清華大學 Tsinghua University
交叉信息研究院 Institute for Interdisciplinary Information Sciences

Lecture 3: 3D Transformations

Li Yi

2025.03.06

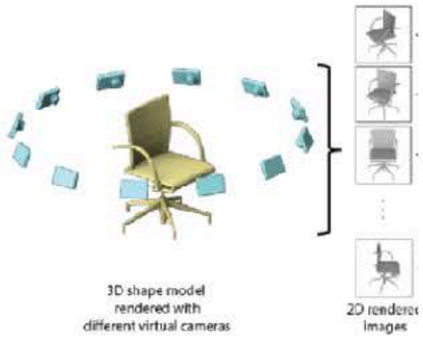
Announcement

- Homework 1 released!

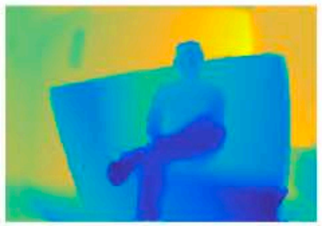
- Please provide your feedback through anonymous WeChat questionnaire (https://www.wjx.cn/vm/rVxWHKl.aspx)

Recap

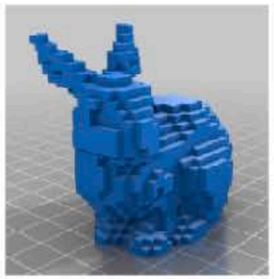
Rasterized form (regular grids)



Multi-view



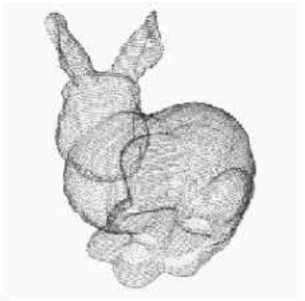
Depth Map



Volumetric

Geometric form (irregular)

Mesh



Point Cloud

$$F(x) = 0$$

Implicit Shape

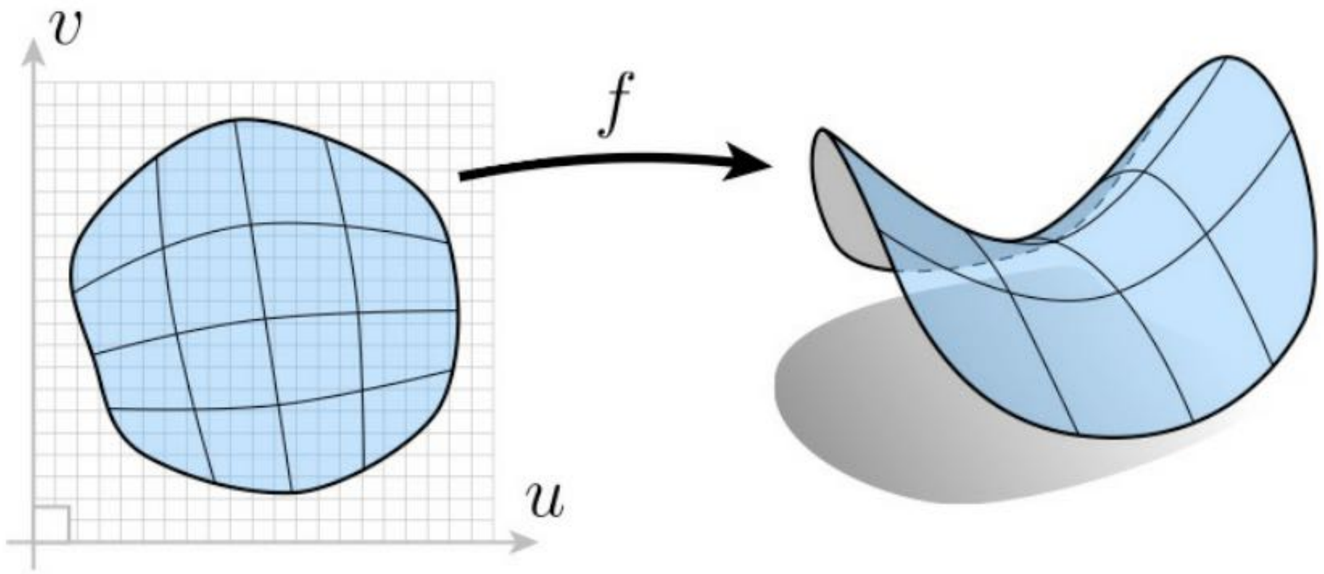
Outline

- Rasterized 3D representations
- Mesh
- Point cloud
- *Implicit representations*

Explicit Representation of Geometry

All points are given directly

Generally: $f : \mathbb{R}^2 \to \mathbb{R}^3; (u, v) \mapsto (x, y, z)$



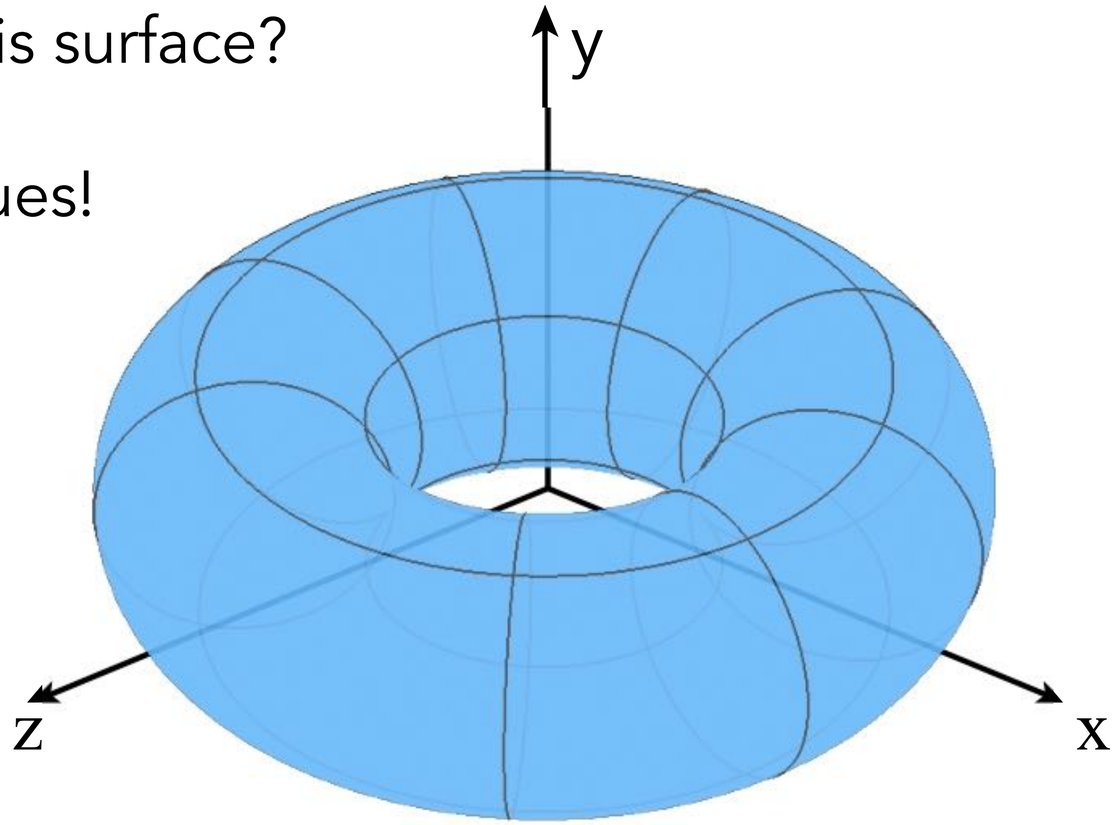
Slides credit: Ren Ng

Explicit Surface – Sampling Is Easy

$$f(u, v) = ((2 + \cos u)\cos v, (2 + \cos u)\sin v, \sin u)$$

What points lie on this surface?

Just plug in (u, v) values!

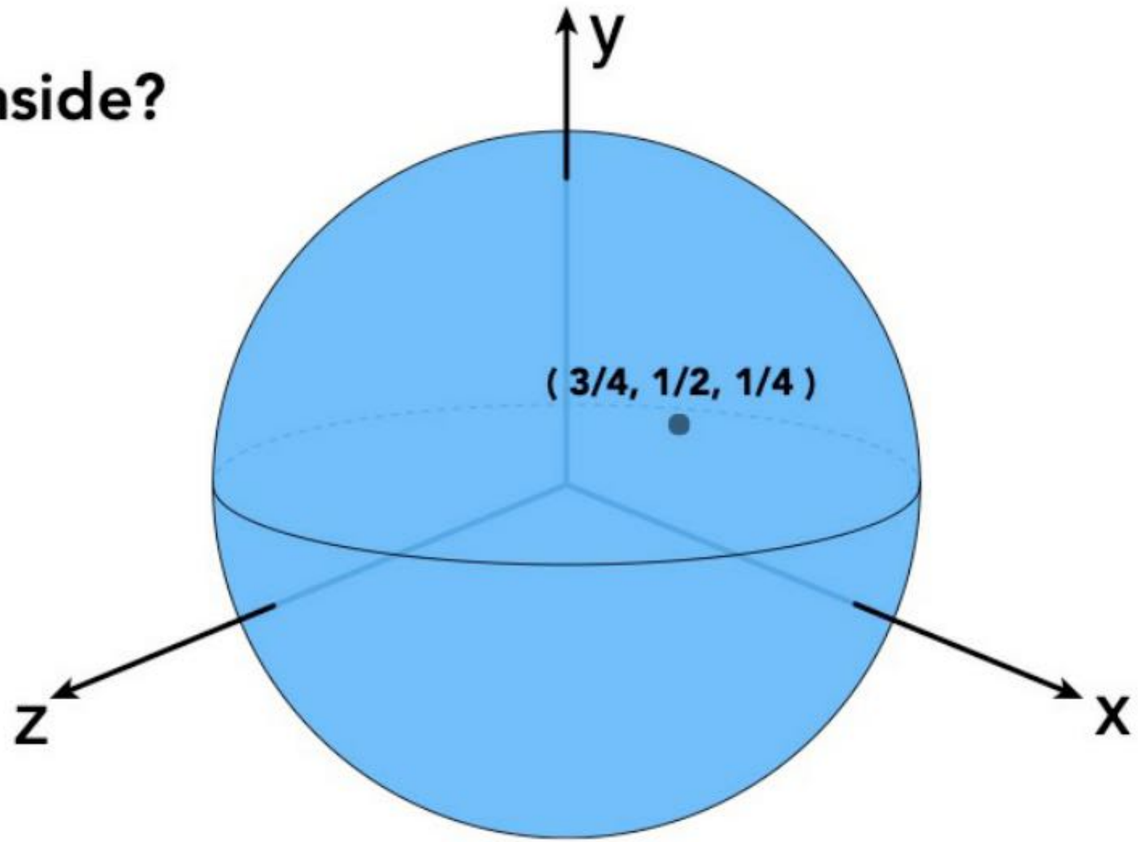


Explicit representations make some tasks easy

Explicit Surface - Inside/Outside Test Hard

$$f(u, v) = (\cos u \sin v, \sin u \sin v, \cos v)$$

Is (3/4, 1/2, 1/4) inside?



Some tasks are hard with explicit representations

Slides credit: Ren Ng

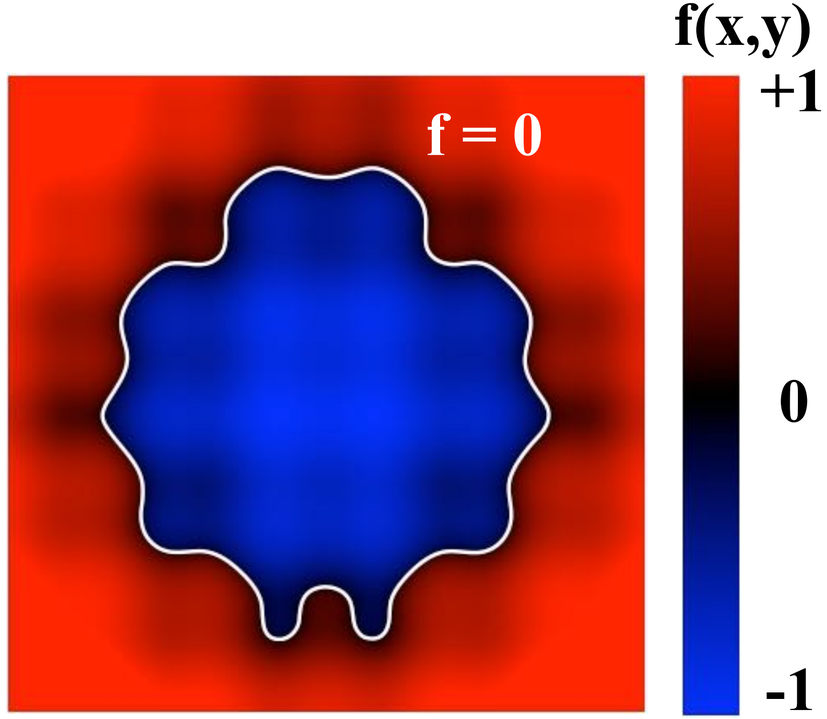
Implicit Representations of Geometry

Based on classifying points

- Points satisfy some specified relationship

E.g. sphere: all points in 3D, where $x^2 + y^2 + z^2 = 1$

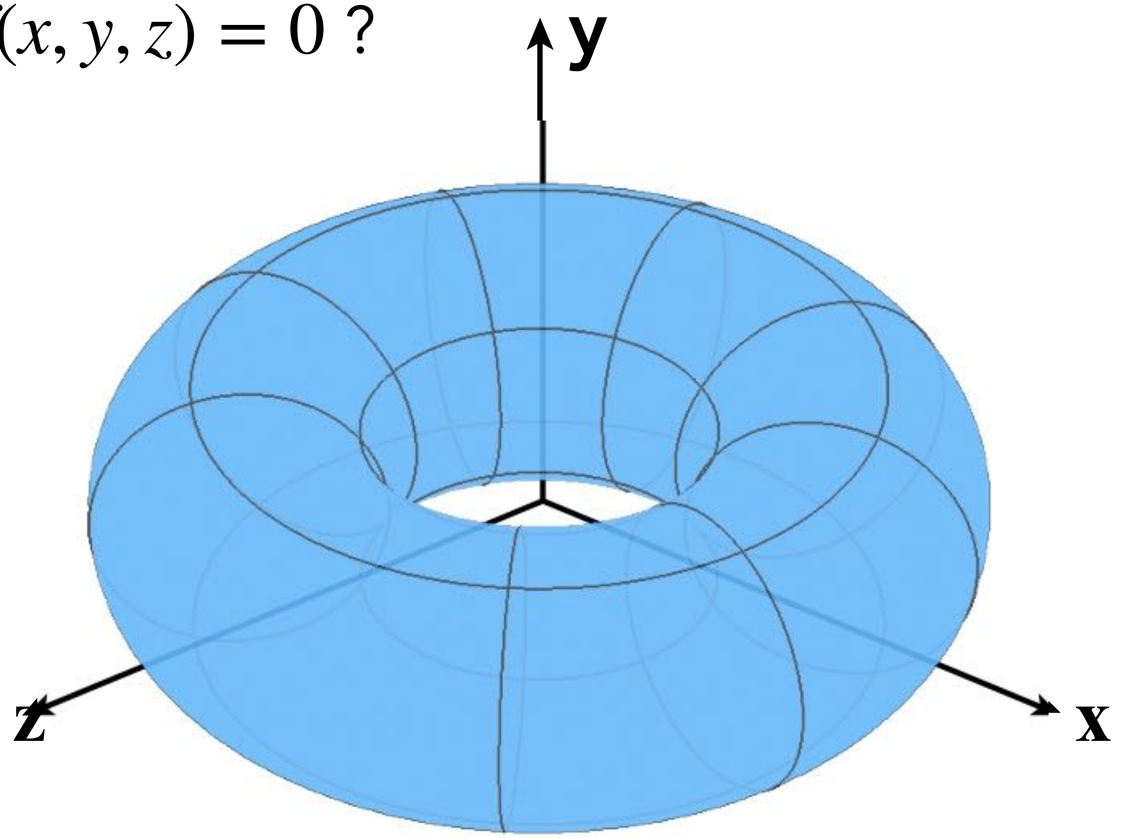
More generally, $f(x, y, z) = 0$



Implicit Surface – Sampling Can Be Hard

$$f(x, y, z) = (2 - \sqrt{x^2 + y^2})^2 + z^2 - 1$$

What points lie on $f(x, y, z) = 0$?



Some tasks are hard with implicit representations

Implicit Surface - Inside/Outside Tests Easy

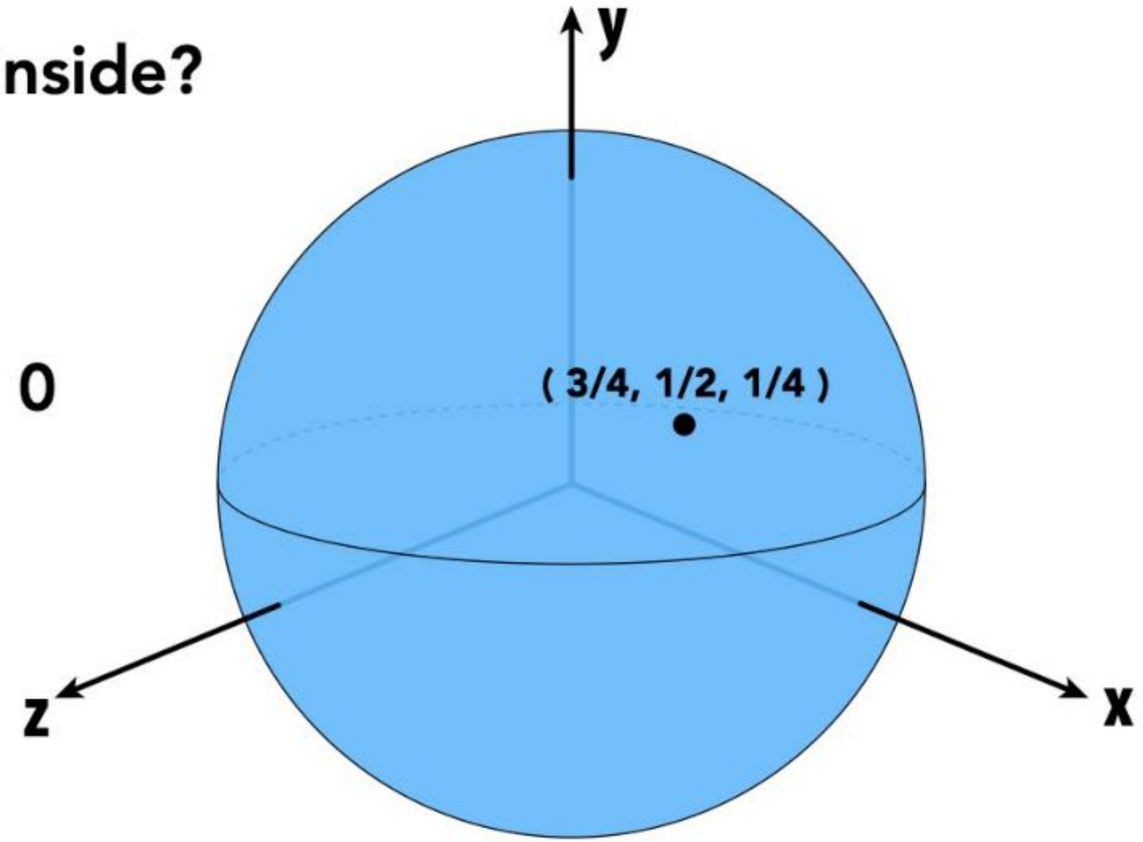
$$f(x, y, z) = x^2 + y^2 + z^2 - 1$$

Is (3/4, 1/2, 1/4) inside?

