FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Lecture #5

Affine Transformations

Computer Graphics
Winter Term 2016/17

Marc Stamminger / Roberto Grosso

Content

- Affine Transformations
  - Definition
  - Subclasses
  - Translations, Rotations, Scalings, Shearings, …
- Simple Projections
- Homogeneous coordinates

Affine Transformations

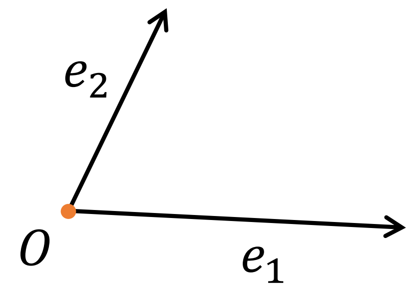
- Important in CG
  - Positioning objects in a scene
  - Object Animations
  - Changing the shape of objects
  - Creation of multiple copies of objects
  - Projection for virtual cameras
  - Changing between coordinate systems
  - Camera Animations

- Applications in CG will be handled in next chapter
- In this chapter we will look at the basics

Affine Transformations

- Coordinate Frames
  - Origin O (point)
  - Coordinate axes e_1, e_2 (vectors)

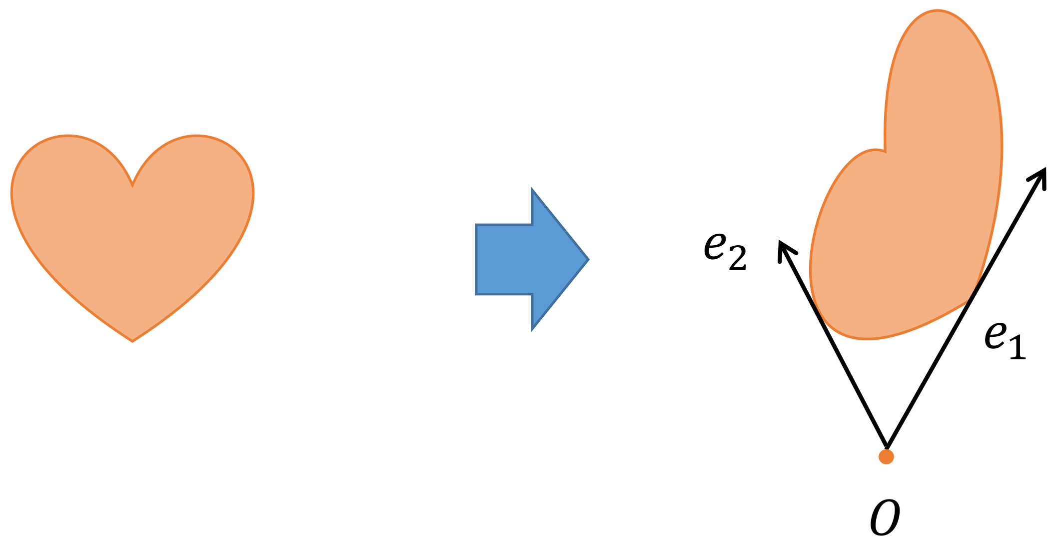
- Standard coordinate frame
  - $O = (0,0)$
  - $e_1 = (1,0), e_2 = (0,1)$



Affine Transformations

- Coordinate system change:

$$f(x,y) = O + xe_1 + ye_2$$



Affine Transformations

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

- We call such mappings *Affine Mappings*:

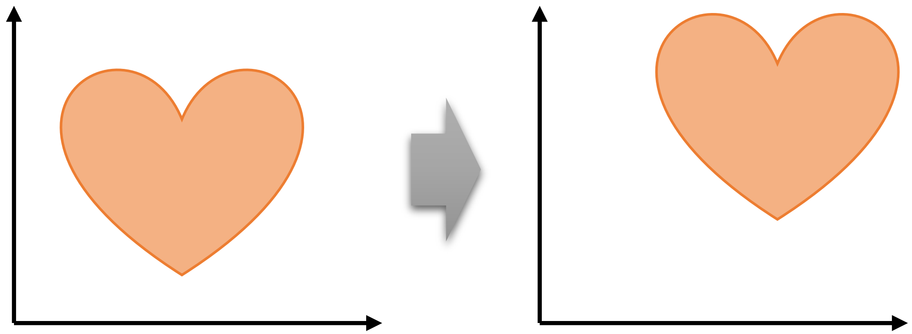
$$(x, y) \to \begin{pmatrix} e_1 & e_2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} O_1 \\ O_2 \end{pmatrix}$$

- Or, more generally:

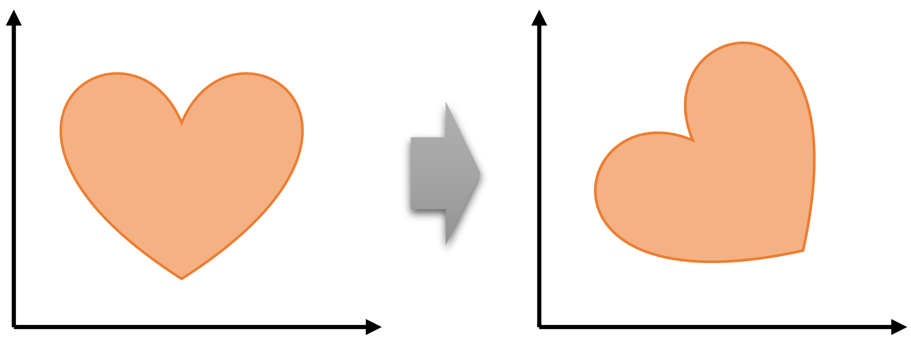
$$x \to Ax + t \qquad\qquad (A \in \mathbb{R}^{2\times 2}, t \in \mathbb{R}^2)$$

