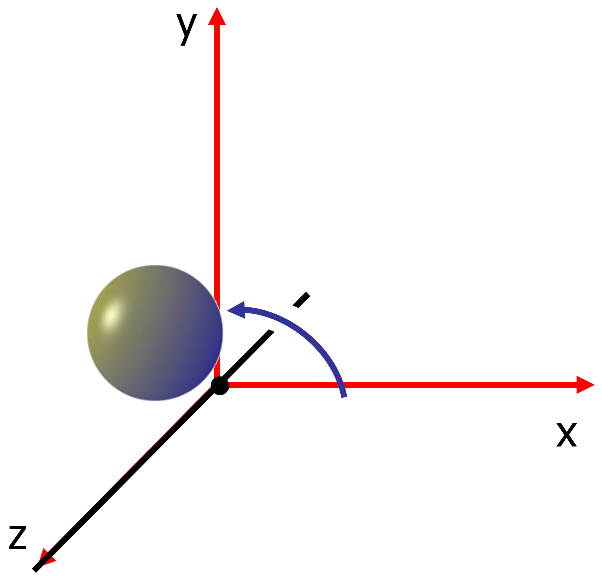
$$P' = \qquad R_x \bullet R_y \bullet T_- \bullet P$$



Concatenating Transformations in 3D

- rotation around arbitrary axis in 3D

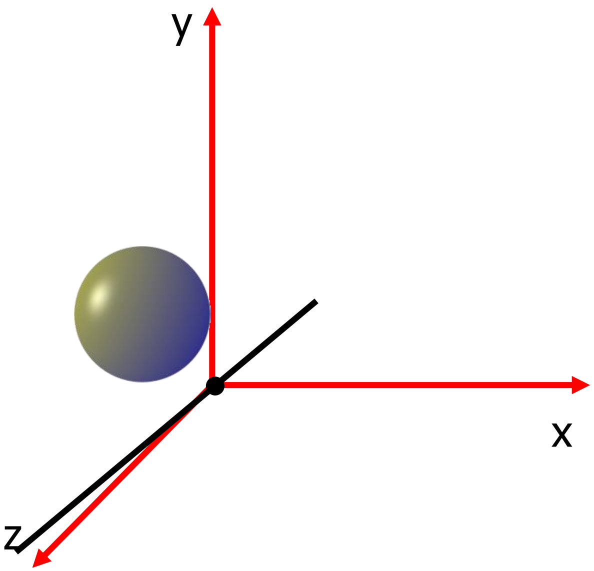
$$P' = \quad R_R \bullet R_x \bullet R_y \bullet T_- \bullet P$$



Concatenating Transformations in 3D

- rotation around arbitrary axis in 3D

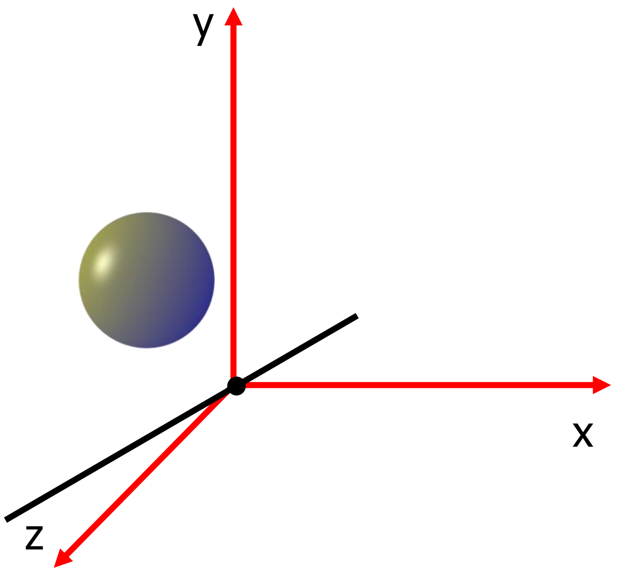
$$P' = R_{-x} \bullet R_R \bullet R_x \bullet R_y \bullet T_- \bullet P$$