Just plug it in:

f(x,y,z) = –1/8 < 0

Yes, inside.



Implicit representations make some tasks easy

Slides credit: Ren Ng

Algebraic Surfaces (Implicit)

Surface is zero set of a polynomial in x, y, z

$$x^2+y^2+z^2=1$$

$$(R-\sqrt{x^2+y^2})^2+z^2=r^2$$

$$(x^2+\frac{9y^2}{4}+z^2-1)^3=$$

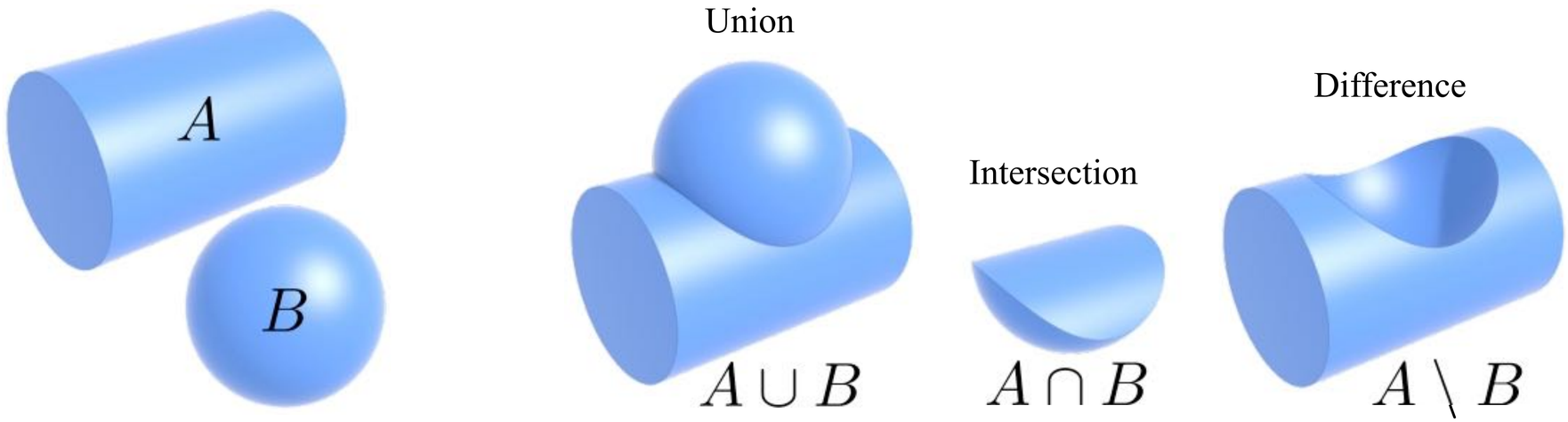
$$x^2z^3+\frac{9y^2z^3}{80}$$

More complex shapes?

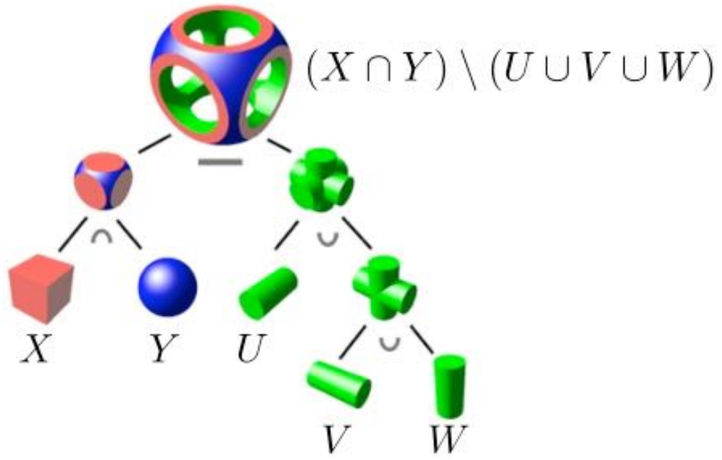
Slides credit: Jiajun Wu

Constructive Solid Geometry (Implicit)

Combine implicit geometry via Boolean operations



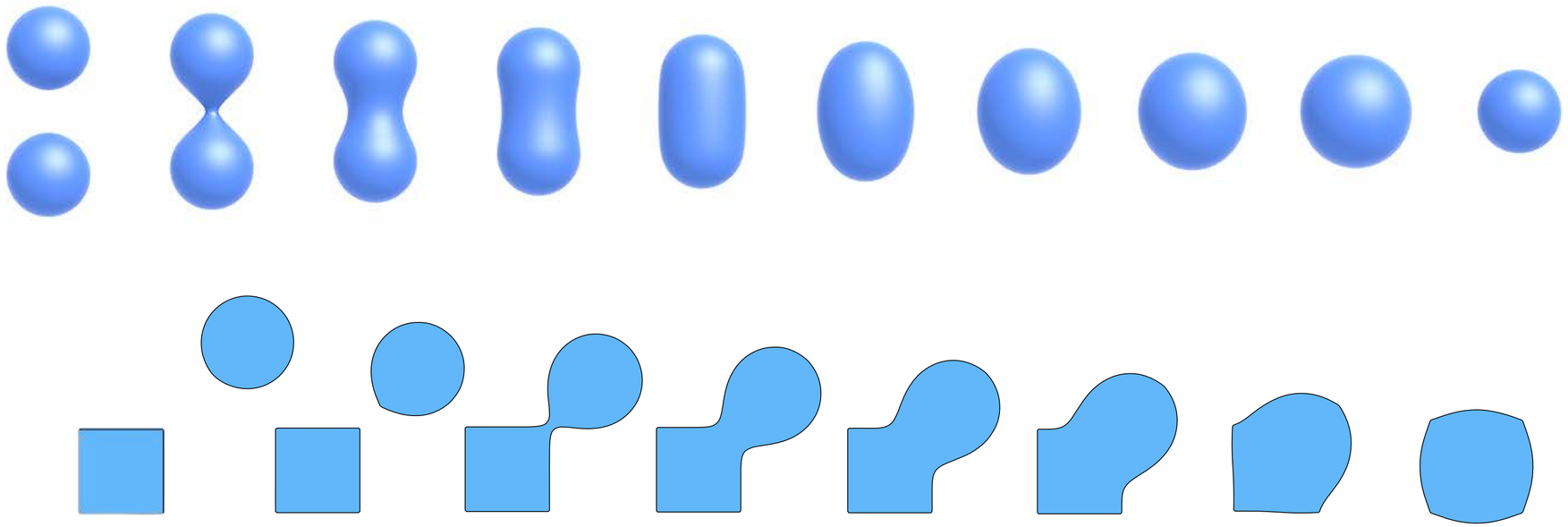
Boolean expressions :



Distance Functions

- Distance functions: giving minimum distance (could be signed distance) from anywhere to object

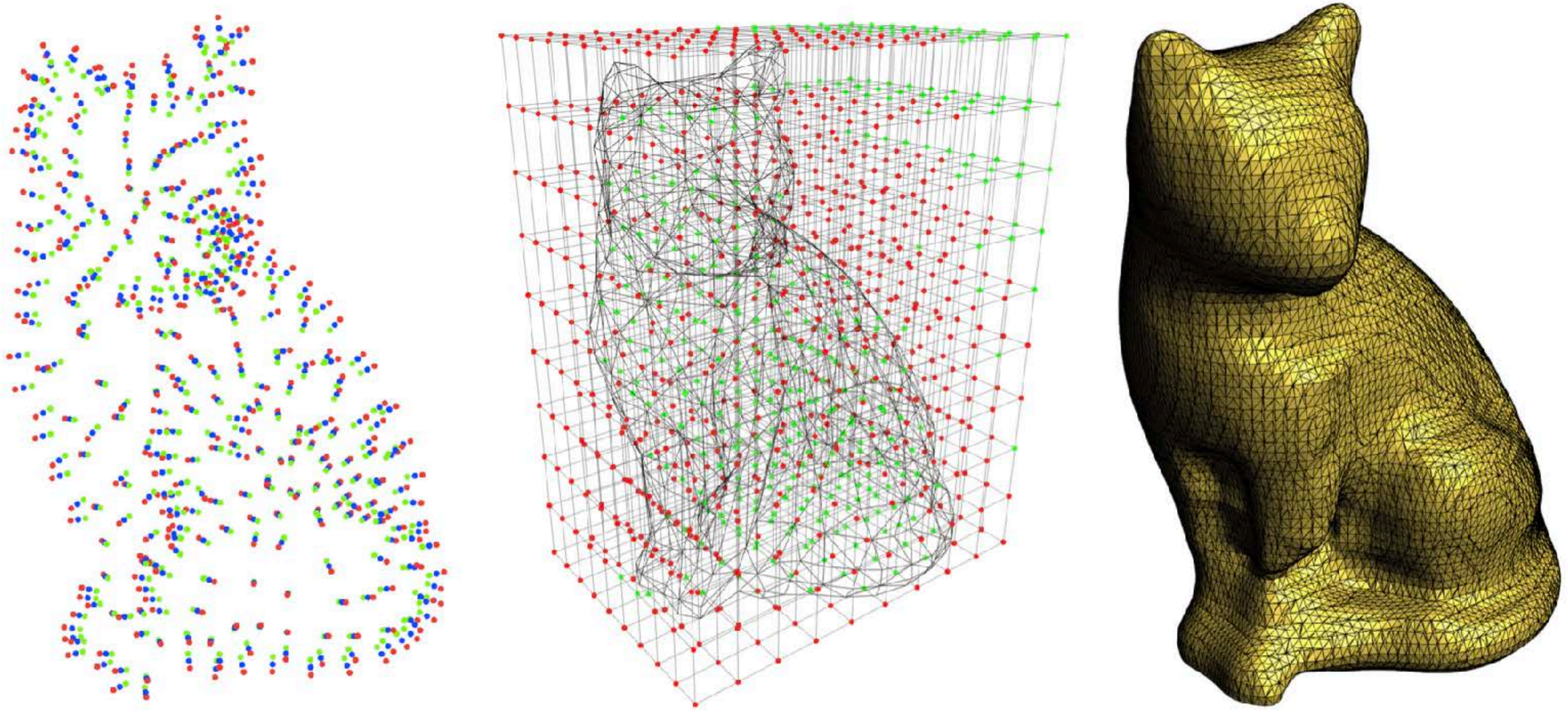
- Instead of booleans, gradually blend surfaces together using distance functions



Scene of Pure Distance Functions (Not Easy!)

Slides credit: Jiajun Wu

Conversion between different representations



Slides credit: Hao Su

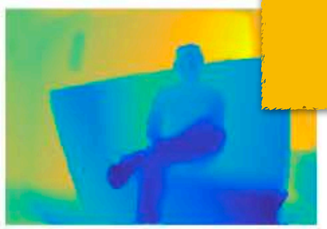
Take Home Message

Rasterized form (regular grids)

Geometric form (irregular)

No "best" geometric representation

Choose your representation depending on the task!



Depth Map

Volumetric

Point Cloud

$$f(x) = 0$$

Implicit Shape

3D Spatial Relationships

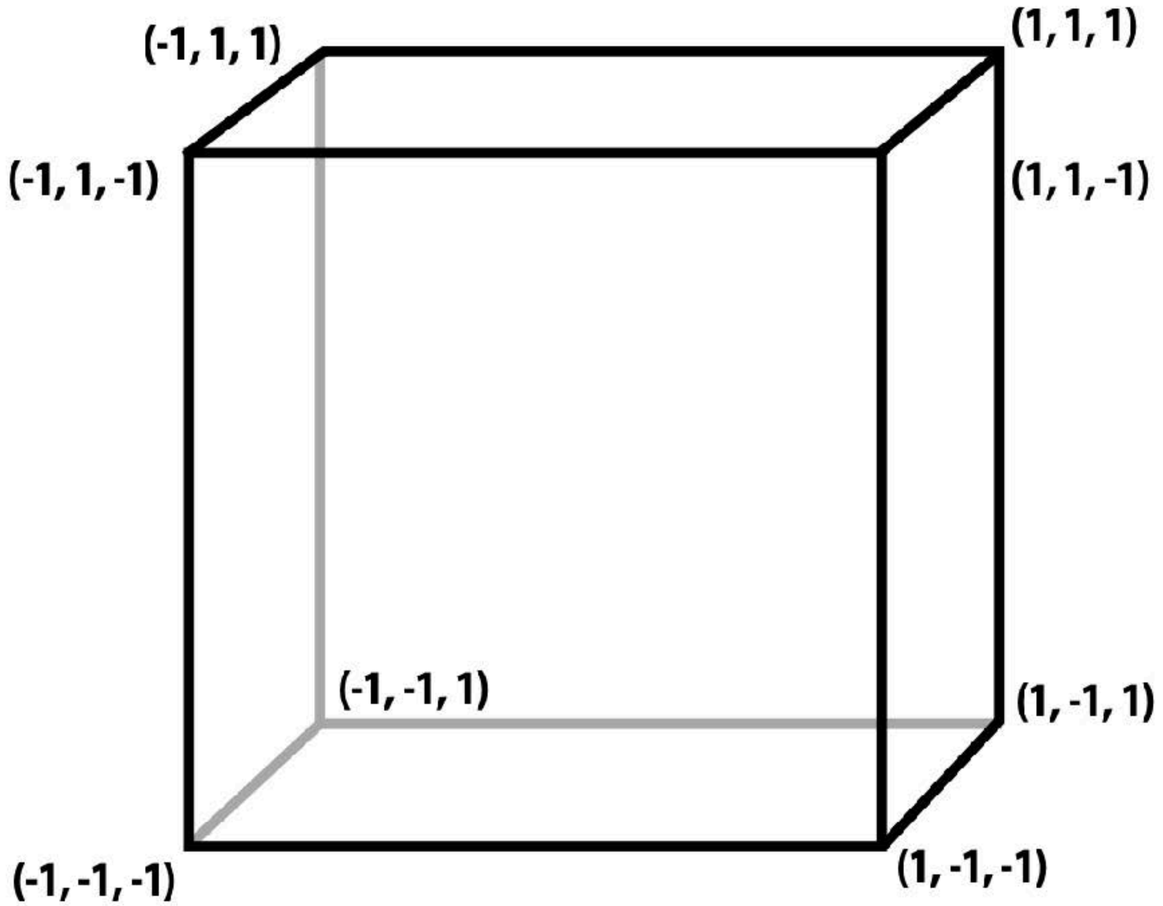
- How to represent the relationships between objects?

Slides credit: Hao Su

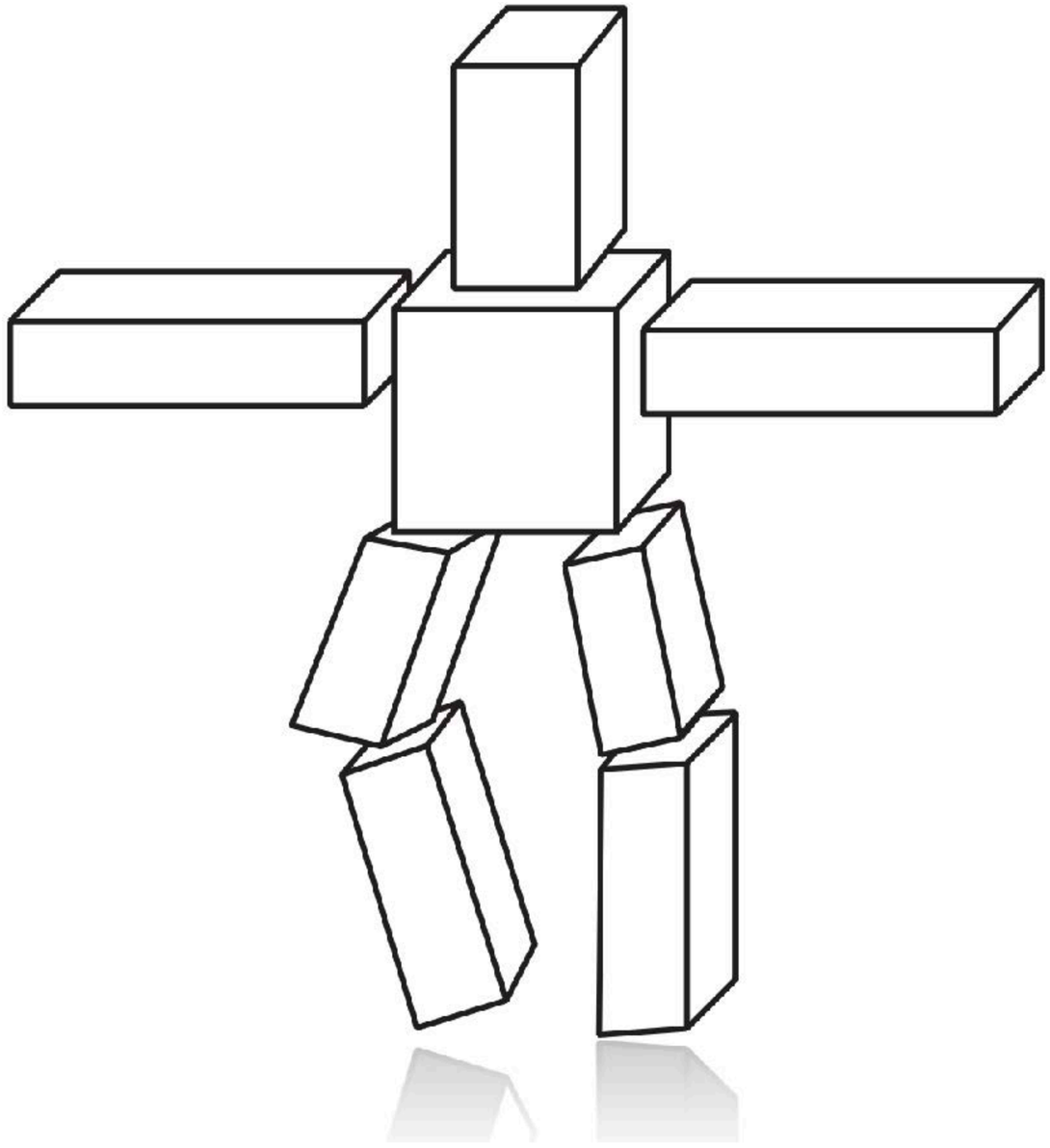
3D Spatial Relationships

- How to represent the relationships between objects?