Affine Transformations

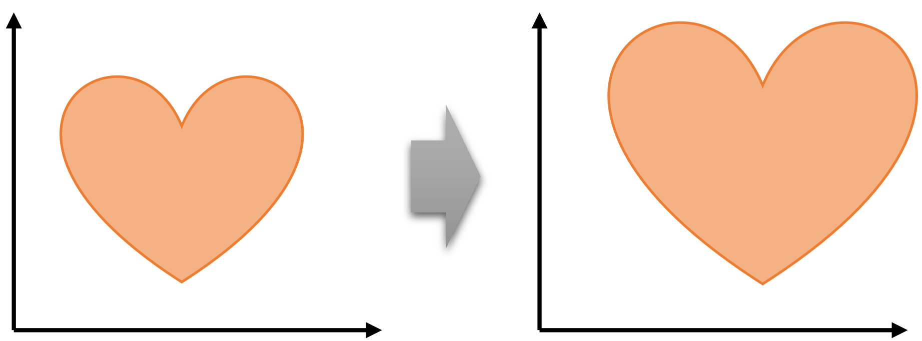
- Special cases
  - Translations



  - Rotations



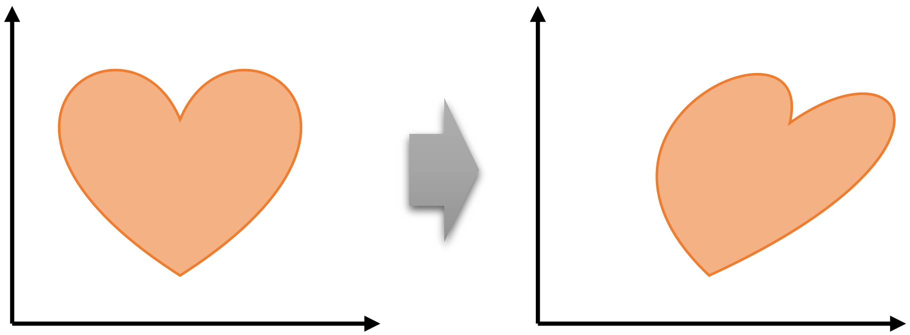
  - Scalings



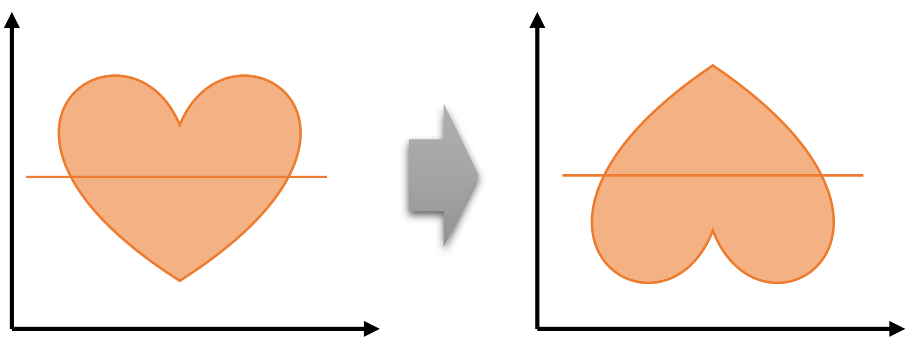
Affine Transformations

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

- Special cases
  - Shearings



  - Reflections



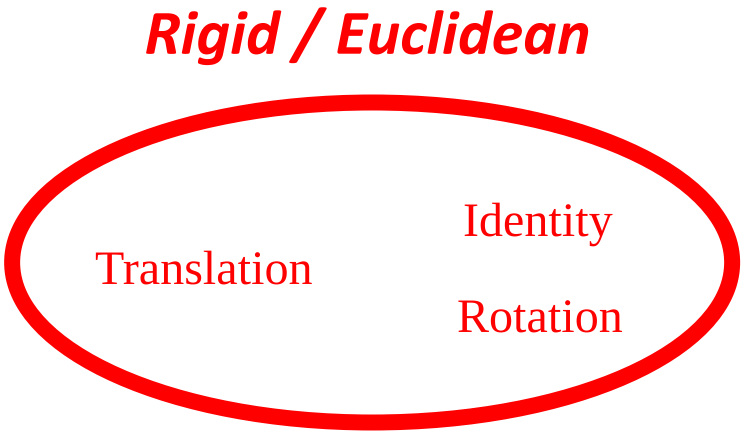
Affine Transformations

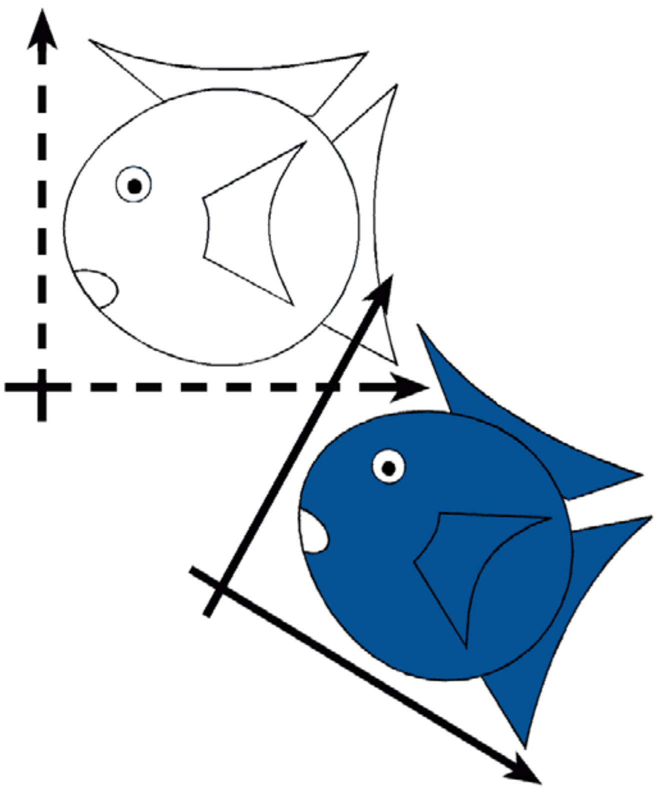
FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

- Classes of Affine Transformations
  - Rigid
  - Similarity
  - Linear

Affine Transformations

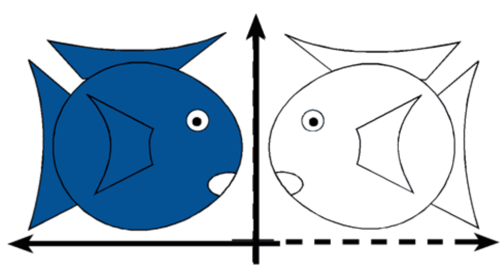
- Rigid Transformation (Euclidean Transform)
  - Preserves distances
  - Preserves angles





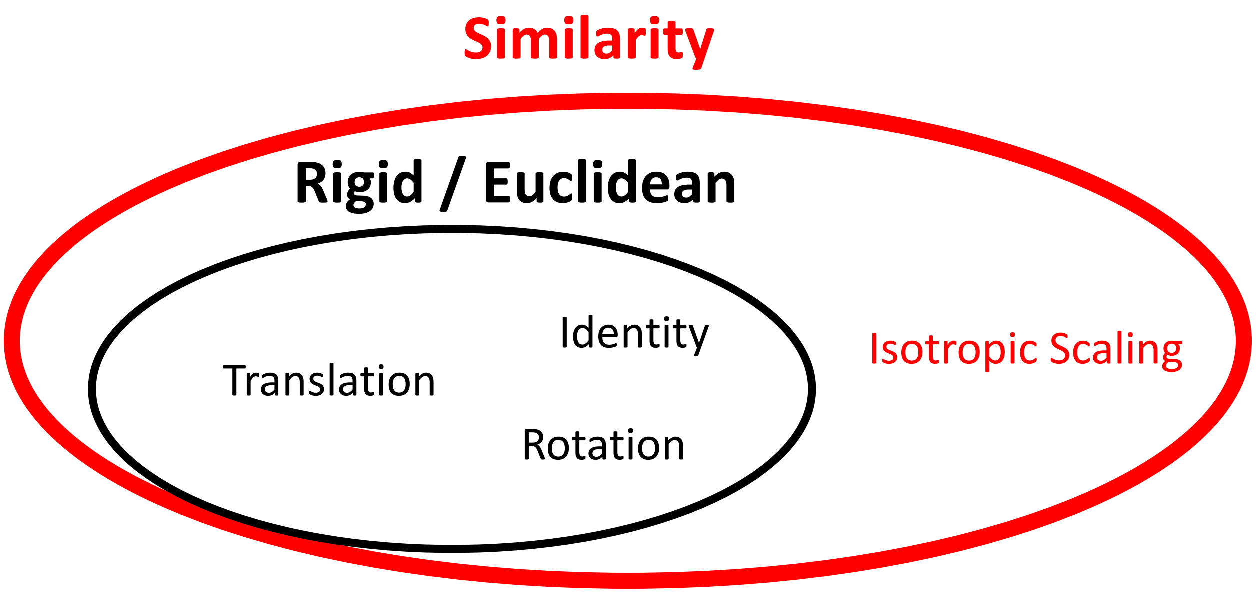
Affine Transformations

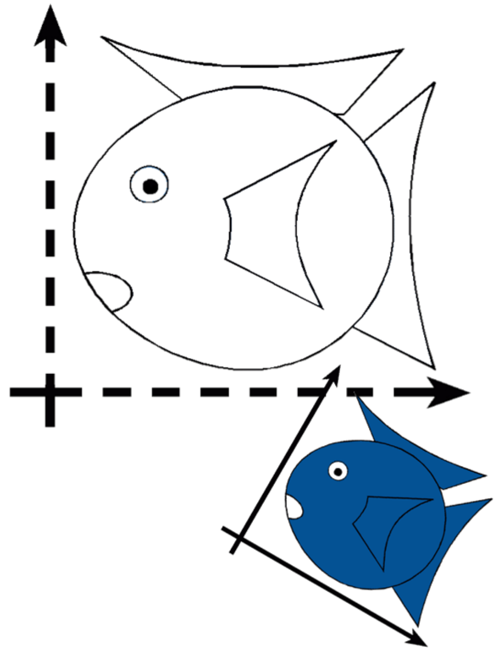
- Rigid transformation:
  e_1 and e_2 remain orthogonal and keep unit length
- $x \rightarrow Ax + t$ with A orthogonal and $\det(A) > 0$
- Application of multiple rigid transformations is a rigid transformation again (also true for following classes)
- If $\det(A) < 0$, A contains a reflection, which is not rigid



Affine Transformations

- Similarity Transforms
  - Preserves angles, but changes distances
  - Rigid + (isotropic) scaling + reflection