Concatenating Transformations in 3D

- rotation around arbitrary axis in 3D

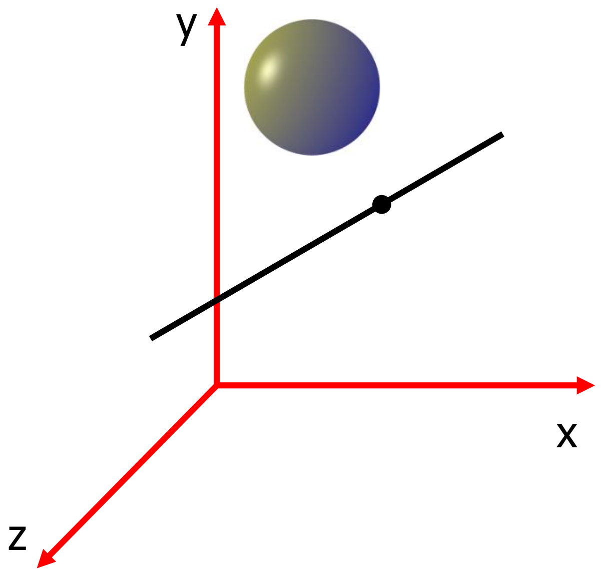
$$P' = \quad R_{-y} \bullet R_{-x} \bullet R_R \bullet R_x \bullet R_y \bullet T_- \bullet P$$



Concatenating Transformations in 3D

- rotation around arbitrary axis in 3D

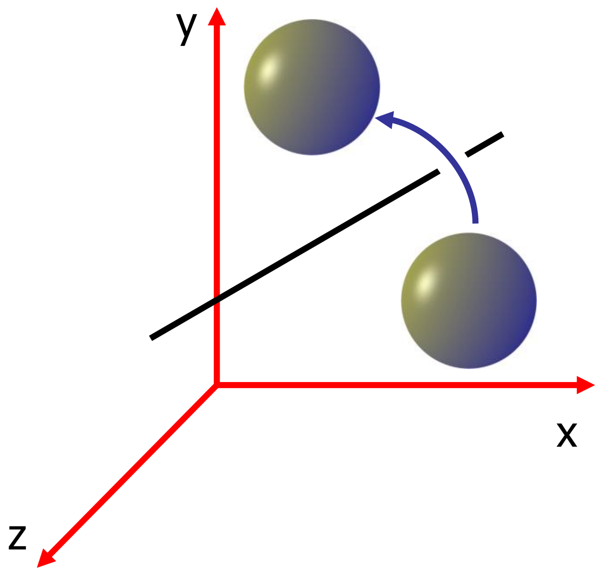
$$P' = T_{+} \bullet R_{-y} \bullet R_{-x} \bullet R_{R} \bullet R_{x} \bullet R_{y} \bullet T_{-} \bullet P$$



Concatenating Transformations in 3D

- rotation around arbitrary axis in 3D

$$P' = T_{+} \bullet R_{-y} \bullet R_{-x} \bullet R_{R} \bullet R_{x} \bullet R_{y} \bullet T_{-} \bullet P$$



Transformations: Summary

- geometric transformations:
  linear mapping from $\Re^n$ to $\Re^n$
- we are interested in
  $\Re^2 \rightarrow \Re^2$ and $\Re^3 \rightarrow \Re^3$
- transformations most relevant for CG:
  - translation
  - rotation
  - scaling
  - mirroring
  - shearing

Transformations: Summary

- unified representation of geometric transformations as matrices in homogeneous coordinates
- concatenation of transformation by multiplying the respective matrices
- order matters: for **column** vectors the **first** transformation comes **last** in the sequence
- concatenated transformations can be pre-computed (saving run-time)