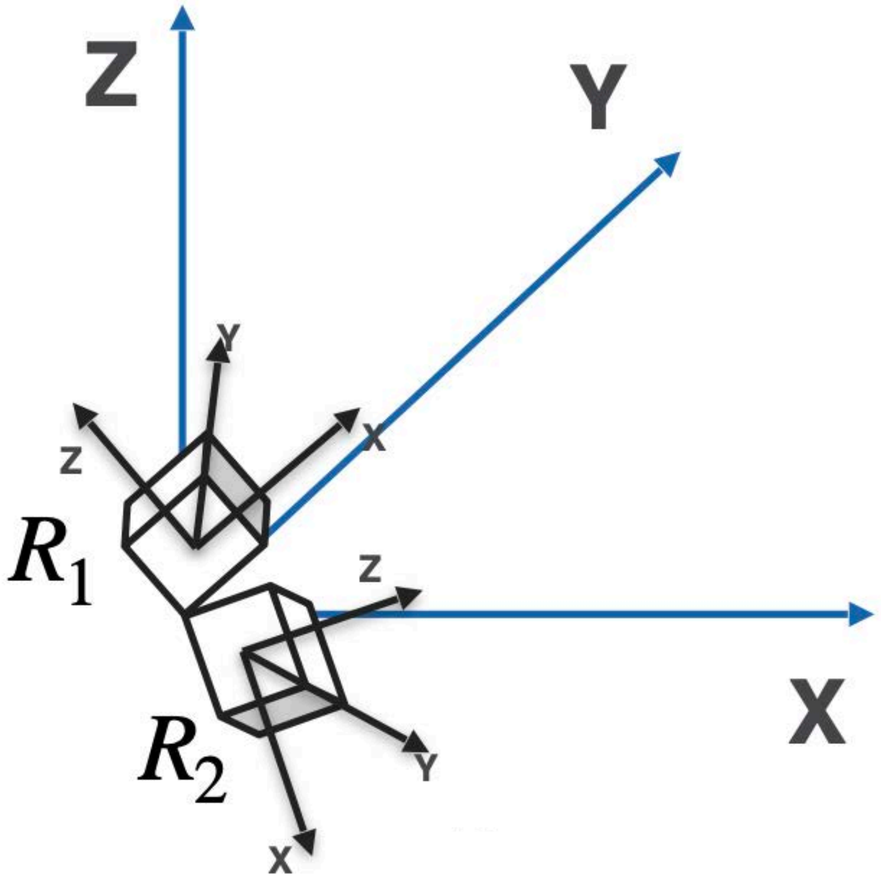
Cube



Drawing a Cube Person



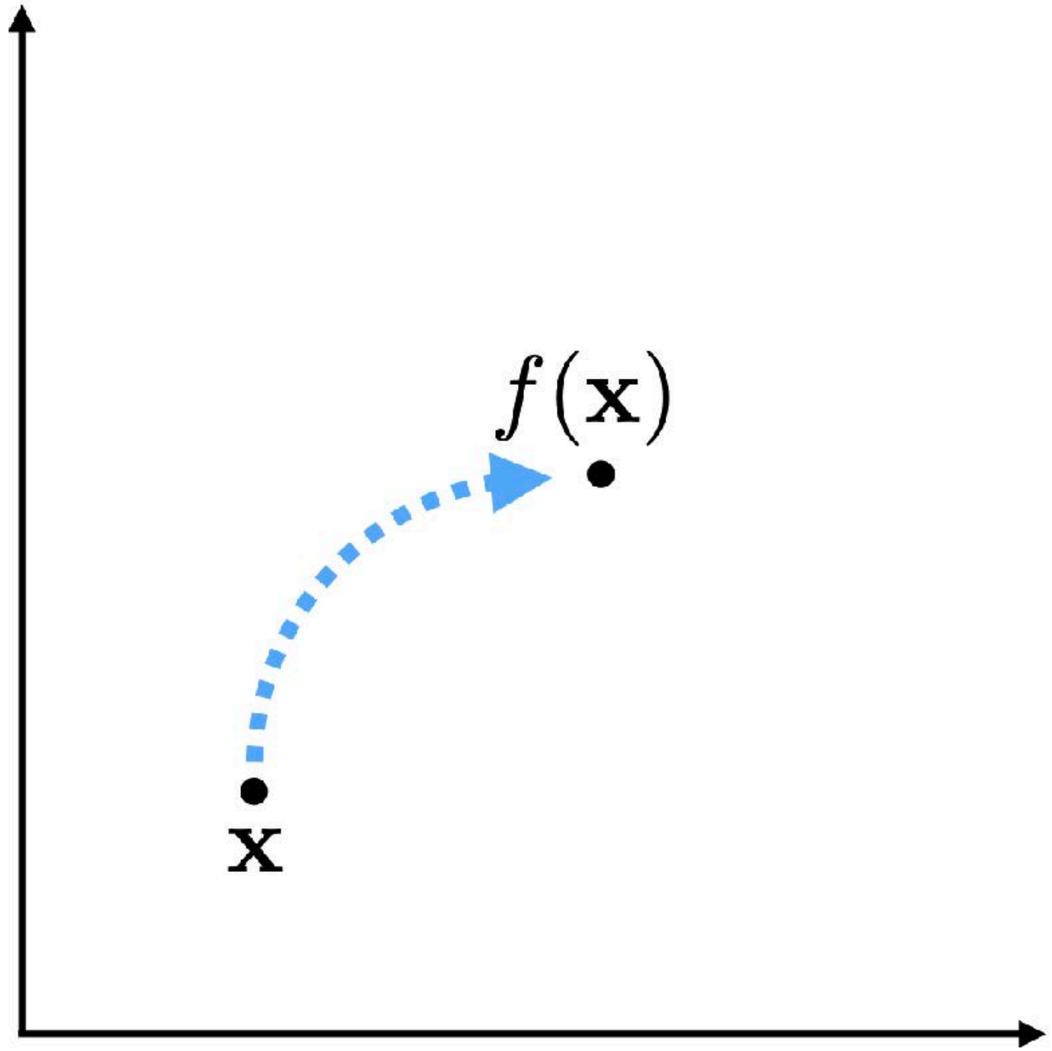
Today's focus: 3D transformations



Outline

- *Transformations and homogeneous coordinates*
- Rotation and SO(n)
- 3D Rotation representation
  - Euler Angles
  - Axis-Angle
  - Quaternion

Basic Idea: f Transforms x to f (x)



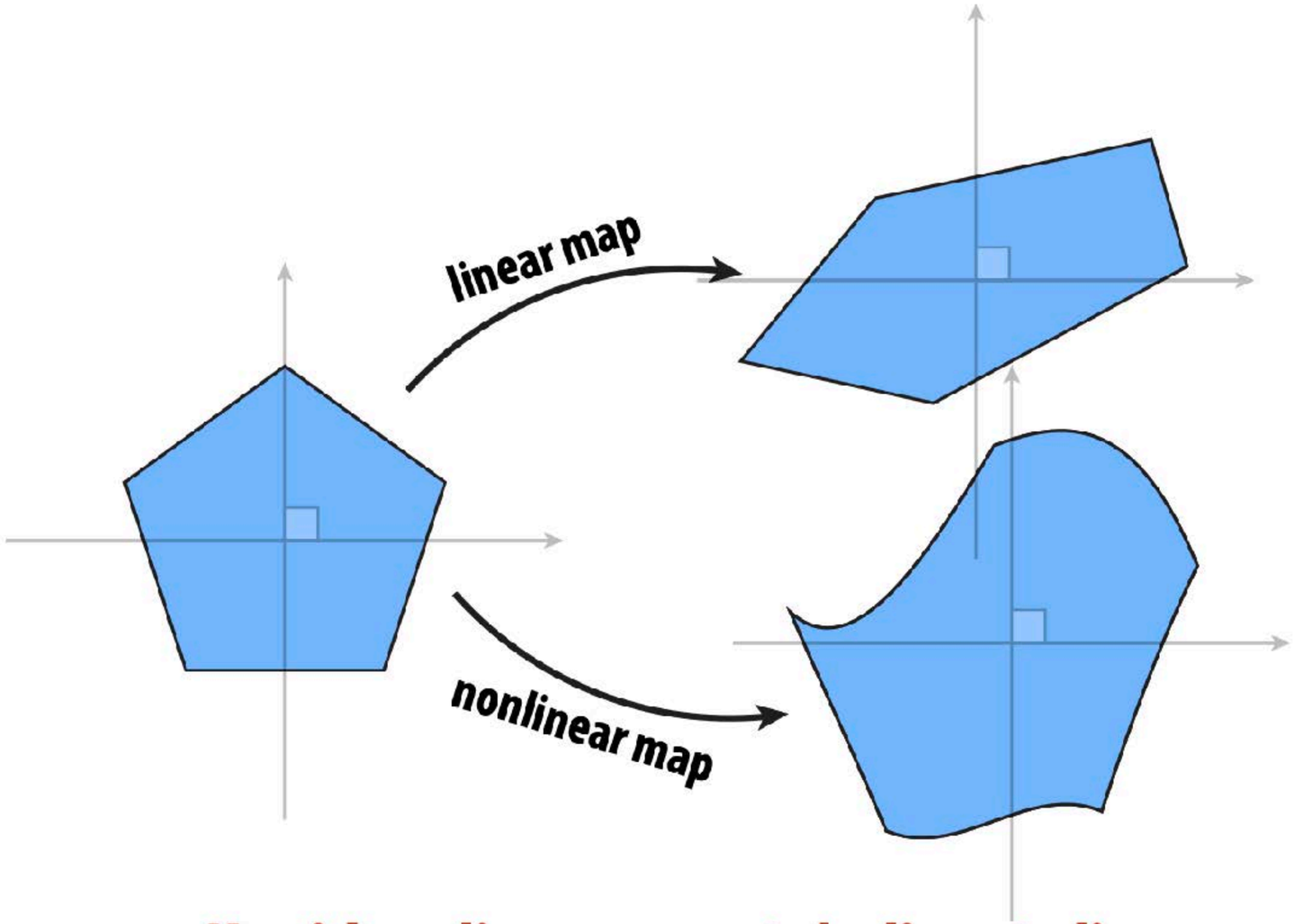
Linear Transformation

- Cheap to compute
- Composition of linear transformations is linear
  - Leads to uniform representation of transformations

$$f(\mathbf{x}+\mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y})$$

$$f(a\mathbf{x}) = af(\mathbf{x})$$

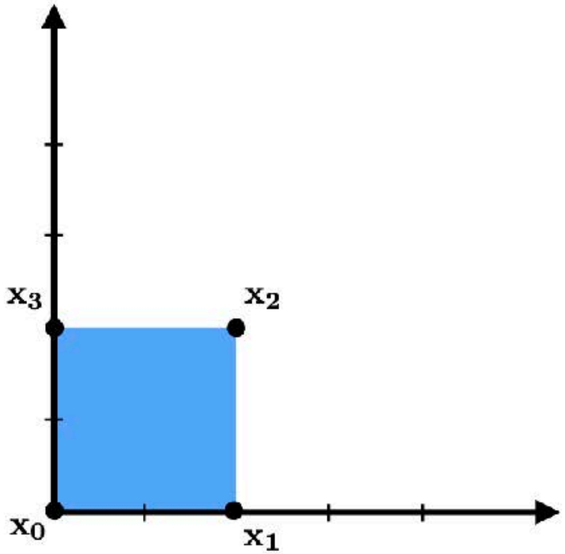
Linear Transformation Visualization

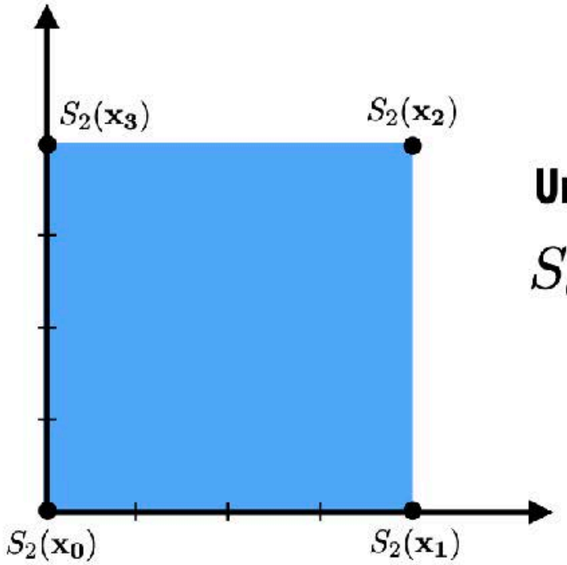


Key idea: *linear maps take lines to lines*

Slides credit: Stanford CS248

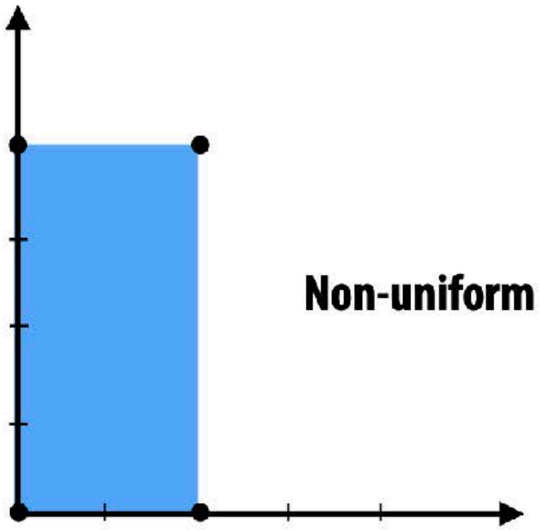
Scale





Uniform scale:

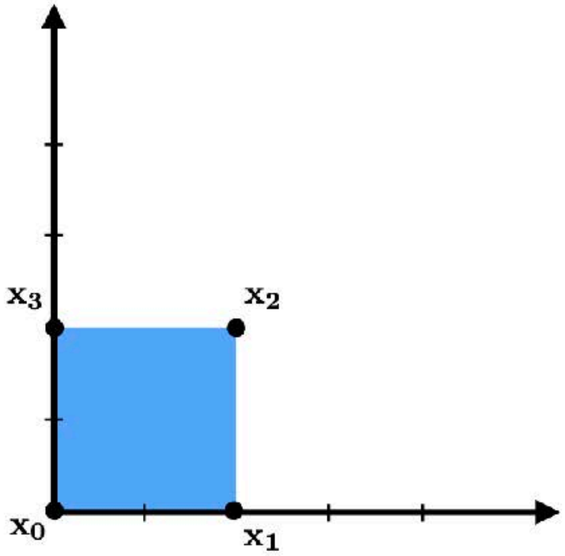
$$S_a(\mathbf{x}) = a\mathbf{x}$$

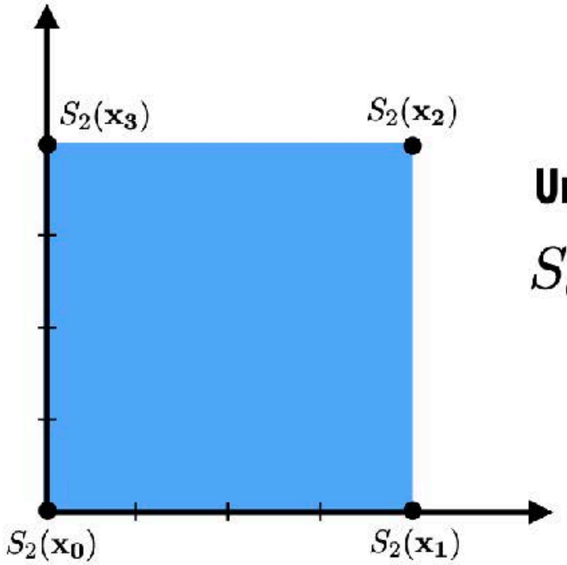


Non-uniform scale??

Slides credit: Stanford CS248

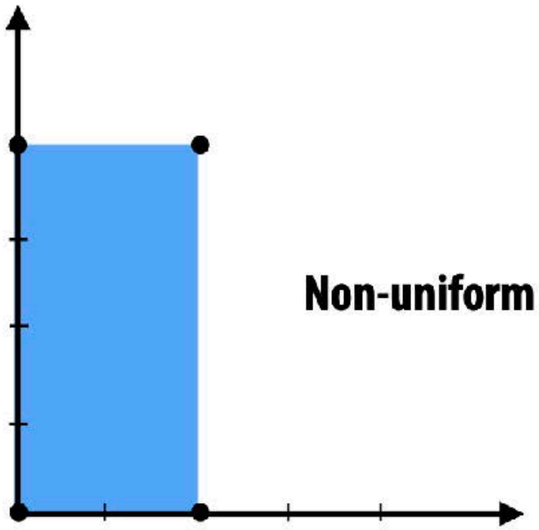
Scale





Uniform scale:

$$S_a(\mathbf{x}) = a\mathbf{x}$$



Non-uniform scale??

$$\mathbf{S_s} = \begin{bmatrix} \mathbf{s}_x & 0 \\ 0 & \mathbf{s}_y \end{bmatrix}$$

Slides credit: Stanford CS248

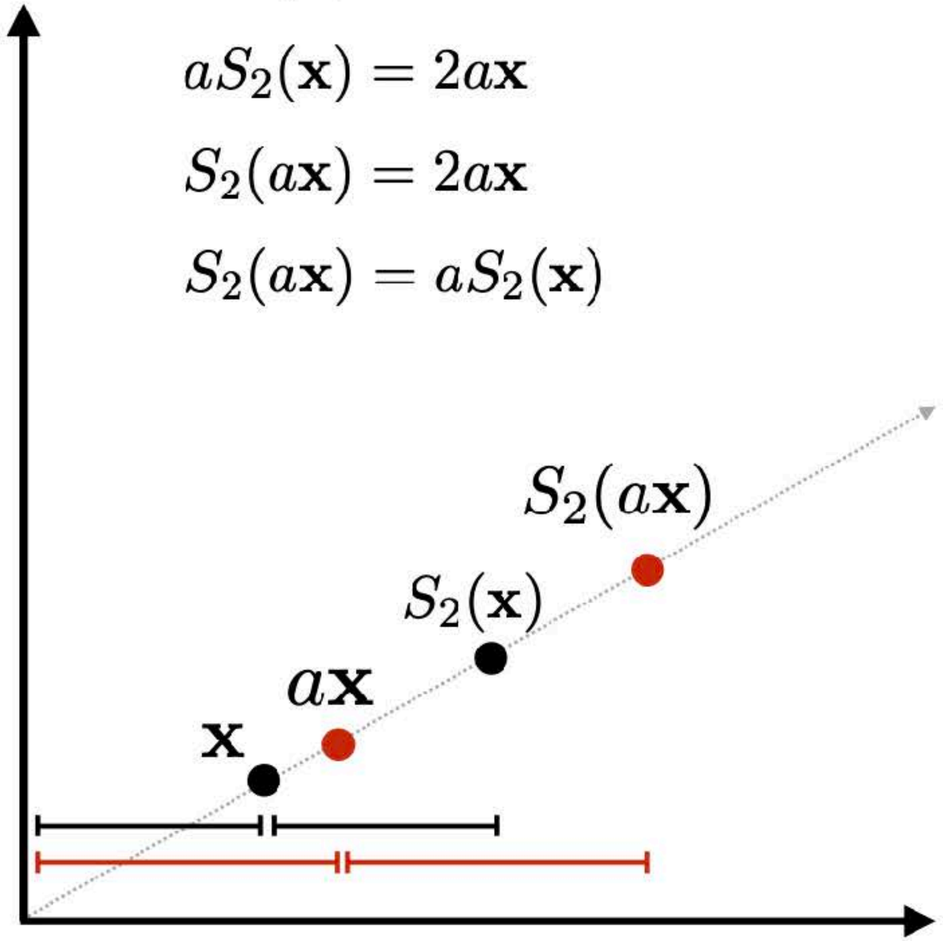
Linearity?