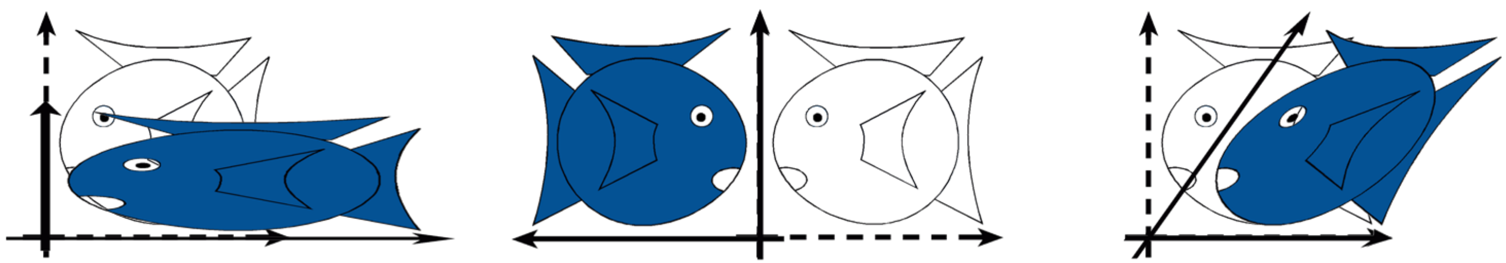


Affine Transformations

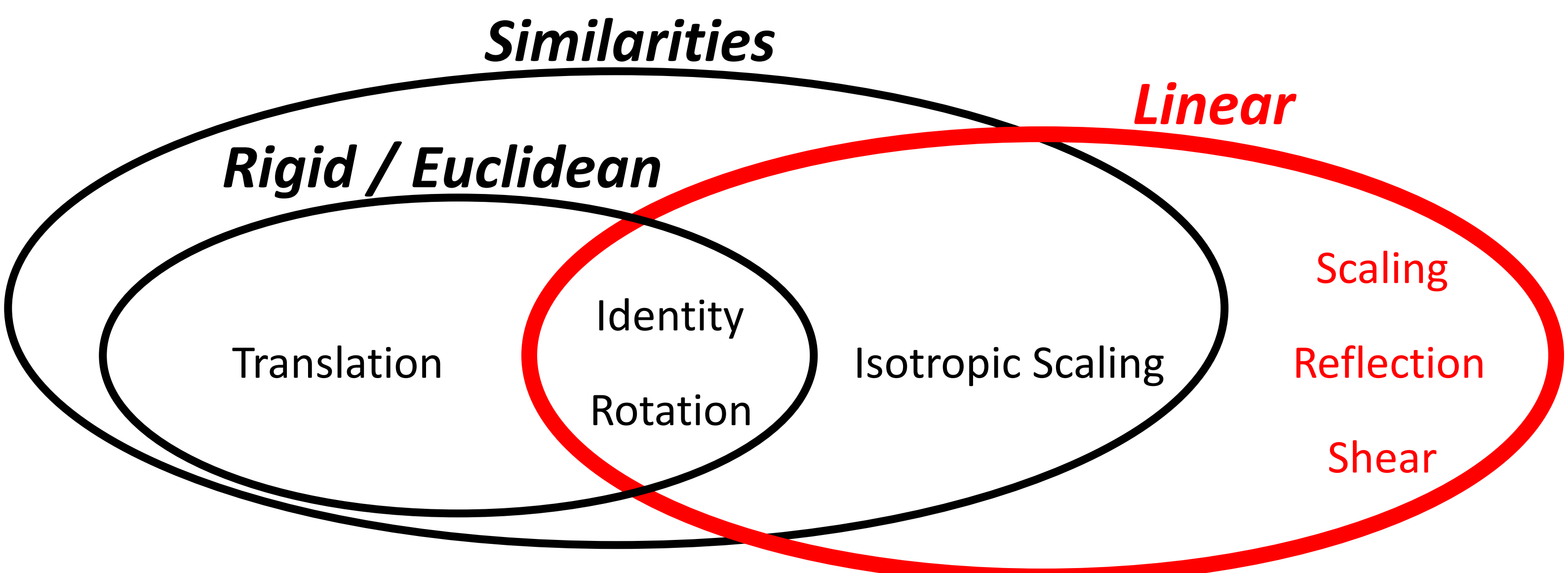
- Similarity transformation:
  $x \rightarrow cAx + t$ with $c \in \mathbb{R}$ and A orthogonal

Affine Transformations

- General Linear Transformations



Scaling Reflection Shear



Linear Transformations

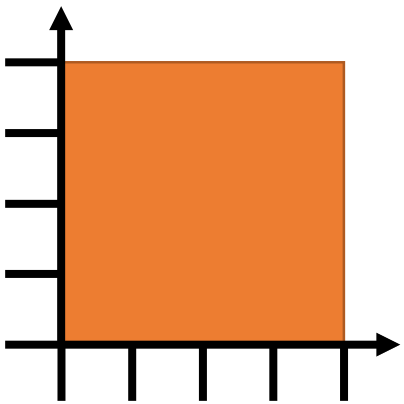
- Linear transformation:
  $x \rightarrow Ax$ with arbitrary $A \in \mathbb{R}^{2\times 2}$
- **Matrix-vector** multiplications
  - Scaling
  - Shear
  - Rotation (around origin)

Scaling

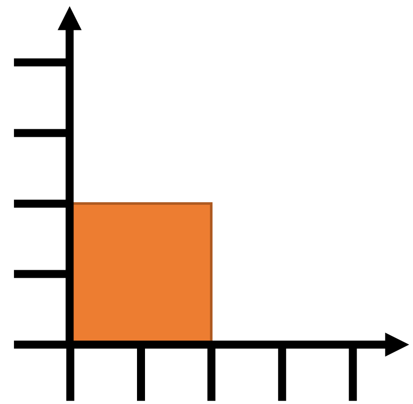
- Scaling
  - $scale(s_x, s_y) = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix}$
  - $scale(s_x, s_y)\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} s_x x \\ s_y y \end{pmatrix}$
  - Changes length and possibly direction

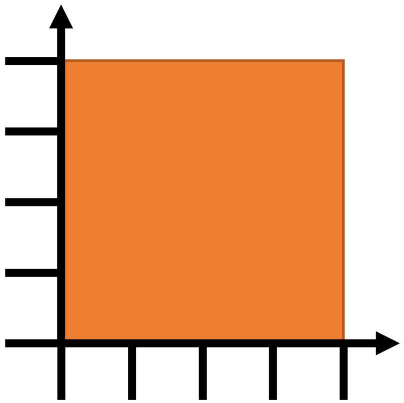
Scaling

- Examples

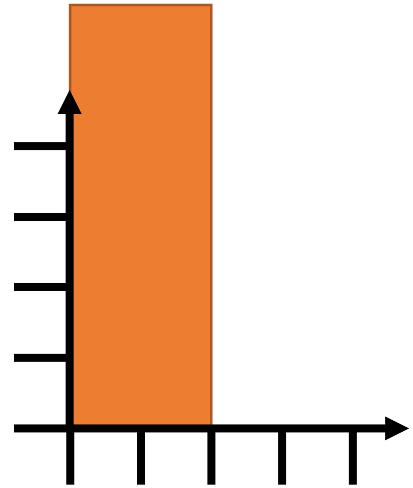


$$\begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$$





$$\begin{pmatrix} 0.5 & 0 \\ 0 & 1.5 \end{pmatrix}$$



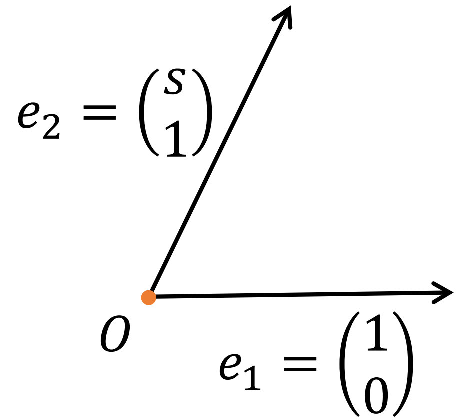
Scaling

- In 3D:

$$scale(s_x, s_y, s_z) = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{pmatrix}$$