$$S_2(\mathbf{x}) = 2\mathbf{x}$$

$$aS_2(\mathbf{x}) = 2a\mathbf{x}$$

$$S_2(a\mathbf{x}) = 2a\mathbf{x}$$

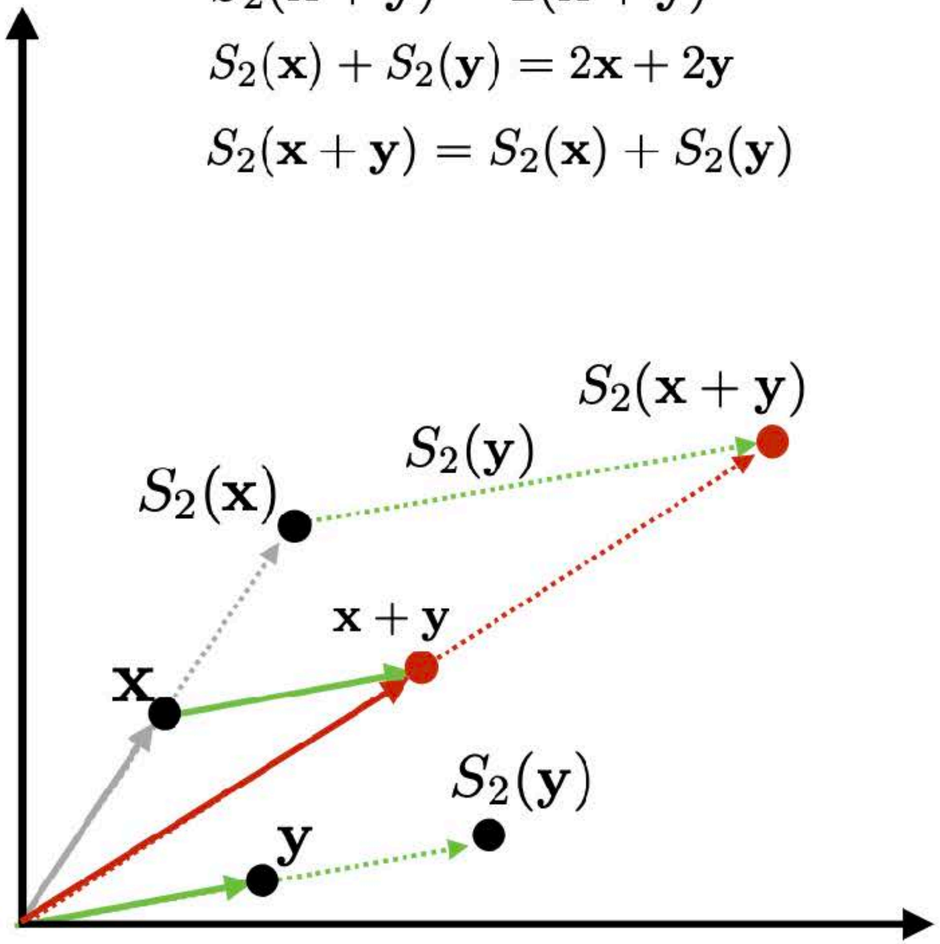
$$S_2(a\mathbf{x}) = aS_2(\mathbf{x})$$



$$S_2(\mathbf{x}+\mathbf{y}) = 2(\mathbf{x}+\mathbf{y})$$

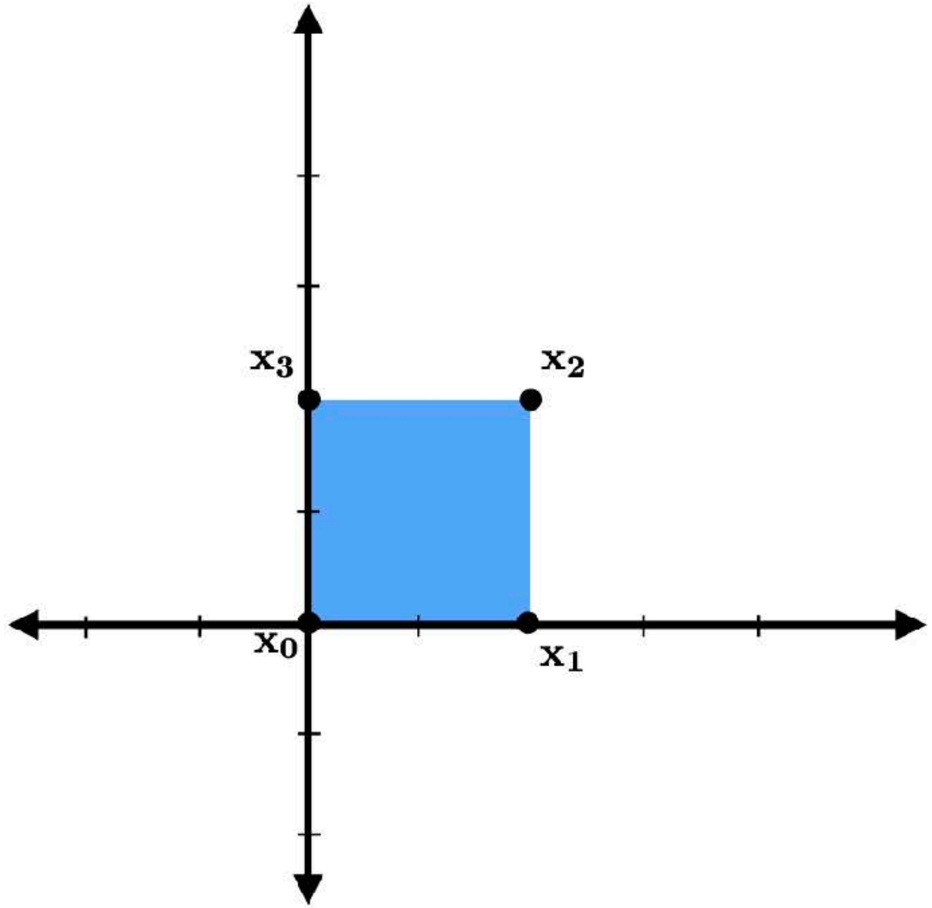
$$S_2(\mathbf{x}) + S_2(\mathbf{y}) = 2\mathbf{x} + 2\mathbf{y}$$

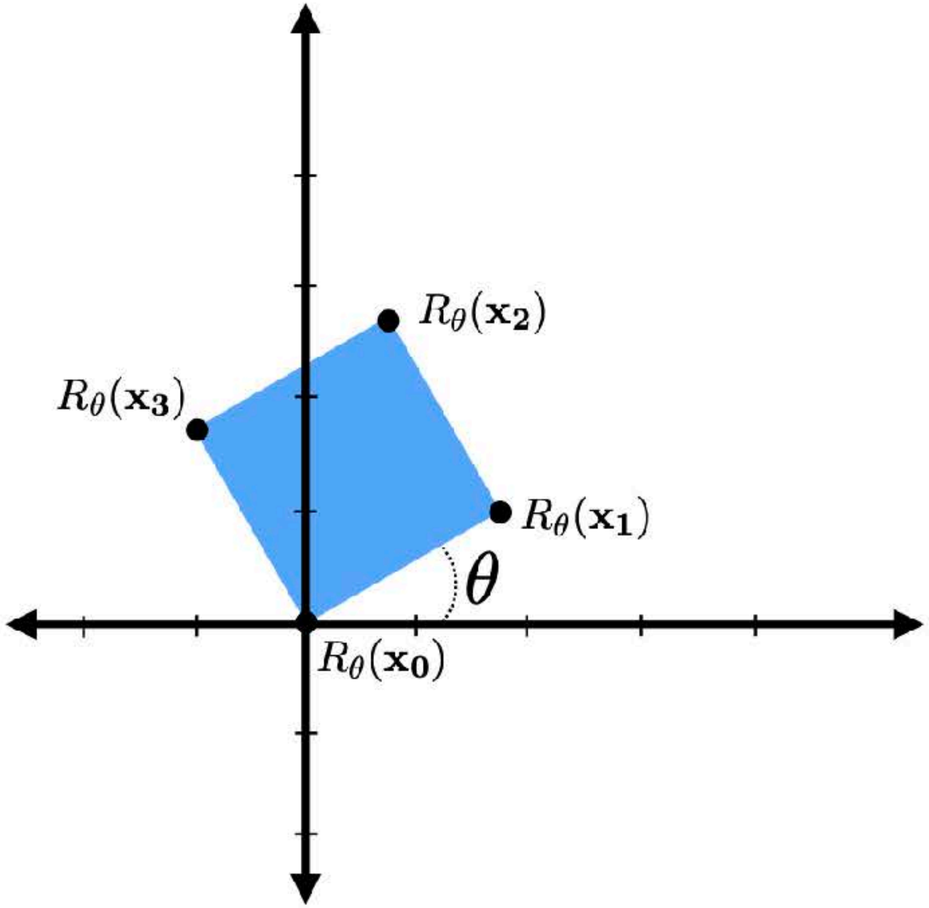
$$S_2(\mathbf{x}+\mathbf{y}) = S_2(\mathbf{x}) + S_2(\mathbf{y})$$



Slides credit: Stanford CS248

Rotation



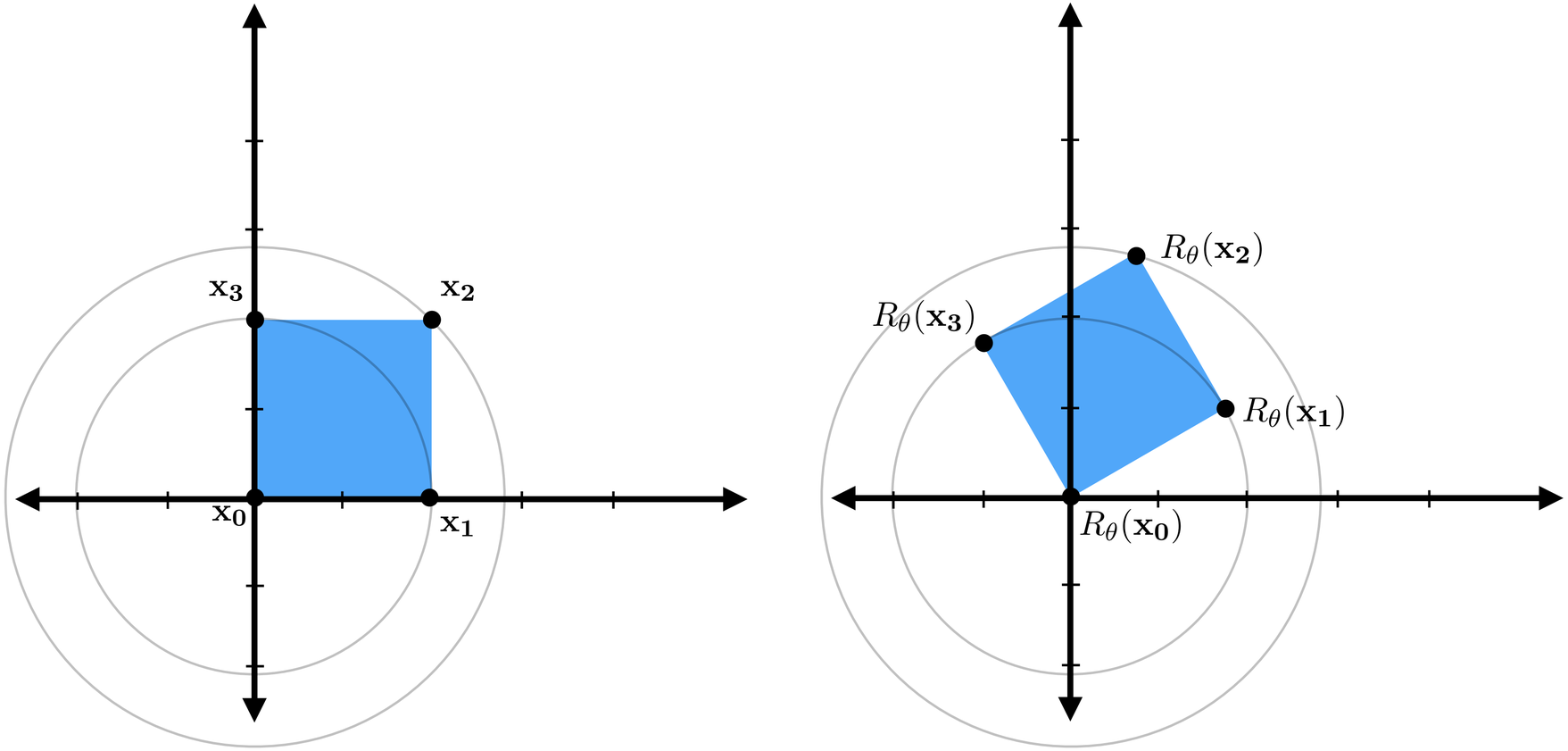


R_θ = rotate counter-clockwise by θ

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Slides credit: Stanford CS248

Rotation as circular motion



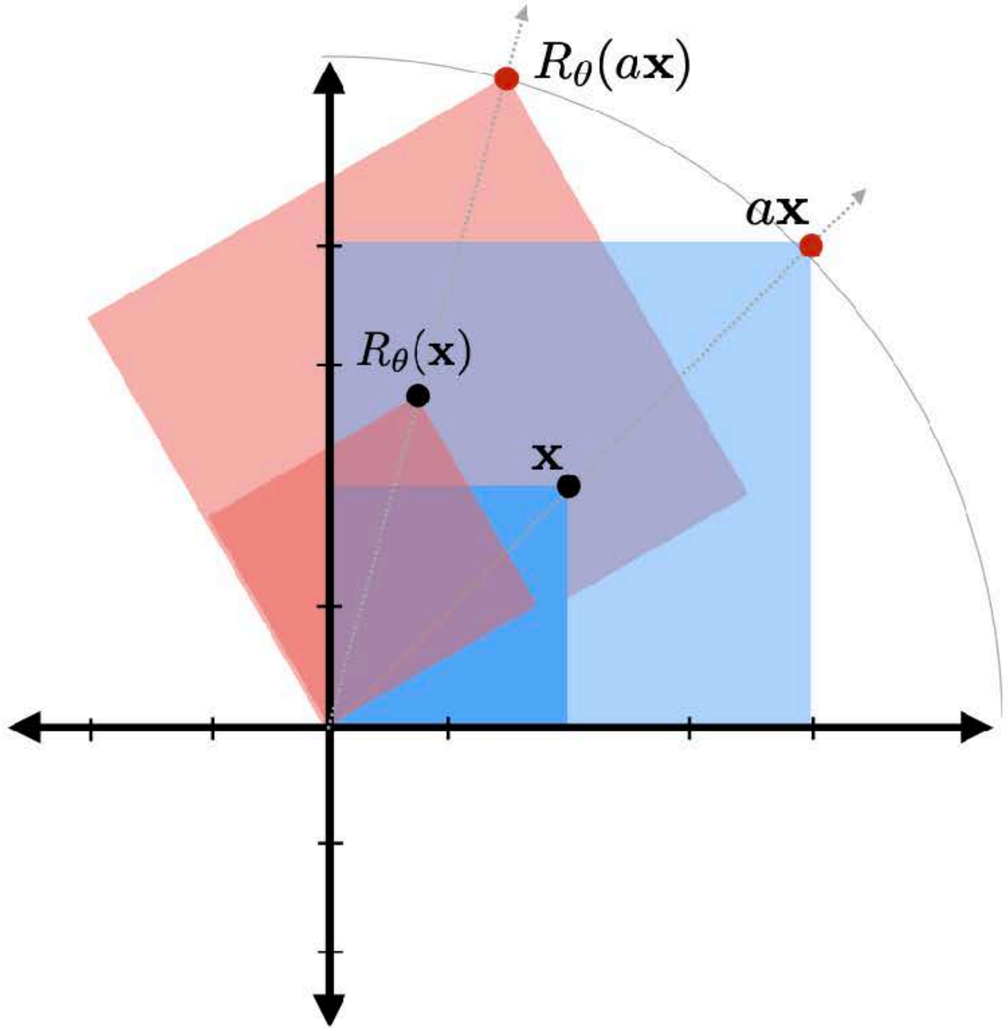
R_θ = **rotate counter-clockwise by** θ

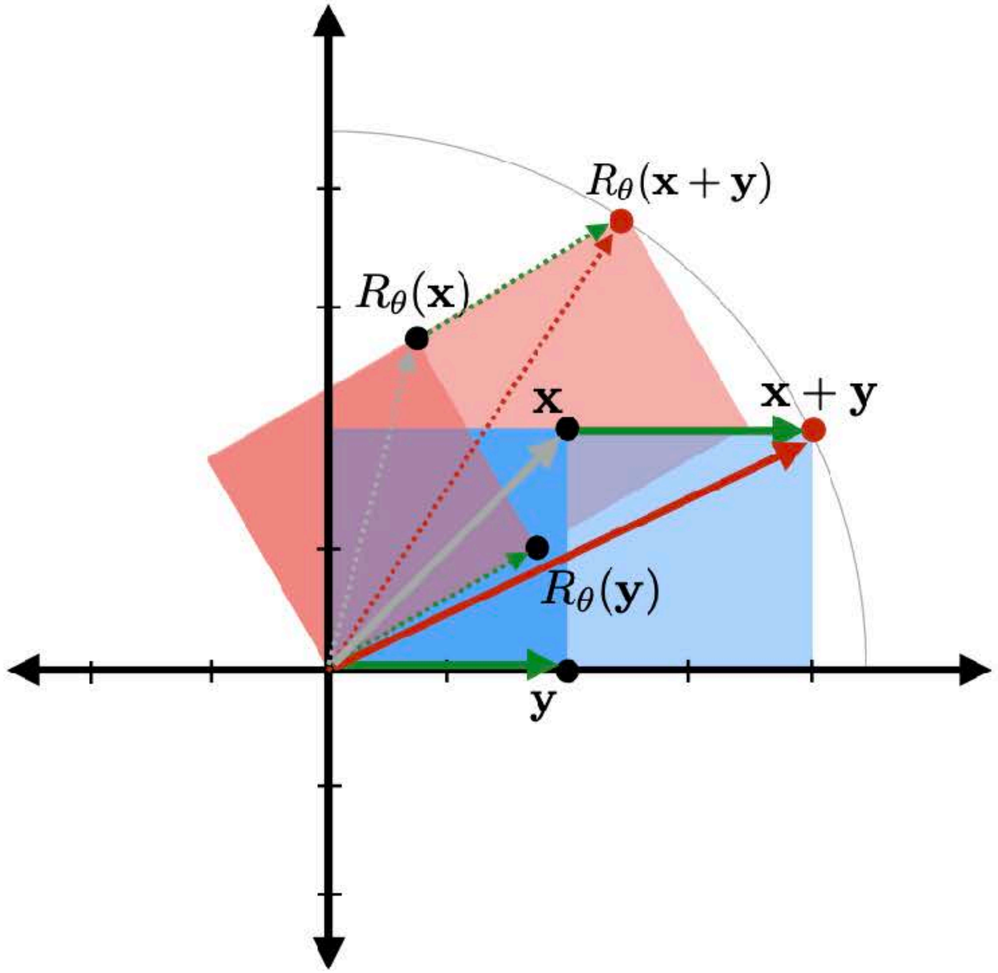
As angle changes, points move along *circular* trajectories.

Hence, rotations preserve length of vectors: $|R_\theta(\mathbf{x})| = |\mathbf{x}|$

Slides credit: Stanford CS248

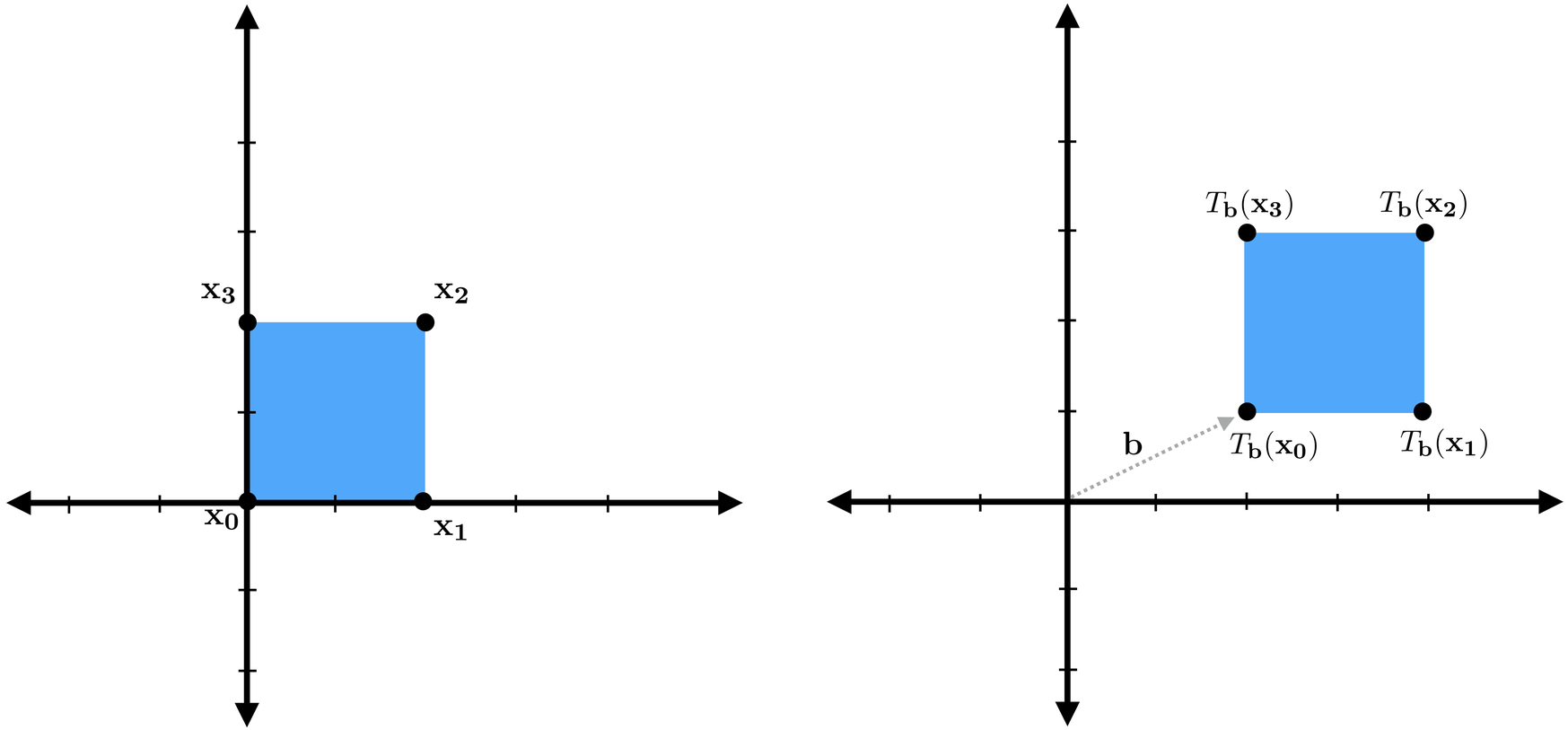
Linearity?





Slides credit: Stanford CS248

Translation

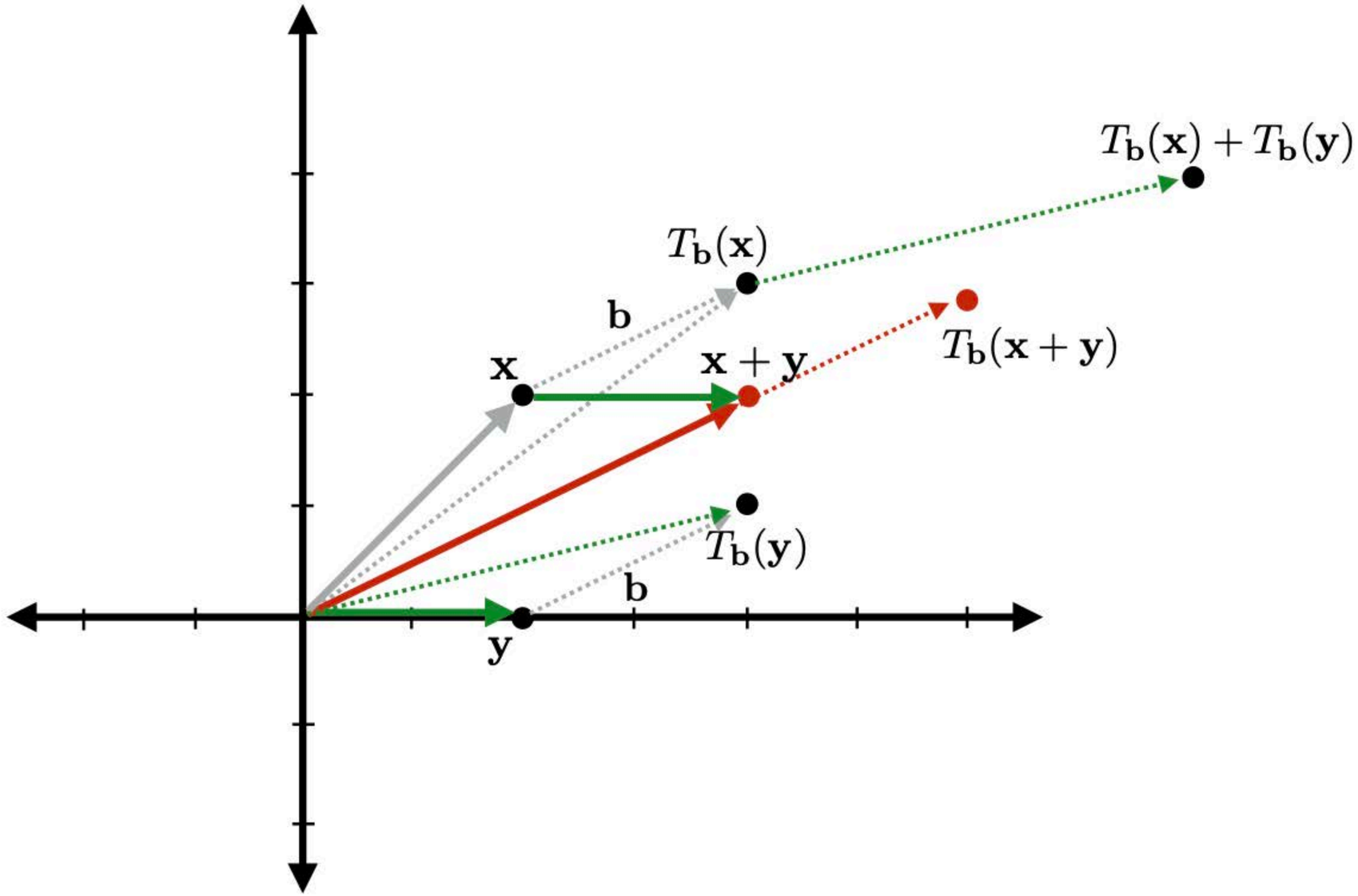


$T_\mathbf{b}$ — **"translate by b"**

$$T_\mathbf{b}(\mathbf{x}) = \mathbf{x} + \mathbf{b}$$

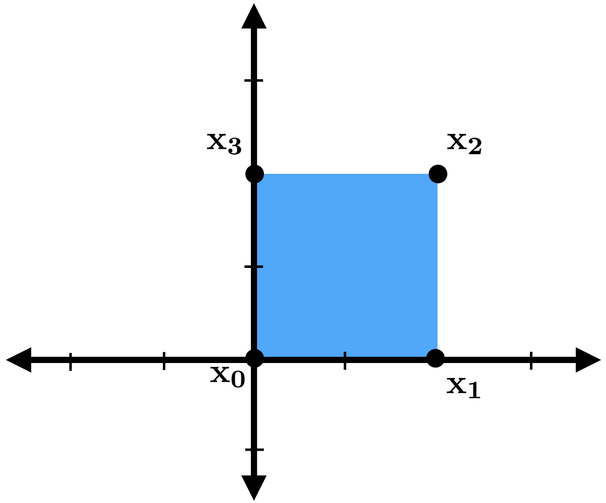
Slides credit: Stanford CS248

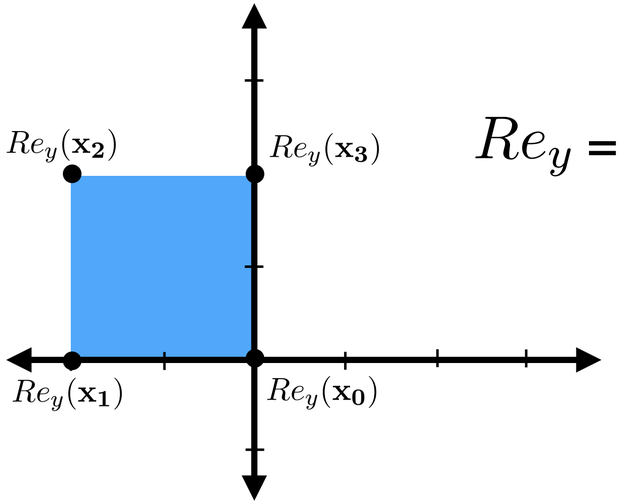
Linearity?



Slides credit: Stanford CS248

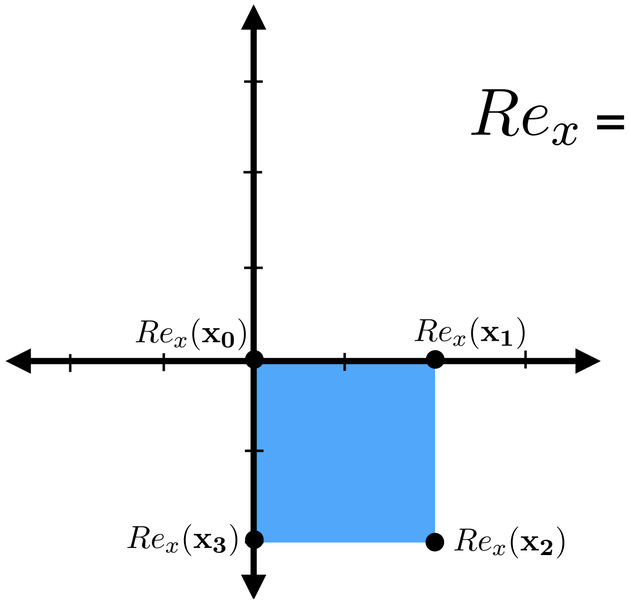
Reflection





Re_y = **reflection about *y***

$$Re_y = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

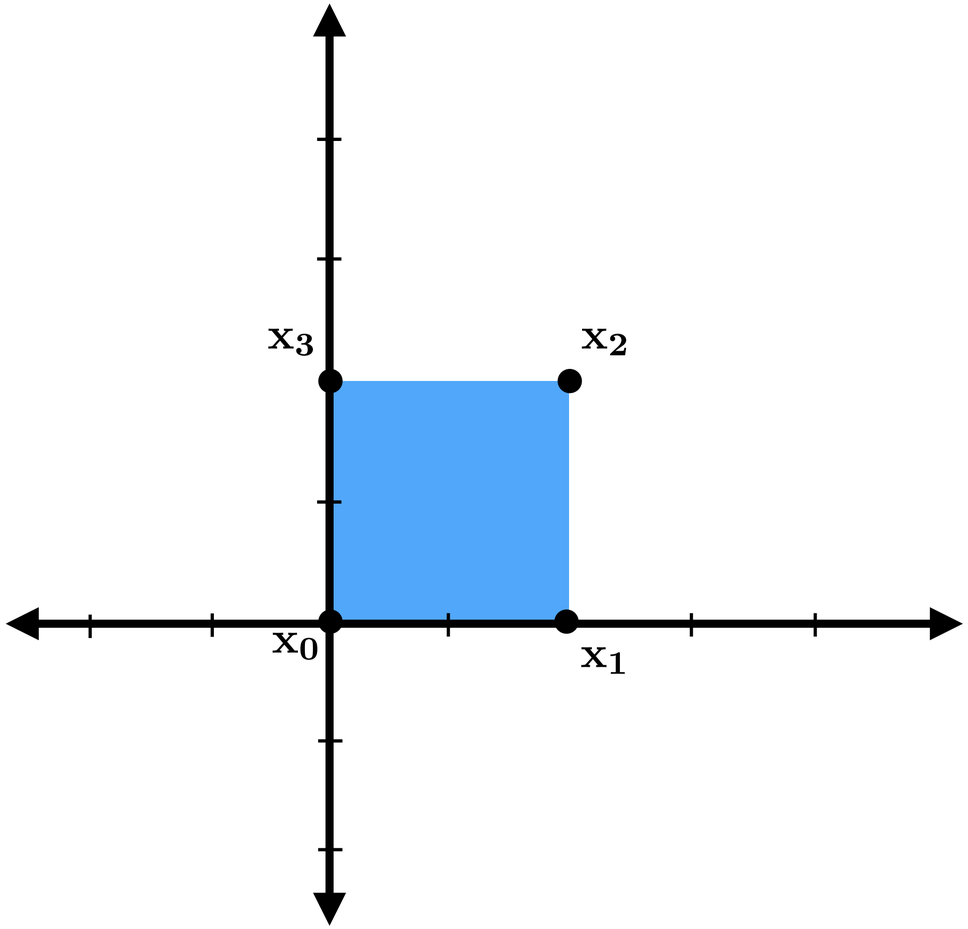


Re_x = **reflection about *x***

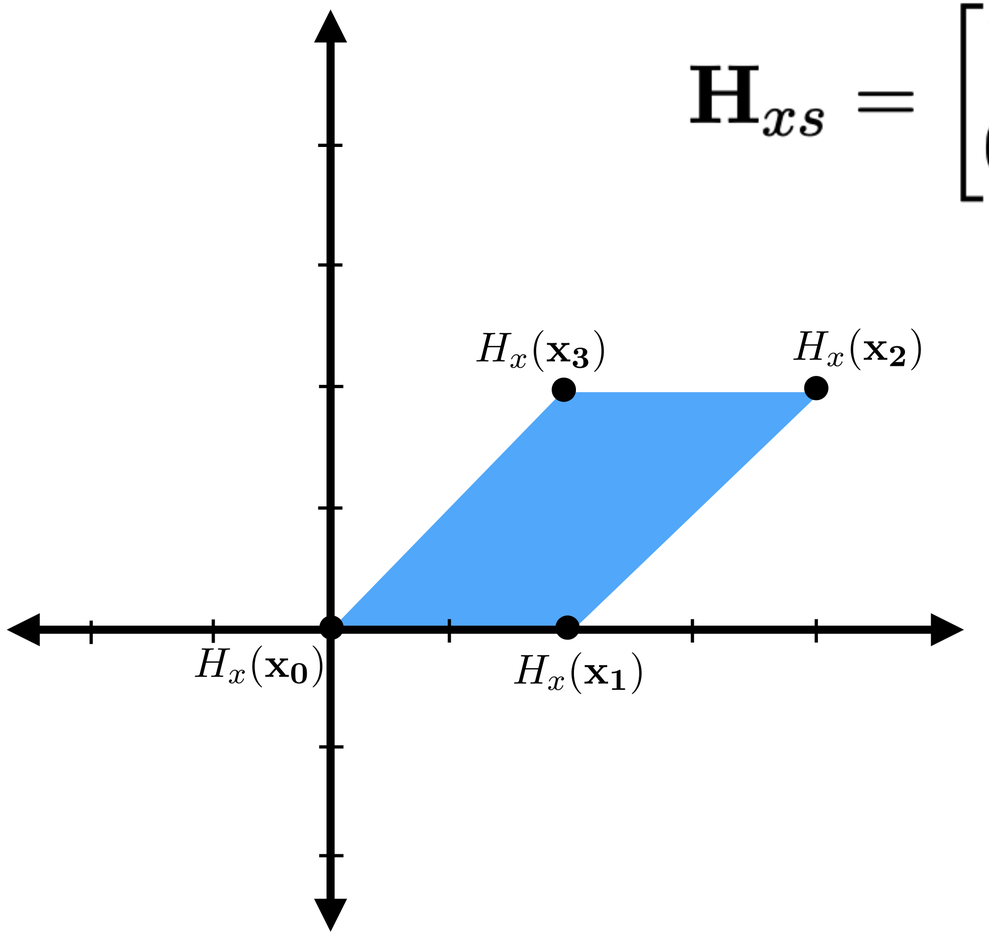
$$Re_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Slides credit: Stanford CS248

Shear (in x direction)



$$\mathbf{H}_{xs} = \begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix}$$



Slides credit: Stanford CS248

Summary of Transformations

Linear:

$$f(\mathbf{x}+\mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y})$$
$$f(a\mathbf{x}) = af(\mathbf{x})$$

Scale
Rotation
Reflection
Shear

Not linear:

Translation

Affine:

Composition of linear transform + translation

$$f(\mathbf{x}) = g(\mathbf{x}) + \mathbf{b}$$

Euclidean: (Isometries)

Preserve distance between points (preserves length)