Shearing

- Shearing
  - Pushing things sideways *(compare deck of cards)*

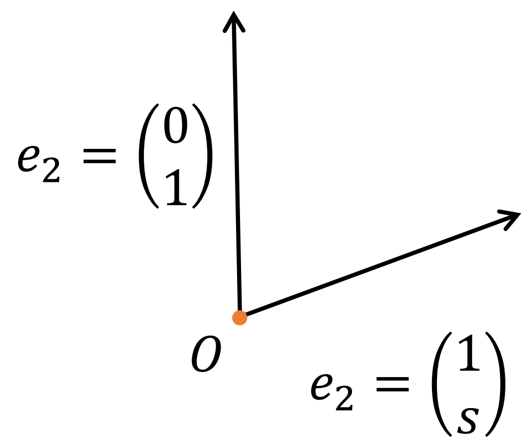


  - Horizontal (y-coordinate will not change)

$$shear_x(s) = \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix}$$

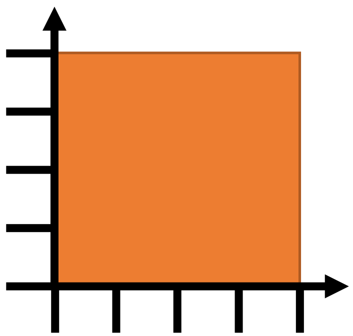
  - Vertical (x-coordinate will not change)

$$shear_y(s) = \begin{pmatrix} 1 & 0 \\ s & 1 \end{pmatrix}$$

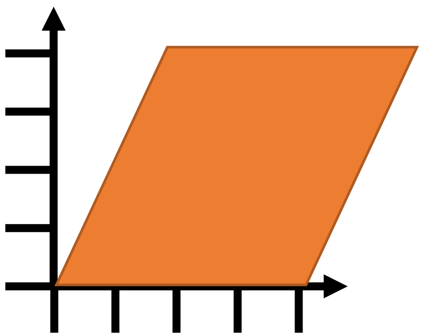


Shearing

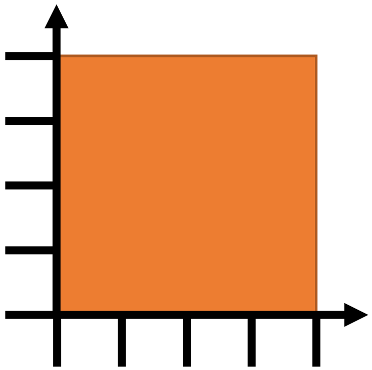
- Examples
  - Horizontal shear: vertical lines → 45°to the right



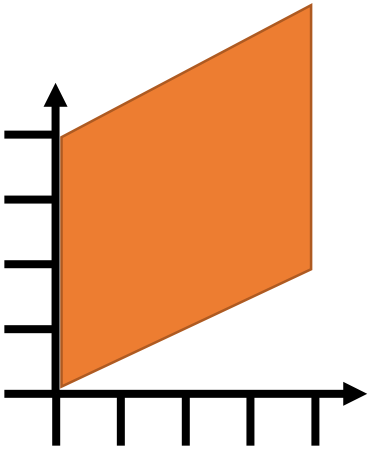
$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$



  - Vertical shear: horizontal lines → 45°to the top



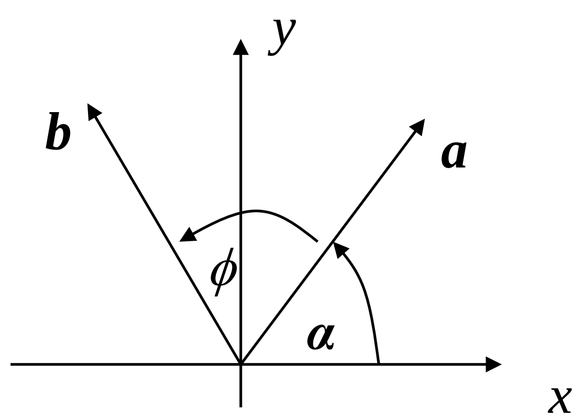
$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$



Simple Rotation in 2D

- Rotation
  - Vector $\boldsymbol{a} = (a_x, a_y)$, angle α with x-axis
  - Length $r = \sqrt{a_x^2 + a_y^2}$
  - By definition: $a_x = r\cos\alpha,$ $a_y = r\sin\alpha$
  - Rotation by an angle ϕ counter-clockwise:

$$b_x = r\cos(\alpha + \phi) = r\cos\alpha\cos\phi - r\sin\alpha\sin\phi$$
$$b_y = r\sin(\alpha + \phi) = r\sin\alpha\cos\phi + r\cos\alpha\sin\phi$$

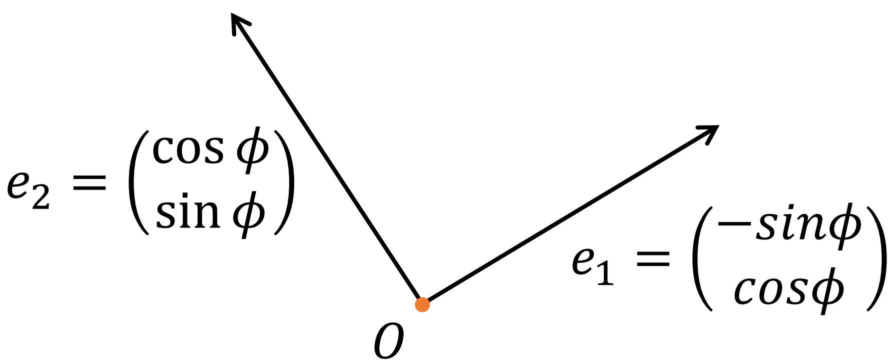


Simple Rotation in 2D

- After substitution
  - $b_x = a_x \cos\phi \; - \; a_y \sin\phi$
  - $b_y = a_y \cos\phi \; + \; a_x \sin\phi$

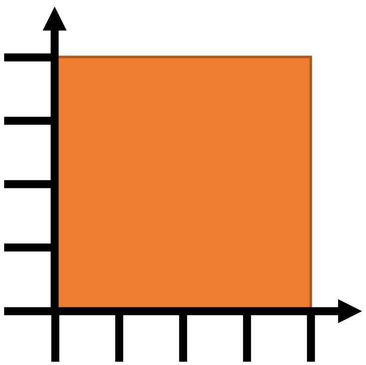
- Matrix form taking a to b

$$rotate(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}$$

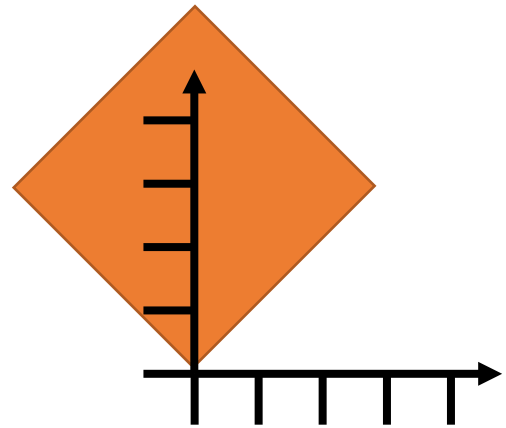


Simple Rotation in 2D

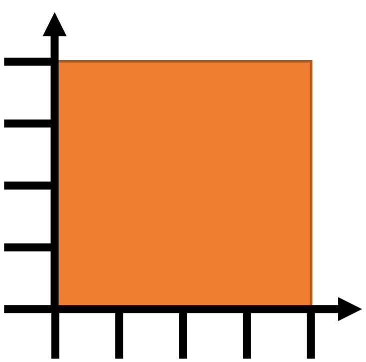
- Rotation by 45°counter-clockwise



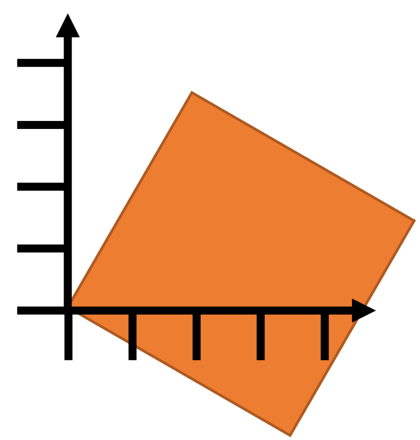
$$\begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix}$$