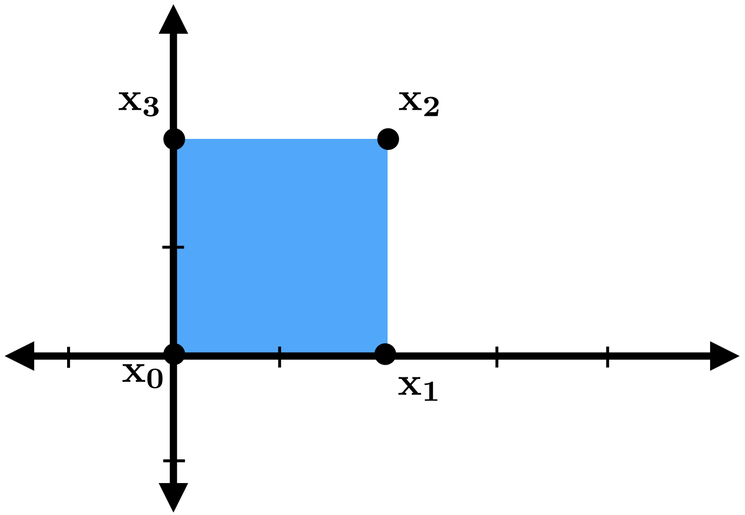
$$|f(\mathbf{x}) - f(\mathbf{y})| = |\mathbf{x} - \mathbf{y}|$$

Translation
Rotation
Reflection

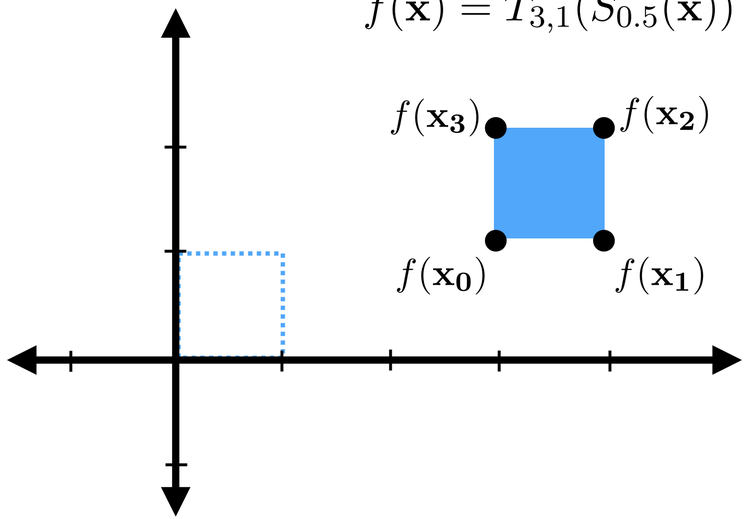
"Rigid body" transformations are distance-preserving motions that also preserve *orientation* (i.e., does not include reflection)

Slides credit: Stanford CS248

Composition of Transformations



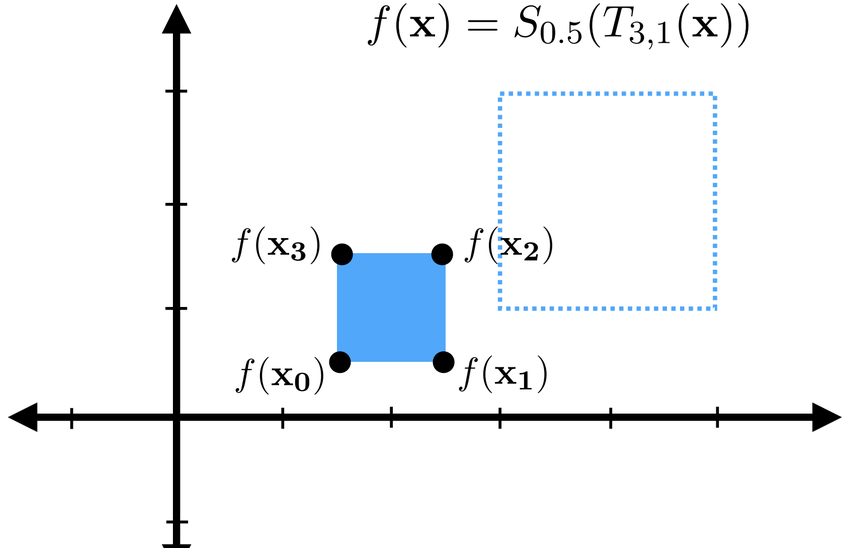
$f(\mathbf{x}) = T_{3,1}(S_{0.5}(\mathbf{x}))$



Note: order of composition matters

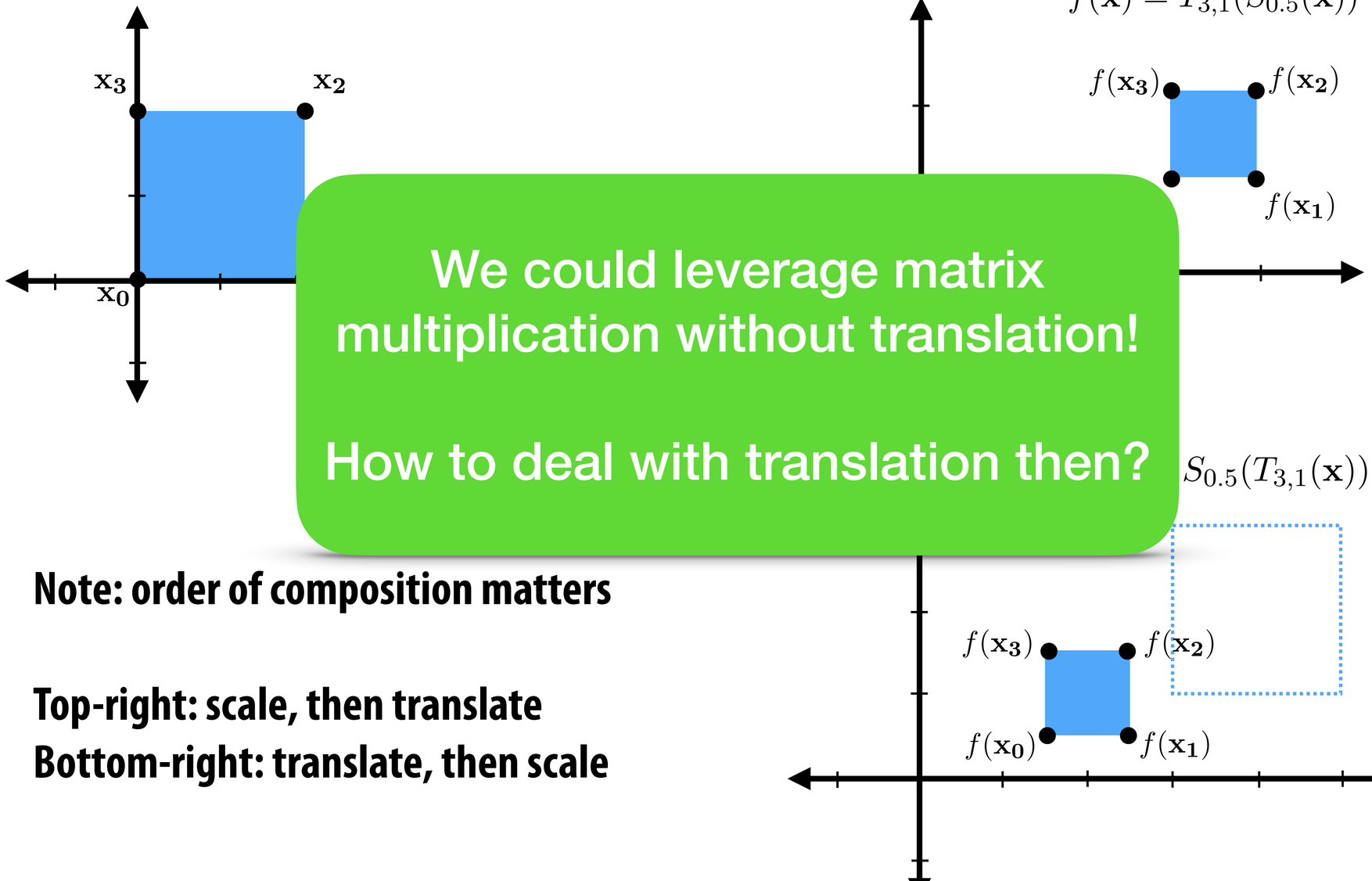
Top-right: scale, then translate
Bottom-right: translate, then scale

$f(\mathbf{x}) = S_{0.5}(T_{3,1}(\mathbf{x}))$



Slides credit: Stanford CS248

Composition of Transformations



Note: order of composition matters

Top-right: scale, then translate
Bottom-right: translate, then scale

Slides credit: Stanford CS248

Homogeneous Coordinates (in 2D)

Idea: represent 2D points with THREE values ("homogeneous coordinates")

So the point (x, y) is represented as the 3-vector: $\begin{bmatrix} x & y & 1 \end{bmatrix}^T$

And transformations are represented a 3x3 matrices that transform these vectors.

Recover final 2D coordinates by dividing by "extra" (third) coordinate

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow \begin{bmatrix} x/w \\ y/w \end{bmatrix}$$

Slides credit: Stanford CS248

Scale and Rotation with 2D-H

- **For transformations that are already linear, not much changes:**

$$\mathbf{S_s} = \begin{bmatrix} \mathbf{S}_x & 0 & 0 \\ 0 & \mathbf{S}_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Notice that the last row/column doesn't do anything interesting. E.g., for scaling:

$$\mathbf{S_s} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{S}_x x \\ \mathbf{S}_y y \\ 1 \end{bmatrix}$$

Now we divide by the 3rd coordinate to get our final 2D coordinates (not too exciting!)

$$\begin{bmatrix} \mathbf{S}_x x \\ \mathbf{S}_y y \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} \mathbf{S}_x x/1 \\ \mathbf{S}_y y/1 \end{bmatrix} = \begin{bmatrix} \mathbf{S}_x x \\ \mathbf{S}_y y \end{bmatrix}$$

Slides credit: Stanford CS248

Translation with 2D-H

Translation expressed as 3x3 matrix multiplication:

$$\mathbf{T_b} = \begin{bmatrix} 1 & 0 & \mathbf{b}_x \\ 0 & 1 & \mathbf{b}_y \\ 0 & 0 & 1 \end{bmatrix}$$

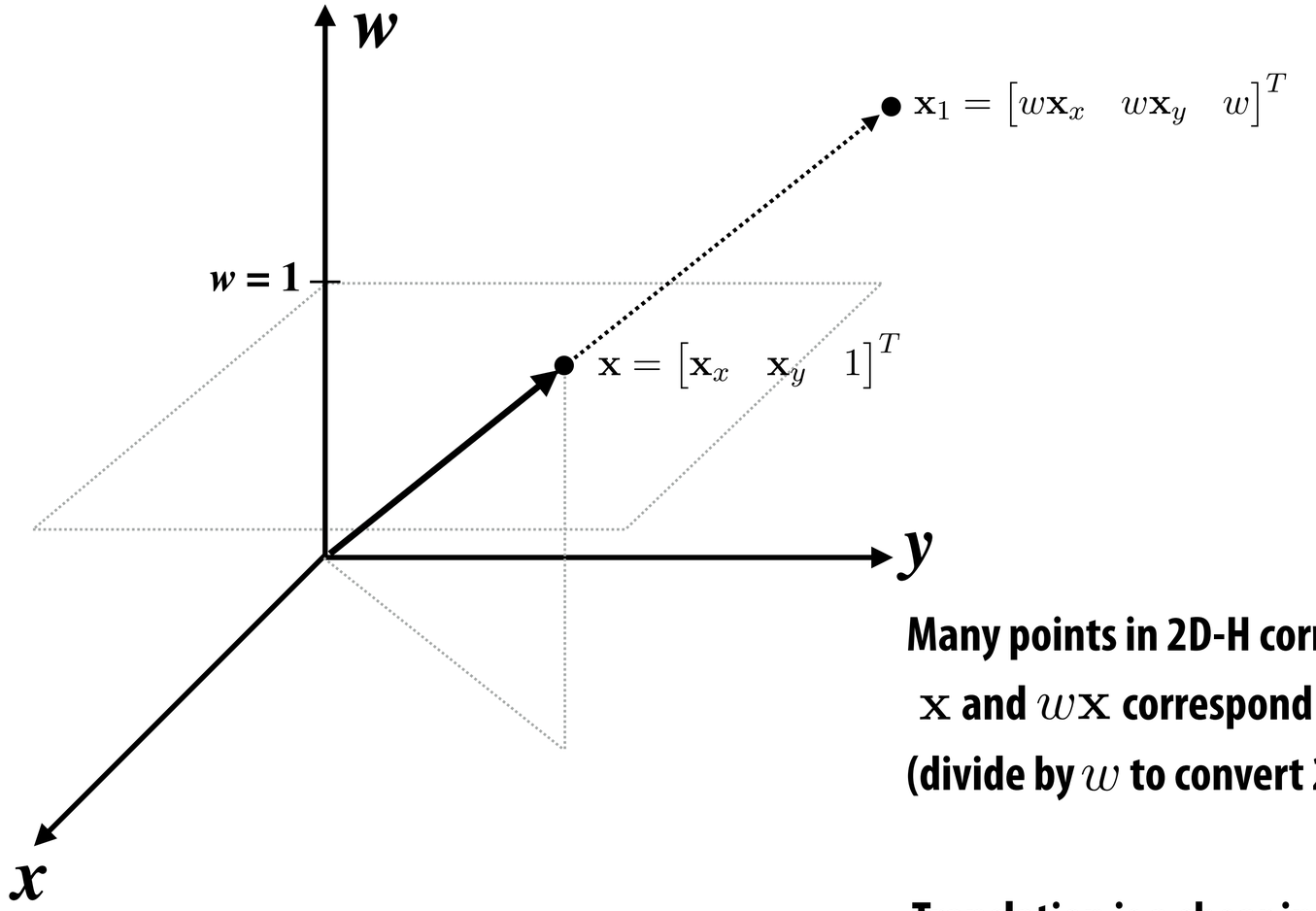
$$\mathbf{T_b x} = \begin{bmatrix} 1 & 0 & \mathbf{b}_x \\ 0 & 1 & \mathbf{b}_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_x \\ \mathbf{x}_y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_x + \mathbf{b}_x \\ \mathbf{x}_y + \mathbf{b}_y \\ 1 \end{bmatrix}$$

(remember: just a linear combination of columns!)

Cool: homogeneous coordinates let us encode translations as *linear* transformations!

Slides credit: Stanford CS248

Intuition behind Homogeneous Coordinates



Many points in 2D-H correspond to same point in 2D

$\mathbf{x}$ and $w\mathbf{x}$ correspond to the same 2D point

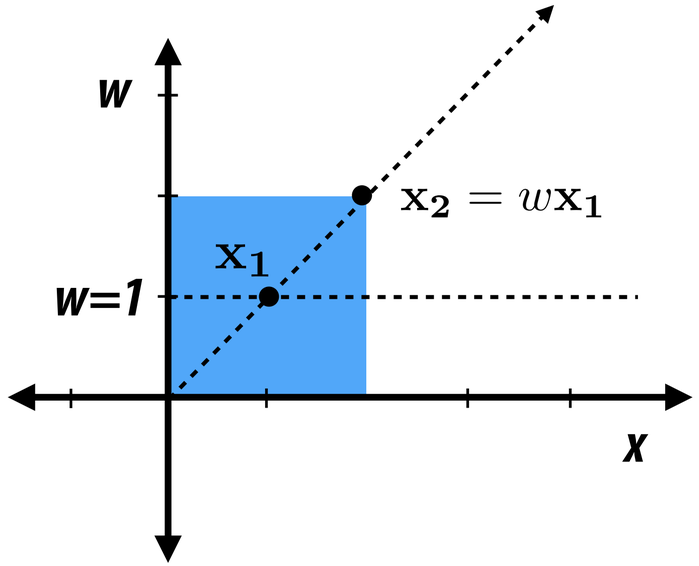
(divide by w to convert 2D-H back to 2D)

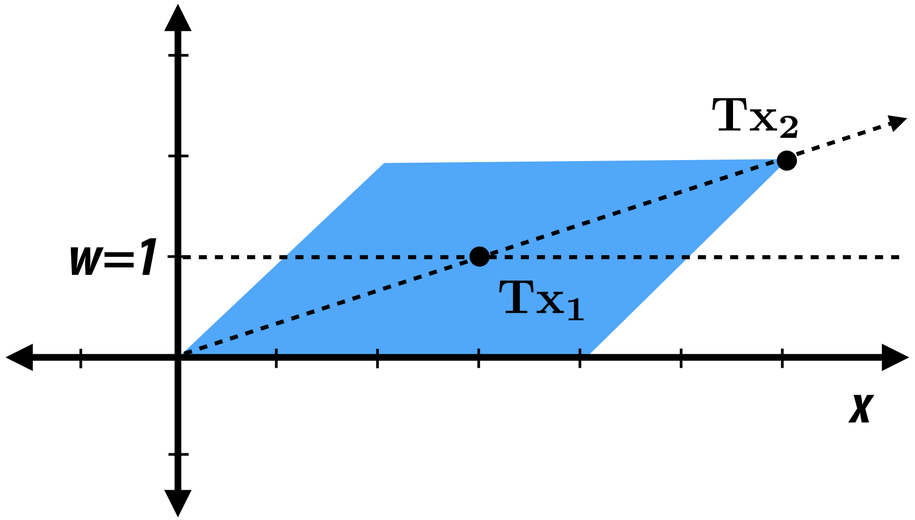
Translation is a shear in x and y in 2D-H space

$$\mathbf{T}_\mathbf{b}\mathbf{x} = \begin{bmatrix} 1 & 0 & \mathbf{b}_x \\ 0 & 1 & \mathbf{b}_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w\mathbf{x}_x \\ w\mathbf{x}_y \\ w \end{bmatrix} = \begin{bmatrix} w\mathbf{x}_x + w\mathbf{b}_x \\ w\mathbf{x}_y + w\mathbf{b}_y \\ w \end{bmatrix}$$

Slides credit: Stanford CS248

Intuition behind Homogeneous Coordinates





For simplicity, consider 1D-H:

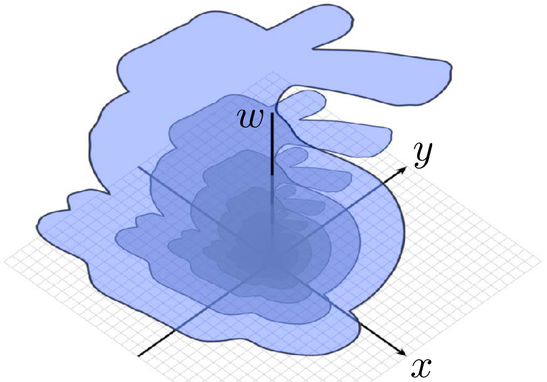
Translate by *t*=2: $\mathbf{T} = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$

Recall: this is a shear in homogeneous *x*.

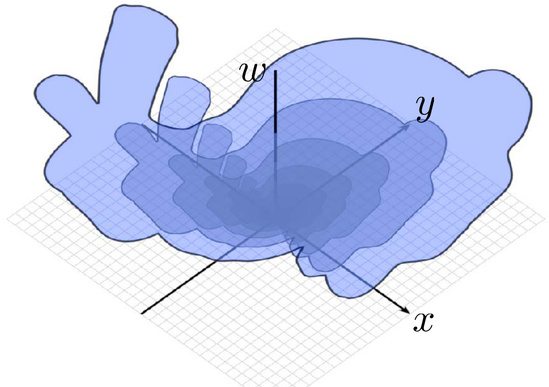
1D translation is affine in 1D ($x + t$),
but it is linear in 1D-H

Slides credit: Stanford CS248

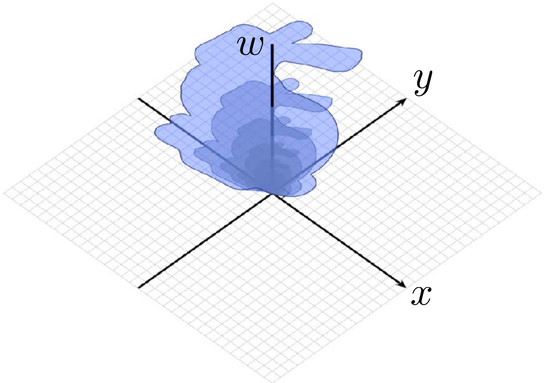
Visualizing 2D Transformations in 2D-H



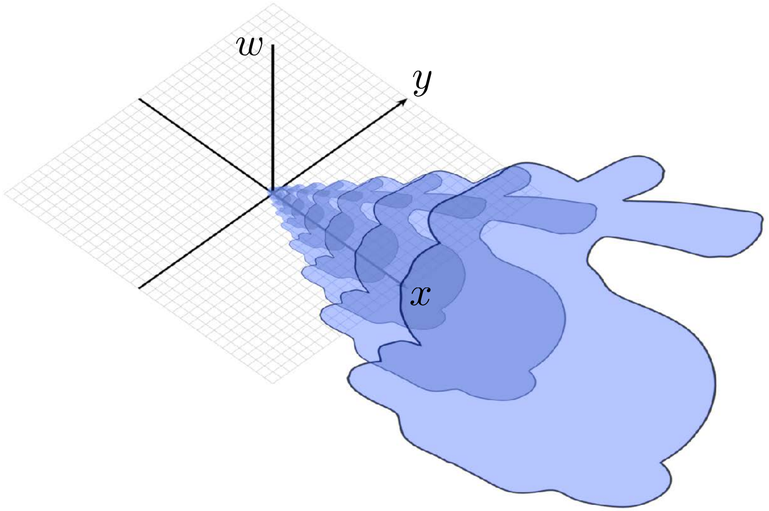
Original shape in 2D can be viewed as many copies, uniformly scaled by w.



2D rotation ↔ rotate around w



2D scale ↔ scale x and y; preserve w
(Question: what happens to 2D shape if you scale x, y, and w uniformly?)



2D translate ↔ shear in 2D-H
(LINEAR!)

Slides credit: Stanford CS248

Moving to 3D (and 3D-H)

Represent 3D transformations as 3x3 matrices and 3D-H transformations as 4x4 matrices

Scale:

3D

$$\mathbf{S_s} = \begin{bmatrix} \mathbf{S}_x & 0 & 0 \\ 0 & \mathbf{S}_y & 0 \\ 0 & 0 & \mathbf{S}_z \end{bmatrix}$$

3D-H

$$\mathbf{S_s} = \begin{bmatrix} \mathbf{S}_x & 0 & 0 & 0 \\ 0 & \mathbf{S}_y & 0 & 0 \\ 0 & 0 & \mathbf{S}_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Shear (in x, based on y,z position):

$$\mathbf{H}_{x,\mathbf{d}} = \begin{bmatrix} 1 & \mathbf{d}_y & \mathbf{d}_z \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{H}_{x,\mathbf{d}} = \begin{bmatrix} 1 & \mathbf{d}_y & \mathbf{d}_z & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Translate:

3D-H

$$\mathbf{T_b} = \begin{bmatrix} 1 & 0 & 0 & \mathbf{b}_x \\ 0 & 1 & 0 & \mathbf{b}_y \\ 0 & 0 & 1 & \mathbf{b}_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

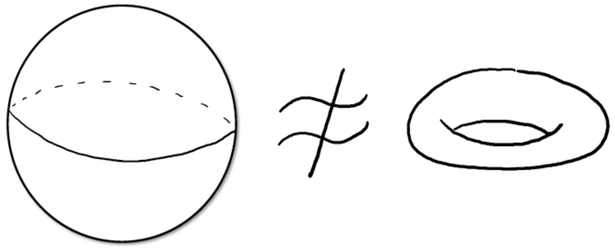
Slides credit: Stanford CS248

Outline

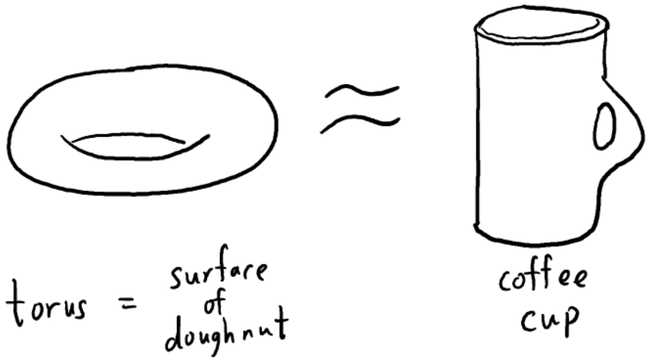
- Transformations and homogeneous coordinates
- ***Rotation and SO(n)***
- 3D Rotation representation
  - Euler Angles
  - Axis-Angle
  - Quaternion

Prereq: Topology

- Topology: Structural Properties of a Manifold



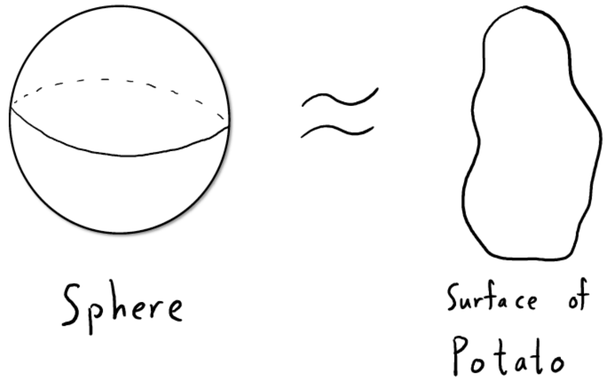
- Two surfaces M and N are *topologically equivalent* if there is a **differentiable bijection** between M and N

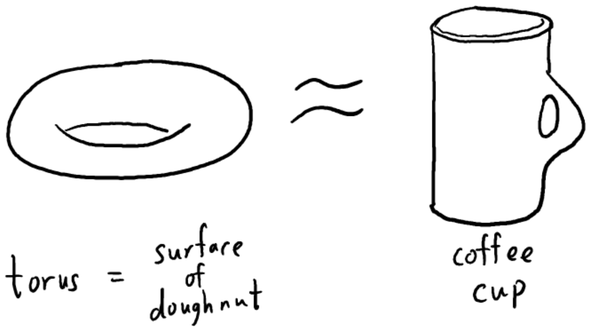


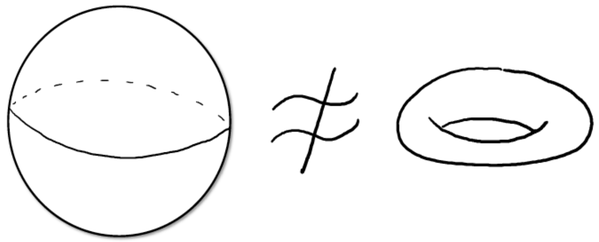
Slides credit: Hao Su

Prereq: Topology

- More examples:







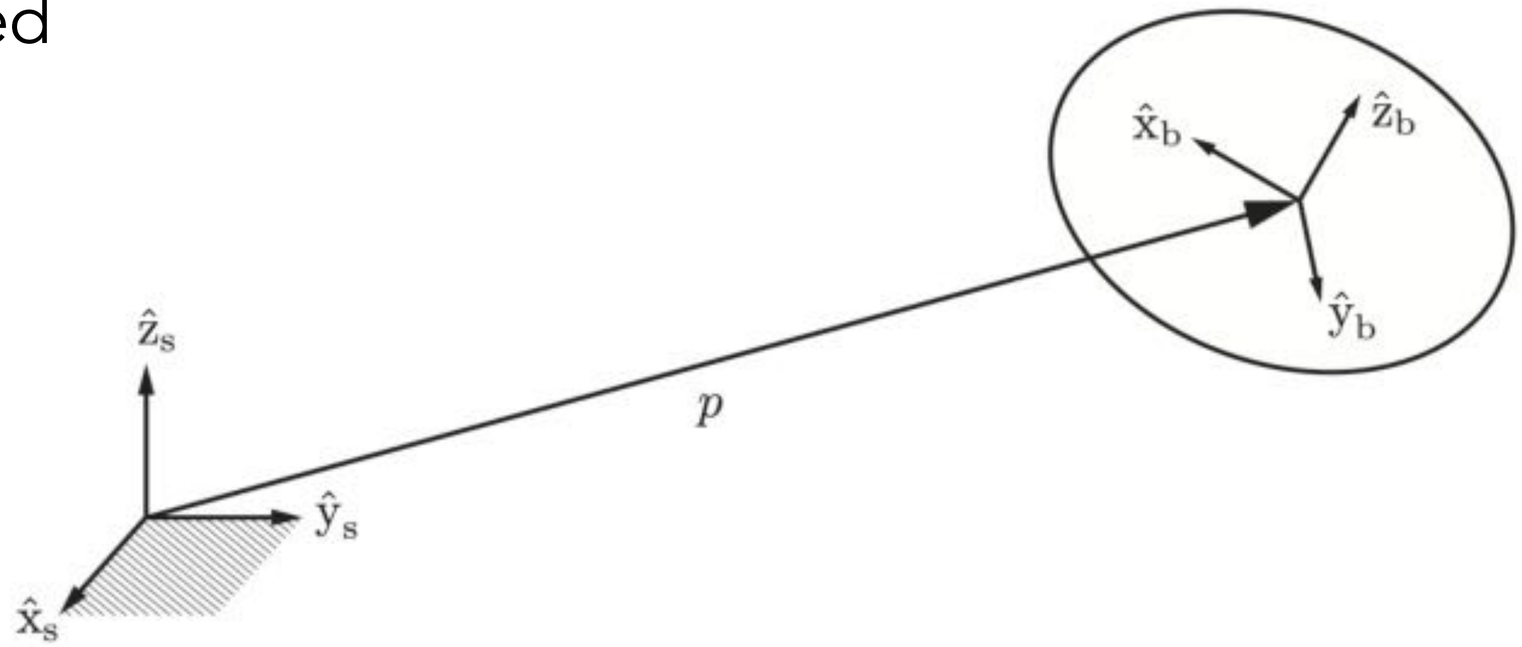
Slides credit: Hao Su

Orientation

We use “rotation” to represent the relative orientation between two frames

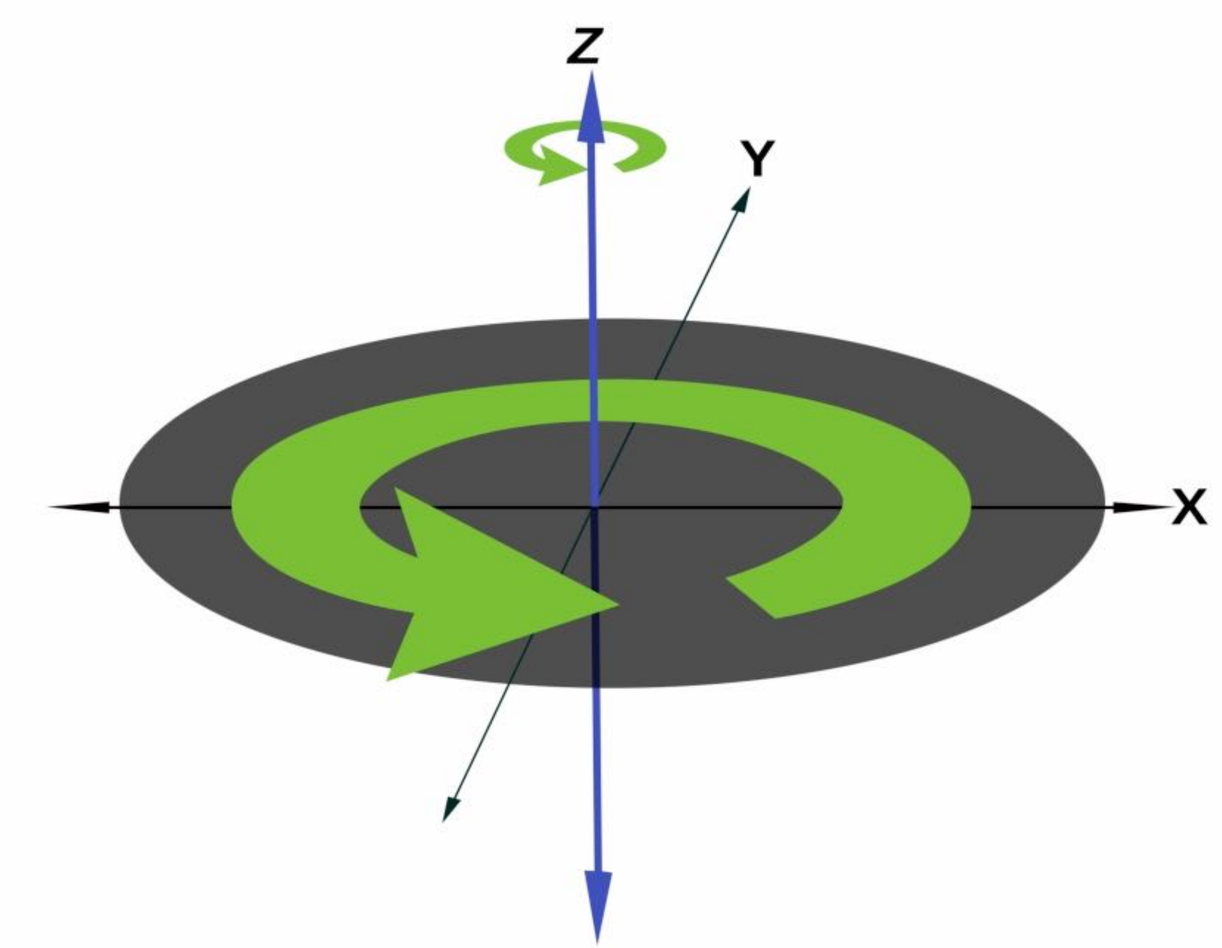
For example,

- Space Frame: $\{ s \} = \{ x_s , y_s , z_s \}$
- Body Frame: $\{ b \} = \{ x_b , y_b , z_b \}$
- R_{sb} rotates the frame of the space to the frame of the body after the origins are aligned



Slides credit: Hao Su

Rotation in $\mathbb{R}^2$



1 Degree of Freedom

Slides credit: Hao Su

Rotation in $\mathbb{R}^3$

3 Degree of Freedoms

Slides credit: Hao Su

The Set of Rotations

- $SO(n) = \{R \in \mathbb{R}^{n \times n} : \det(R) = 1, RR^T = I\}$
- $SO(n)$: "Special Orthogonal Group"

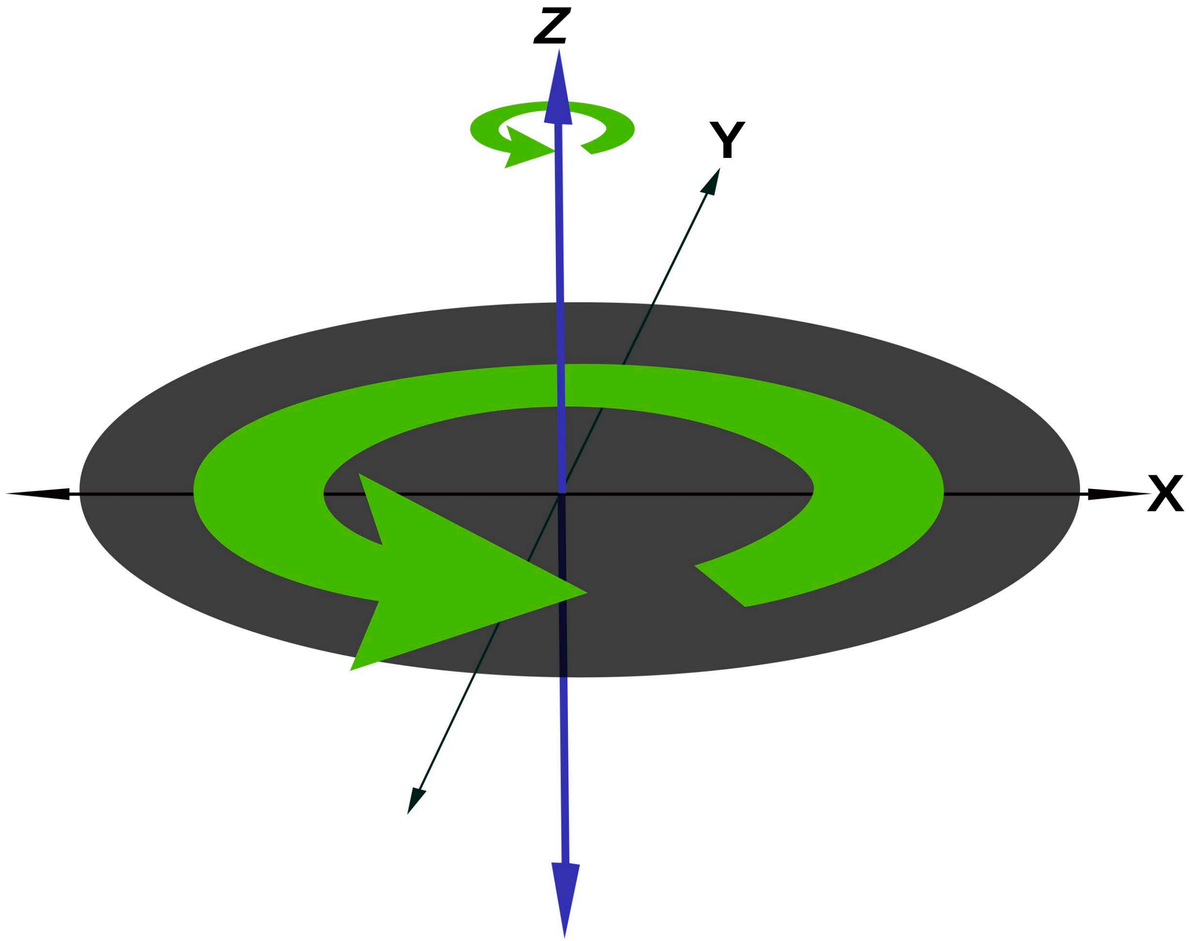
- "Group": a group under the *matrix multiplication*
- "Orthogonal": $RR^T = I$
- "Special": $\det(R) = 1$

- $SO(2)$: 2D rotations, 1 DoF
- $SO(3)$: 3D rotations, 3 DoF

Slides credit: Hao Su

Topology of SO(n)

- The topology of $SO(2)$ is the same as a circle

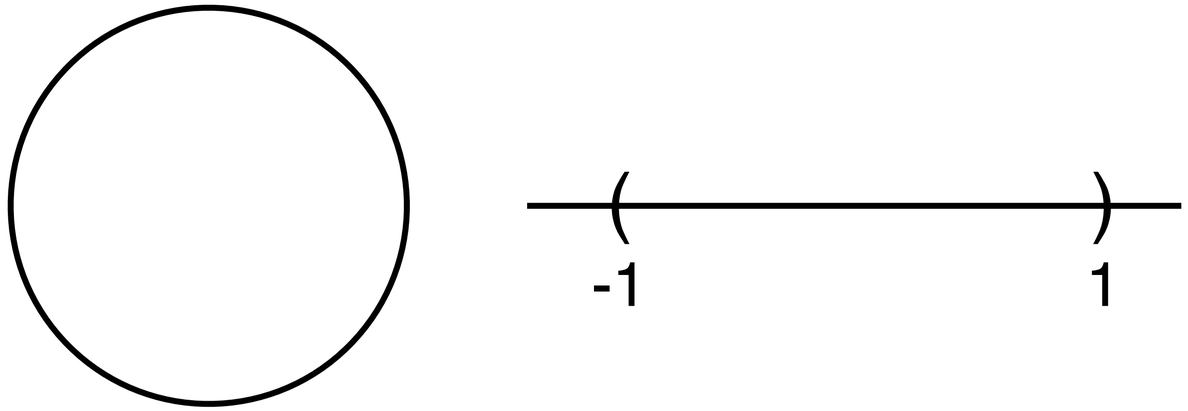


Slides credit: Hao Su

Topology of SO(n)

- Circles do not have the same topology as $(-1,1)^n$

$\Longrightarrow$ No differentiable bijections between $SO(2)$ and $(-1,1)^n$



- The topology of $SO(3)$ is also different from $(-1,1)^n$

Why do we care about the topology?