- Rotation by 30° clockwise

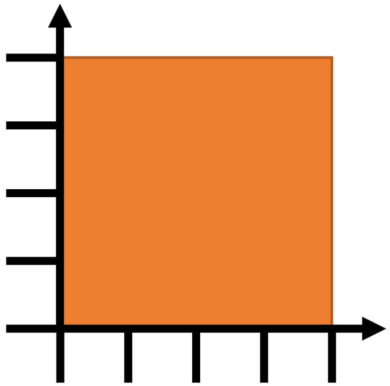


$$\begin{pmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{pmatrix}$$

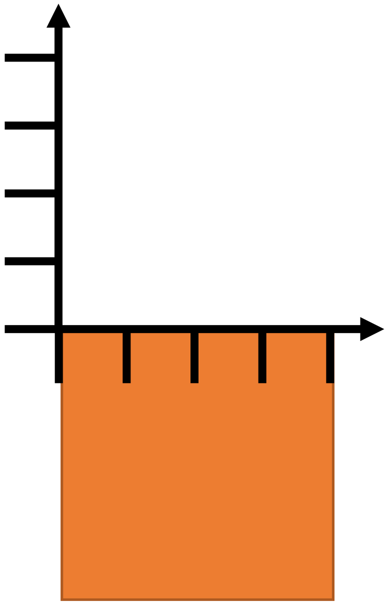


Reflection

- Reflection
  - Reflect a vector across either of the coordinate axes
  - About x-axis (multiply y by -1)

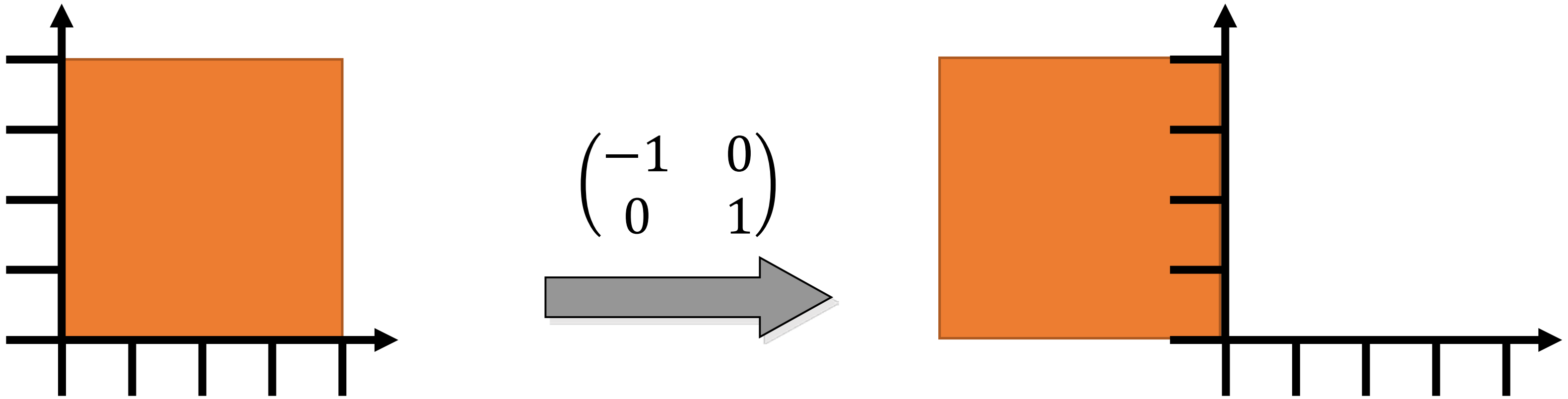


$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$



Reflection

- Across y-axis (multiply x coordinates by -1)



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

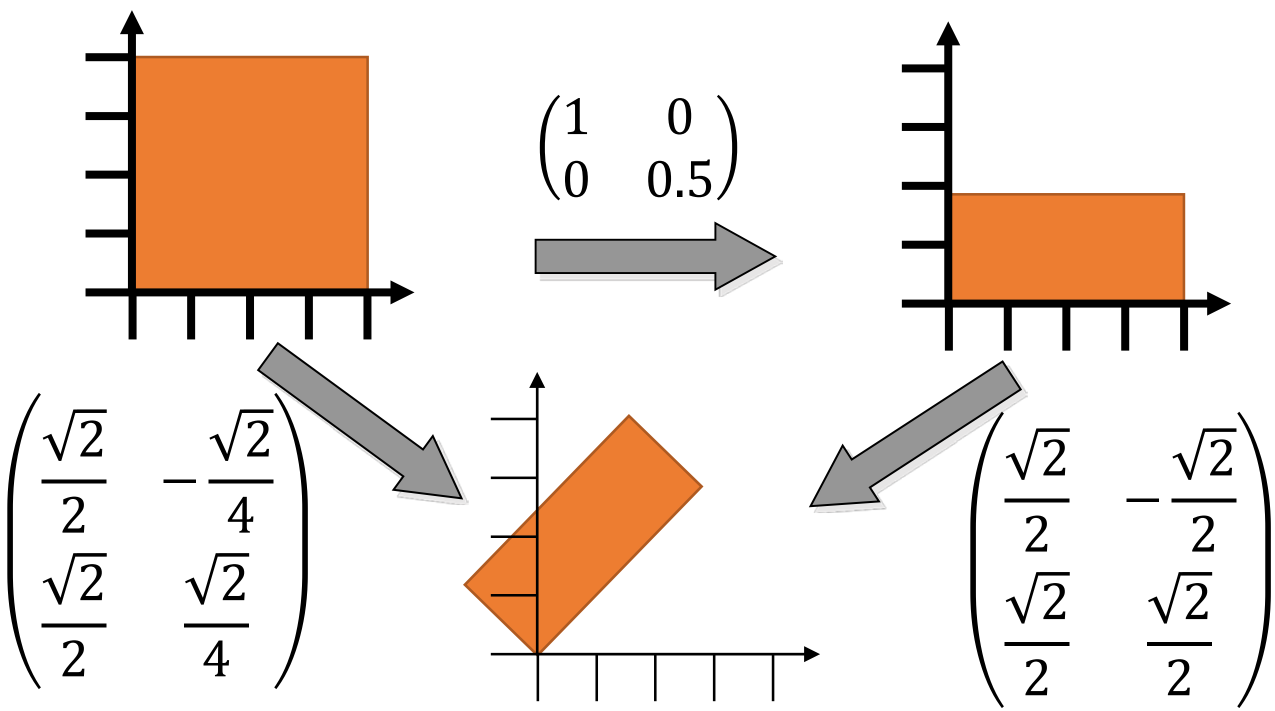
Reflection

- Remark:
  - Determinant of a reflection is negative
  - In 3D: reflection with respect to a plane, e.g. the plane $x = 0$:

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

Linear Transformations

- Compositing of 2D transformations
  - First $v_2 = Sv_1$ then $v_3 = Rv_2$
  - Equivalently $v_3 = R(Sv_1) = (RS)v_1$



Linear Transformations

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

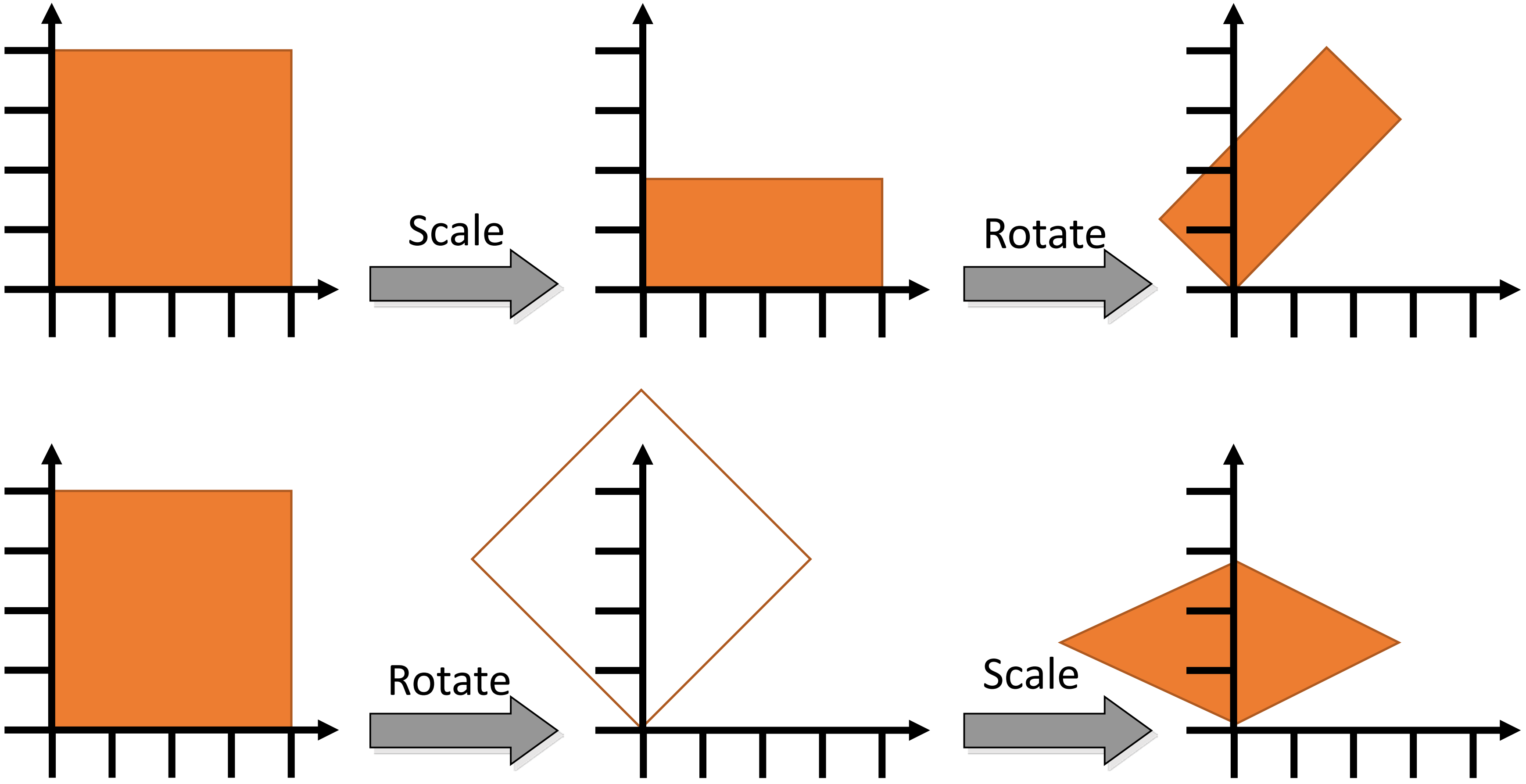
- Matrix multiplications are associative:

$$(RS)T = R(ST) \rightarrow v_3 = (RS)v_1 = Mv_1 \text{ with } M = RS$$

- Matrix multiplications are **not** commutative
  - **The order** of transformations **does matter** !!!
  - Note the difference
    - Scaling then rotating
    - Rotating then scaling

Linear Transformations

- Note that the order of transformations is important



Linear Transformations

- Decomposition of transformations
  - Any nonsingular matrix transformation M can be written as the product of various matrices, e.g. $M = M_1 M_2$

- In 2D: Decomposition of any linear 2D transform into product: rotation → scale → rotation = $R_2 S R_1$
  - From existence of singular value decomposition (SVD) (Singulärwertzerlegung, Ausgleichsprobleme)
  - Note that the scale can have negative entries

Linear Transformations

- Example: shearing
  - σ_i singular values, R_1 and R_2 rotations

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = R_2 \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} R_1 = \begin{pmatrix} 0.851 & -0.526 \\ 0.526 & 0.851 \end{pmatrix} \begin{pmatrix} 1.618 & 0 \\ 0 & 0.618 \end{pmatrix} \begin{pmatrix} 0.526 & 0.851 \\ -0.851 & 0.526 \end{pmatrix}$$

R$_1$ SR$_1$ R$_2$SR$_1$

Linear Transformations

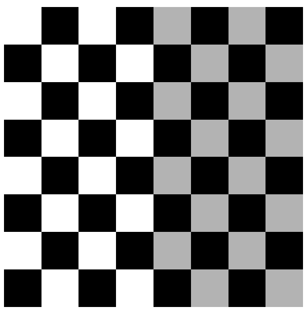
- Matrix decomposition: represent rotations with shears

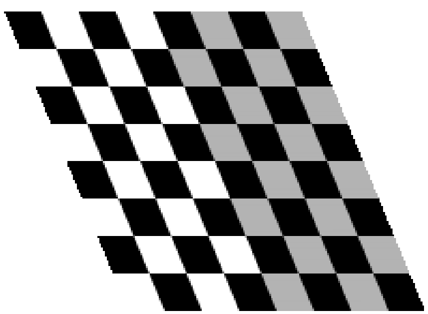
$$\begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} = \begin{pmatrix} 1 & \dfrac{\cos\phi - 1}{\sin\phi} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \sin\phi & 1 \end{pmatrix} \begin{pmatrix} 1 & \dfrac{\cos\phi - 1}{\sin\phi} \\ 0 & 1 \end{pmatrix}$$

- Useful for raster rotation
  - Very efficient raster operation for images: only column-wise and row-wise operations!
  - Introduces some jaggies but no holes

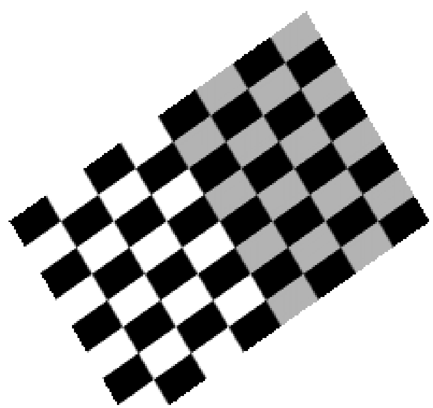
Linear Transformations

- $rotate\left(\frac{\pi}{4}\right) = S_3 S_2 S_1 = \begin{pmatrix} 1 & 1-\sqrt{2} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \frac{\sqrt{2}}{2} & 1 \end{pmatrix} \begin{pmatrix} 1 & 1-\sqrt{2} \\ 0 & 1 \end{pmatrix}$

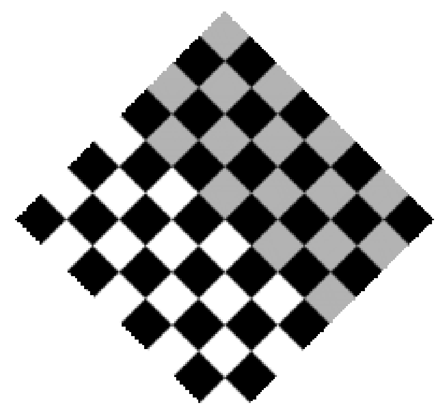




S_1



$S_2 S_1$



$S_3 S_2 S_1$

Linear Transformations

- Remark: images – simple raster rotation
  - Take raster position (i, j) and apply horizontal shear
$$\begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix} = \begin{pmatrix} i + sj \\ j \end{pmatrix}$$
  - Round sj to nearest integer: in every row a constant shift
  - Move each row sideways by a different amount
  - Resulting image has no gaps

Affine Transformations

- Back to affine transformations:
  Another aspect to look is the difference of **Points** and **Vectors**

Points

- Position in space
- Elements of Euclidean space E
- No algebraic structure
  - No addition
  - no multiplication with scalars, etc.

Vectors

- Difference between two points
- Elements of a **vector space** $\mathcal{V}$
- Algebraic structure
  - Addition of vectors
  - Multiplication with scalars
  - Linear combination of vectors
  - etc.

Affine Transformations

- Affine space: points and vectors
  - Given two points a, b ∈ E there exists a unique vector in V denoted by **ab** from a to b
  - Algebraic structure:
    - **ab** = b – a difference of two points is vector
    - b = a + **ab** point + vector = point
    - **ab +bc = ac** vector + vector = vector

- Often, Euclidean space and vector space are mixed
- Point is considered as vector from origin to point

Affine Transformations

- Special linear combinations of points p_i
  - Barycenter, affine combination
    $p = \sum_i \lambda_i p_i$ is a point if $\sum_i \lambda_i = 1$
    - Example.: 0.5 a + 0.5 b = c = midpoint of a and b

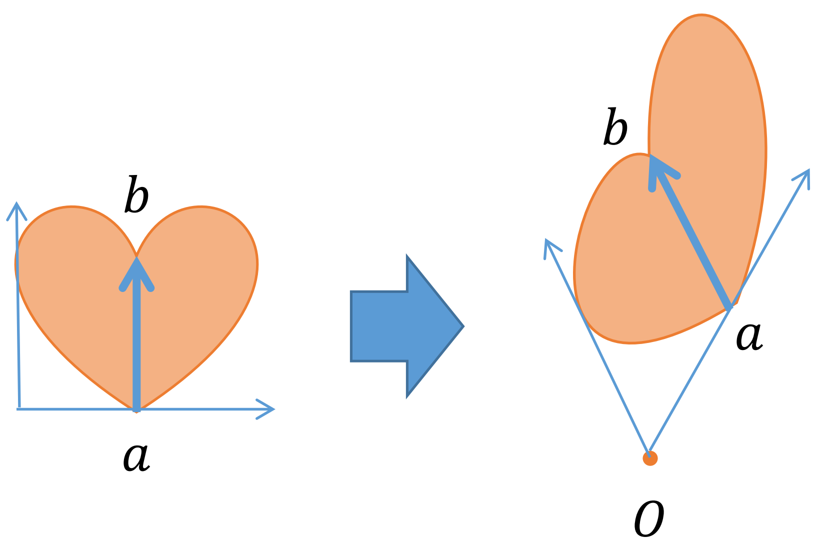
  - Generalized Difference:
    $v = \sum_i \lambda_i p_i$ is a vector if $\sum_i \lambda_i = 0$
    - Example.: -a + b = b-a = vector from a to b

Affine Transformations

- Remember: affine transformation of points

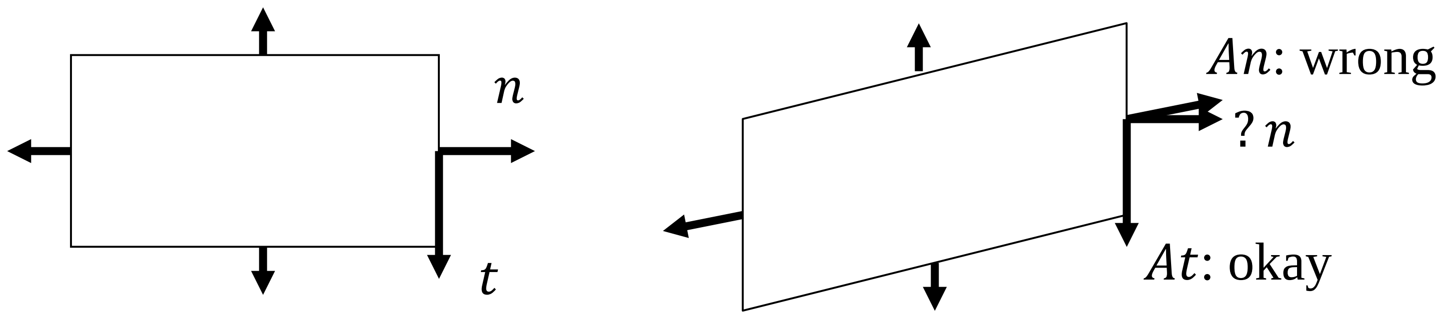
$$x \rightarrow Ax + t$$

- but vectors?
  - $v = b - a$
  - $v \rightarrow Ab + t - (Aa + t) = A(b - a) = Av$
  - becomes linear transformation with A, without translation t



Affine Normal Transformations

- More tricky: transforming normal vectors
  - Normals are perpendicular to tangent plane of a surface
  - Transformation with matrix A will differ from transformation of underlying surface



Affine Normal Transformations

- Matrix N for transformation of normal n
  - Perpendicular to tangent vector $n^T \cdot t = 0$
  - Transformed normal n' should also be perpendicular to transformed tangent $t' = At$
  - $n^T \cdot t = n^T \cdot I \cdot t$

$$= n^T \cdot A^{-1} A \cdot t$$

$$= \left(n^T A^{-1}\right) \cdot \mathrm{t}' \overset{!}{=} 0$$

  - $\rightarrow\ n' = \left(n^T A^{-1}\right)^T = (A^{-1})^T n = A^{-T} n$
- Result: transform normal vectors by matrix $\boldsymbol{A^{-T}}$
- Remark: if M is orthogonal (rigid transformation) then $A^{-T} = A$
- Remark: length of normal vector can change!

Homogenous coordinates

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

- Homogenous coordinates
  - Add 1 as third homogeneous coordinate

$$(x, y) \rightarrow (x, y, 1)$$

  - Then apply a 3x3 matrix
  - Finally, get rid of homogeneous coordinate
    → Dehomogenization:

$$(x, y, w) \rightarrow \left(\frac{x}{w}, \frac{y}{w}\right)$$

  - Identify (x, y) with the line $\{(\alpha x, \alpha y, \alpha) \mid \alpha \in \mathbb{R}\}$ in 3D or with any non-zero point on this line – e.g. $(x, y, 1)$
  - $(x, y, 1)$, $(3x, 3y, 3)$, $(0.5x, 0.5y, 0.5)$ represent the same 2D-point (x, y)

Homogenous coordinates

- Transformations on homogeneous coordinates:

$$\begin{pmatrix} x \\ y \end{pmatrix} \to \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \xrightarrow{\cdot A} \begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} \to \begin{pmatrix} \frac{x'}{w'} \\ \frac{y'}{w'} \end{pmatrix}$$

- For a 3x3 matrix $A = (a_{ij})$:

$$\begin{pmatrix} x \\ y \end{pmatrix} \to \frac{1}{a_{31}x + a_{32}y + a_{33}} \begin{pmatrix} a_{11}x + a_{12}y + a_{13} \\ a_{21}x + a_{22}y + a_{23} \end{pmatrix}$$

Homogenous coordinates

- If the last row of A is $(0{,}0{,}1)$ the mapping is affine

- Structure of a general affine transformation in homogeneous coordinates

linear part		translation
0	0	1

base vectors after transf.

$\begin{pmatrix} e_{1,x} \\ e_{1,y} \end{pmatrix}, \begin{pmatrix} e_{2,x} \\ e_{2,y} \end{pmatrix}$		$\begin{pmatrix} t_x \\ t_y \end{pmatrix}$
0	0	1

Homogenous coordinates

- Transformation Rules & Matrix Operations

Multiplication $\equiv$ composition

$$x \xrightarrow{T} Tx = y \xrightarrow{S} Sy = z \quad \equiv \quad x \xrightarrow{ST} STx = z$$

Inverse matrix $\equiv$ Inverse transformation

- Note order of multiplication: ST means: first T, then S

Let’s play

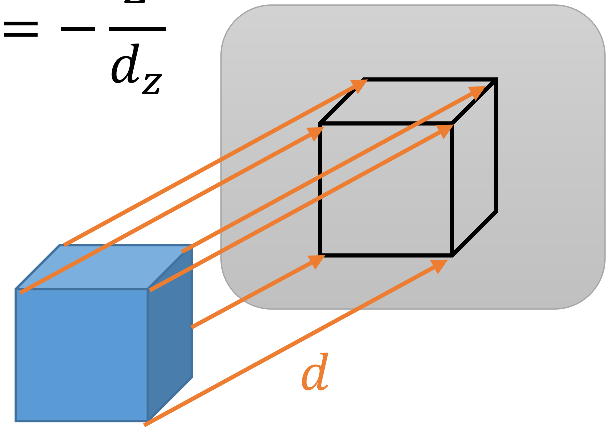
- Affine Transformations with the HTML Canvas

3D Transformations

- Ultimately, we want to do 3D graphics
  → objects have 3D coordinates (x, y, z)
- But our screen is only 2D…
- So we have to map the 3D world to the 2D screen

- First attempt:
  project along a given direction $d = (d_1, d_2, d_3)$ to a screen plane, e.g. $z = 0$:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} + \alpha d = \begin{pmatrix} x' \\ y' \\ 0 \end{pmatrix} \rightarrow \alpha = -\frac{z}{d_z}$$

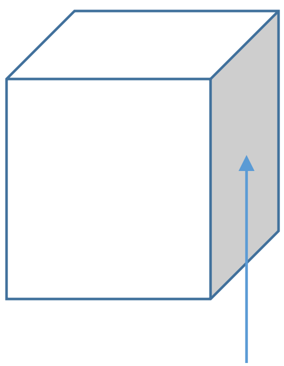


- $\rightarrow \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\frac{d_x}{d_z} \\ 0 & 1 & -\frac{d_y}{d_z} \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$

3D Perspective ?

- Look at the image of a cube

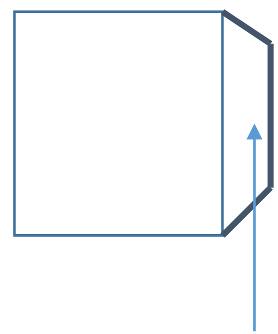
parallel projection:
affine mapping



shearing

→ possible with current tool set

real perspective projection:
not affine



not affine

→ next lecture: projective transform.

3D Transformations

- Basic 3D transformation: as in 2D
  - Translation: $translate(d_x, d_y, d_z) = \begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$
  - Scaling: $scale(s_x, s_y, s_z) = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
  - Z-Shear: $shear_z(d_x, d_y) = \begin{pmatrix} 1 & 0 & d_x & 0 \\ 0 & 1 & d_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

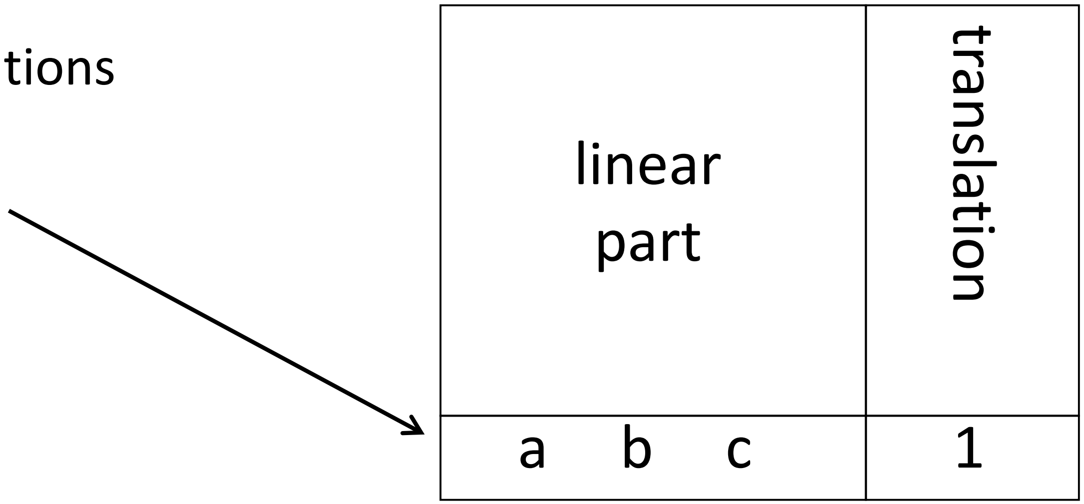
Homogenous coordinates

- Rotation around the x-, y- and z-axis
  - $Rot_x(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
  - $Rot_y(\phi) = \begin{pmatrix} \cos\phi & 0 & \sin\phi\ 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
  - $Rot_z(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

Homogeneous Coordinates

- Homogeneous coordinates helpful to represent affine transformations as matrix multiplications
- But what about the bottom row?

  → projective transformations



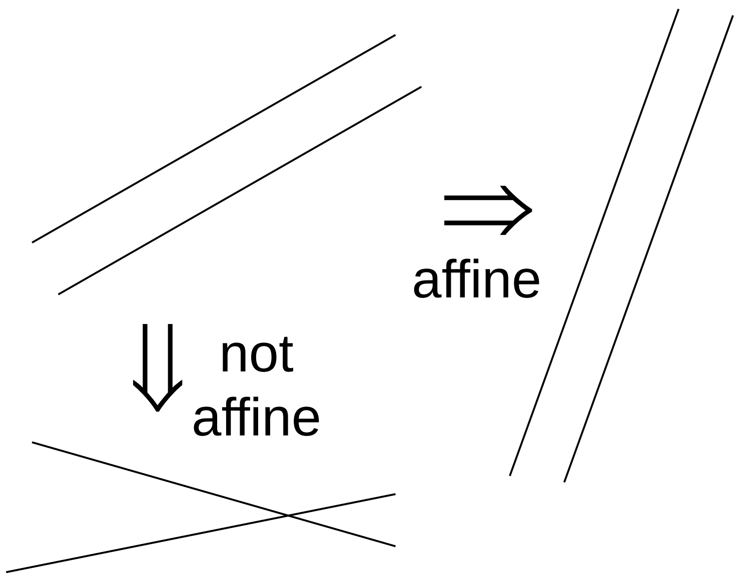
Affine Transformations

- Affine = combination of linear transformation and translation

$$x \rightarrow Ax + b$$

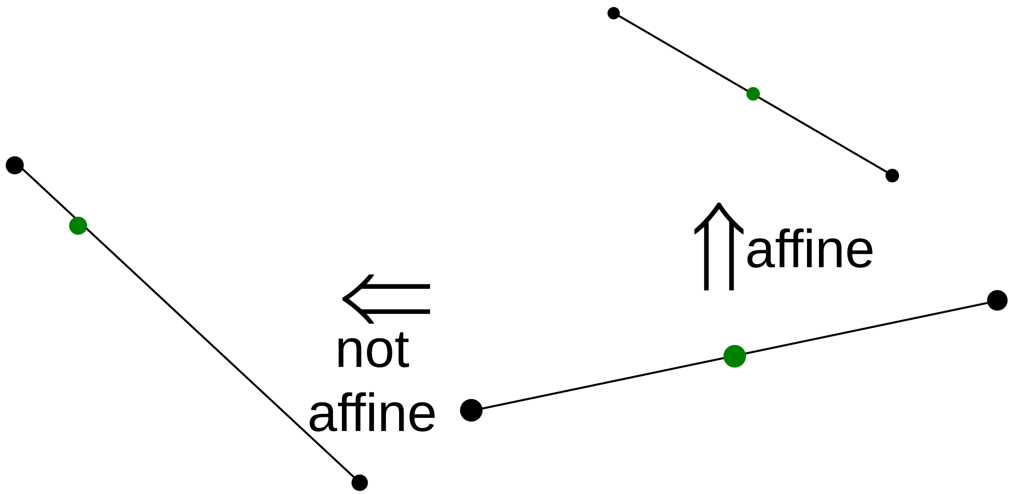
- Abstract characterization
  - Maps lines to lines
  - Parallel lines will be mapped to parallel lines
  - Division ratios are preserved
  - Angles are not preserved
  - Examples: rotations, translation, scaling, shears

Affine Transformations



parallel lines remain parallel

relative position of points
on lines remain:
ratios are maintained



Perspective ?

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

- Perspective is obviously more than affine
  - But it also maps lines to lines
  - It does not maintain parallelity
  - It does not maintain ratios
- New class of **projective** mappings

Side Remark
Felix Klein's "Erlanger Programm"

Aus Wikipedia „Erlanger Programm"

- Die elementare euklidische Geometrie oder Kongruenzgeometrie ist die Geometrie des Anschauungsraumes, deren Transformationsgruppe die Gruppe der Bewegungen (also der Translationen, Drehungen oder Spiegelungen) ist, die alle längen- und winkeltreue Abbildungen sind.
- Verzichtet man bei den zugelassenen Transformationen auf die Längentreue und lässt auch Punktstreckungen zu, so erhält man die äquiforme Gruppe der Transformationen, die die Ähnlichkeits- oder äquiforme Geometrie kennzeichnet.
- Verzichtet man auch auf die Winkeltreue, so gelangt man zur Transformationsgruppe der bei Koordinatendarstellung linearen Transformationen, d.h. der Kollineationen, die das Teilverhältnis je dreier Punkte erhalten. Sie kennzeichnen die affine Geometrie.
- Fügt man schließlich zum Anschauungsraum noch unendlich ferne oder uneigentliche Punkte als Schnittpunkte von Parallelen hinzu, so lassen die Kollineationen in diesem Raum das Doppelverhältnis von je vier Punkten invariant und bilden die Gruppe der projektiven Transformationen, deren zugehörige Geometrie die projektive Geometrie ist.

Side side remark

- Felix Klein is also the inventor of the **Klein Bottle**
- We will come back to this later in this lecture

From Futurama