Slides credit: Hao Su

Parameterizing Rotation in Networks is Tricky

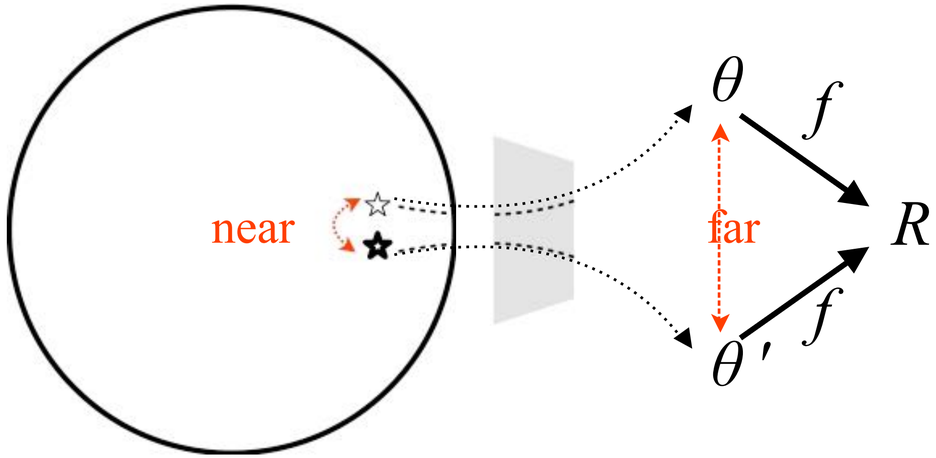
- An ideal parameterization $f(\theta) : U \mapsto SO(2)$ to use in networks:
  1. The domain is $(-l, l)^n$ (as network output)

Slides credit: Hao Su

Parameterizing Rotation in Networks is Tricky

- An ideal parameterization $f(\theta) : U \mapsto SO(2)$ to use in networks:
  1. The domain is $(-l, l)^n$ (as network output)
  2. f is a differentiable *bijection*

Otherwise:



- If input data points to network are close, but the θ predictions happen to be far after convergence, the network (a continuous function) will make awful predictions between the two data points !

- Need special network design to overcome the issue

Slides credit: Hao Su

Outline

- Transformations and homogeneous coordinates
- Rotation and SO(n)
- 3D Rotation representation
  - ***Euler Angles***
  - Axis-Angle
  - Quaternion

Euler Angle is Very Intuitive

Euler Angle to Rotation Matrix

- Rotation about principal axis is represented as:

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

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

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

- $R = R_z(\gamma)R_y(\beta)R_x(\alpha)$ for arbitrary rotation

Slides credit: Hao Su

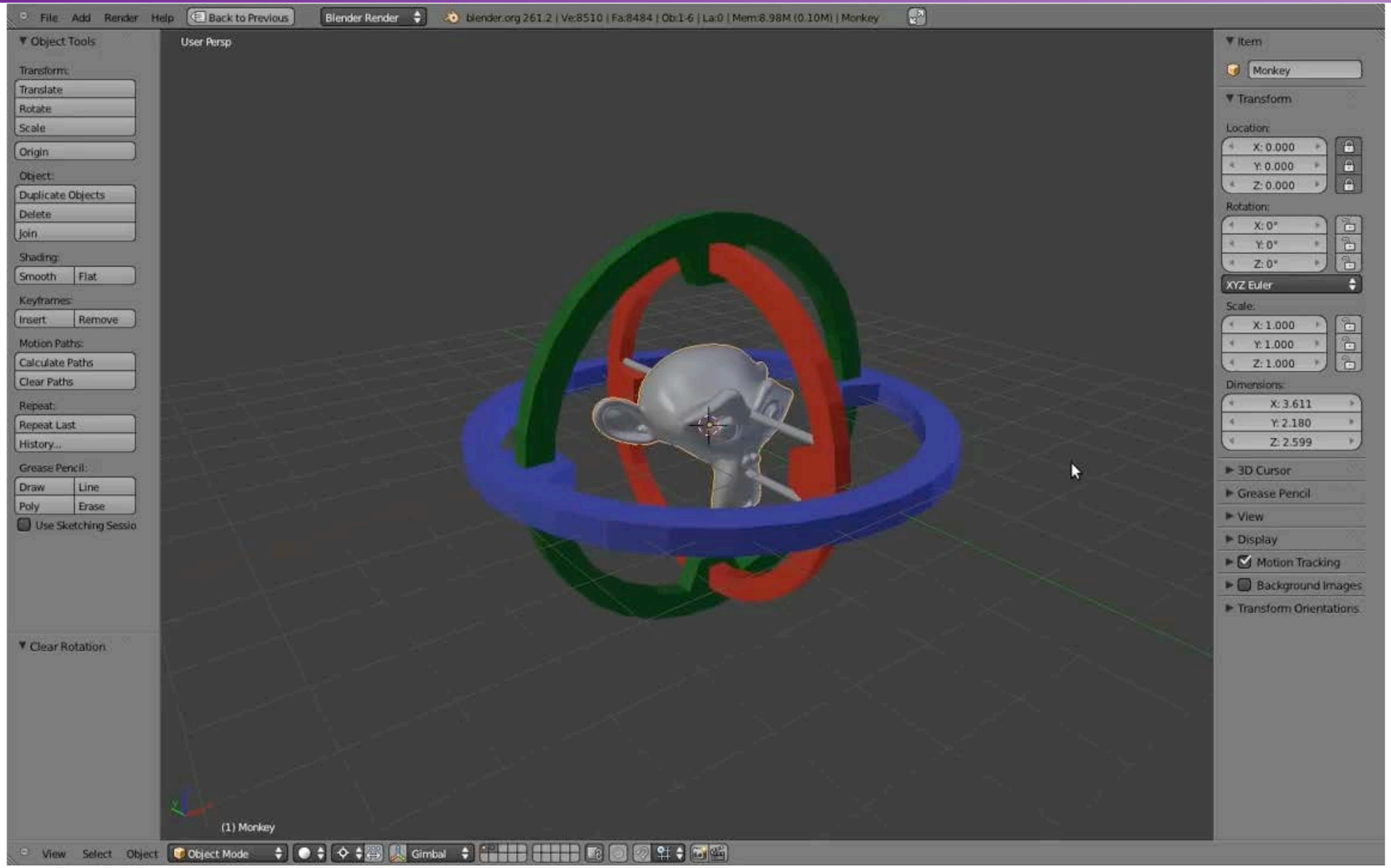
Gimbal Lock

- Euler Angle is **not unique** for some rotations. For example,

$$R_z(45°)R_y(90°)R_x(45°) = R_z(90°)R_y(90°)R_x(90°)$$

$$= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

Gimbal Lock


File Add Render Help Back to Previous Blender Render blender.org 261.2 | Ve:8510 | Fa:8484 | Ob:1-6 | La:0 | Mem:8.98M (0.10M) | Monkey
Object Tools
Transform:
Translate
Rotate
Scale
Origin
Object:
Duplicate Objects
Delete
Join
Shading:
Smooth Flat
Keyframes:
Insert Remove
Motion Paths:
Calculate Paths
Clear Paths
Repeat:
Repeat Last
History...
Grease Pencil:
Draw Line
Poly Erase
Use Sketching Sessio
Clear Rotation
User Persp
(1) Monkey
Item
Monkey
Transform
Location:
X: 0.000
Y: 0.000
Z: 0.000
Rotation:
X: 0°
Y: 0°
Z: 0°
XYZ Euler
Scale:
X: 1.000
Y: 1.000
Z: 1.000
Dimensions:
X: 3.611
Y: 2.180
Z: 2.599
3D Cursor
Grease Pencil
View
Display
Motion Tracking
Background Images
Transform Orientations
View Select Object Object Mode Gimbal

Gimbal Lock

- For example: When $\beta = \pi/2$,

$$R = R_z(\gamma)R_y(\beta)R_x(\alpha)$$

$$= \begin{bmatrix} 0 & 0 & 1 \\ \sin(\alpha+\gamma) & \cos(\alpha+\gamma) & 0 \\ -\cos(\alpha+\gamma) & \sin(\alpha+\gamma) & 0 \end{bmatrix}$$

since changing α and γ has the same effects, a degree of freedom disappears!

Summary

- Euler angle can parameterize every rotation and has good interpretability
- It is not a unique representation at some points
- There are some points where not every change in the target space (rotations) can be realized by a change in the source space (Euler angles)

Outline

- Transformations and homogeneous coordinates
- Rotation and SO(n)
- 3D Rotation representation
  - Euler Angles
  - ***Axis-Angle***
  - Quaternion

Euler Theorem

- Any rotation in $SO(3)$ is equivalent to rotation about a fixed axis $\omega \in \mathbb{R}^3$ through a positive angle θ
- $\hat{\omega}$: unit vector of rotation axis ($\|\hat{\omega}\| = 1$)
- θ: angle of rotation
- $R \in SO(3) := Rot(\hat{\omega}, \theta)$

Slides credit: Hao Su

Euler Theorem

- Any rotation in $SO(3)$ is equivalent to rotation about a fixed axis $\omega \in \mathbb{R}^3$ through a positive angle θ
- $\hat{\omega}$: unit vector of rotation axis ($\|\hat{\omega}\| = 1$)
- θ: angle of rotation
- $R \in SO(3) := Rot(\hat{\omega}, \theta)$

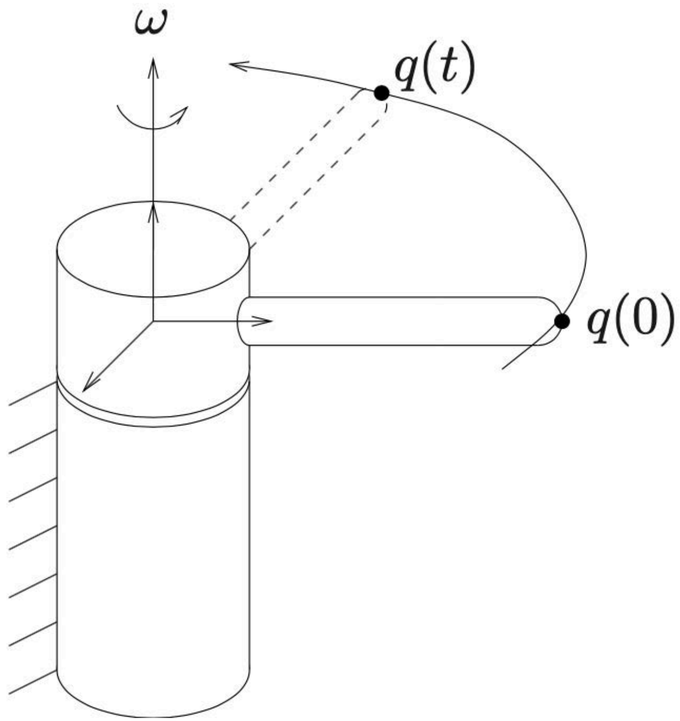
Given $\hat{\omega}$ and θ, what is $R \in SO(3)$?

Slides credit: Hao Su

Given $\hat{\omega}$ and θ, what is $R \in SO(3)$?

- Consider a point q. At time $t = 0$, the position is q_0

- Rotate q with **unit** angular velocity around axis $\hat{\omega}$, i.e.,
  - $v = \hat{\omega} \times r$
  - $\dot{q}(t) = \hat{\omega} \times q(t) = [\hat{\omega}]q(t)$



Slides credit: Hao Su

Skew-Symmetric Matrix

- A is skew-symmetric $A = -A^T$

- Skew-symmetric matrix operator:

$$\omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}, \quad [\omega] := \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

- Cross product can be a linear transformation
  - $a \times b = [a]b$

Slides credit: Hao Su

Given $\hat{\omega}$ and θ, what is $R \in SO(3)$?

$$\dot{q}(t) = \hat{\omega} \times q(t) = [\hat{\omega}]q(t)$$
$$\Rightarrow q(t) = e^{[\hat{\omega}]t} q_0 \text{ (solution of the ODE)}$$

$$\|\hat{\omega}\| = 1$$
$$\Rightarrow \text{the swept angle } \theta = \|\hat{\omega}t\| = t$$
$$\Rightarrow q(\theta) = e^{[\hat{\omega}]\theta} q_0$$
$$\Rightarrow Rot(\hat{\omega}, \theta) = e^{[\hat{\omega}]\theta} = e^{[\hat{\omega}\theta]} \text{ (exponential map)}$$

- $\vec{\omega} = \hat{\omega}\theta$ is also called **rotation vector** or **exponential coordinate**

Slides credit: Hao Su

Given $\hat{\omega}$ and θ, what is $R \in SO(3)$?

- Definition of Matrix Exponential:

$$e^{[\hat{\omega}]\theta} = I + \theta[\hat{\omega}] + \frac{\theta^2}{2!}[\hat{\omega}]^2 + \frac{\theta^3}{3!}[\hat{\omega}]^3 + \cdots$$

- Sum of infinite series? **Rodrigues Formula**
  - Can prove that $[\hat{\omega}]^3 = -[\hat{\omega}]$
  - Then, use Taylor expansion of **sin** and **cos**
  - $e^{[\hat{\omega}]\theta} = I + [\hat{\omega}]\sin\theta + [\hat{\omega}]^2(1 - \cos\theta)$

Slides credit: Hao Su

Given $\hat{\omega}$ and θ, what is $R \in SO(3)$?

- First question: Is there a **unique** parametrization?
  - No:
    1. $(\hat{\omega}, \theta)$ and $(-\hat{\omega}, -\theta)$ give the same rotation
    2. when $R = I$, $\theta = 0$ and $\hat{\omega}$ can be arbitrary

- When 2 does not happen, and if we also restrict $\theta \in (0, \pi]$, a unique parameterization exists:
  - when $\mathrm{tr}(R) \neq -1$, can be computed by

$$\theta = \arccos \frac{1}{2}[\mathrm{tr}(R) - 1], \quad [\hat{\omega}] = \frac{1}{2\sin\theta}(R - R^T)$$

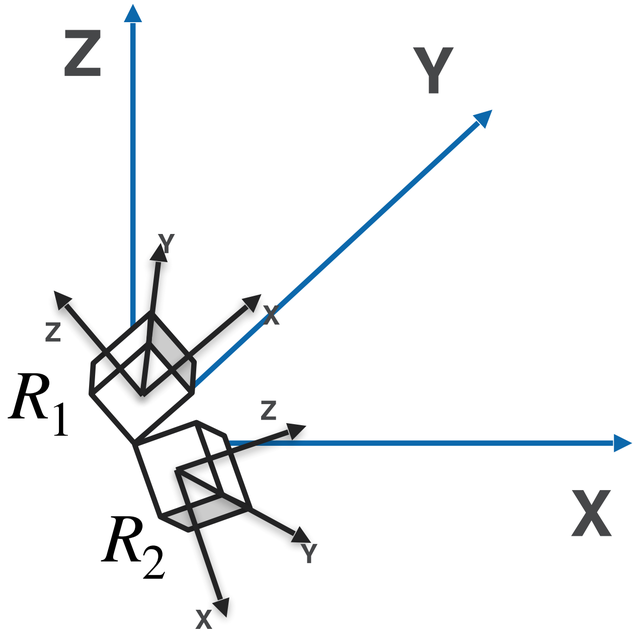
  - when $\mathrm{tr}(R) = -1$, they are the cases that $\theta = \pi$ for rotations around x/y/z axis

Slides credit: Hao Su

Distance between Rotations

- How to measure the distance between rotations (R_1, R_2)?
- A natural view is to measure the (minimal) effort to rotate the body at R_1 pose to R_2 pose:

$$\because (R_2 R_1^T) R_1 = R_2 \quad \therefore \text{dist}(R_1, R_2) = \theta(R_2 R_1^T) = \arccos \frac{1}{2}[\text{tr}(R_2 R_1^T) - 1]$$



Slides credit: Hao Su

Inspection from Learning Perspective

- When used in networks, one prominent issue is:
  - Suppose that you are estimating $\theta\hat{\omega}$ as a 3D vector
  - To keep a unique parameterization, you assume that $\theta \in (0,\pi]$
  - Your current solution is $\pi\hat{\omega}$
  - $(\pi - \epsilon)(-\hat{\omega})$ is mapped to a neighborhood point in $SO(3)$, but it is not in the neighborhood of the domain, hence gradient descent could not achieve it

Slides credit: Hao Su

Summary of Axis-Angle

- Axis-Angle is an intuitive rotation representation
- By adding a constraint to the domain of θ , the parameterization can be unique at most points
- Can be converted to and from rotation matrices by exponential map and its inverse (when possible)
- Induced a distance between rotations which is a metric in $\mathrm{SO}(3)$ (independent of parameterization)

Outline

- Transformations and homogeneous coordinates
- Rotation and SO(n)
- 3D Rotation representation
  - Euler Angles
  - Axis-Angle
  - ***Quaternion***

Mathematical Definition

- Recall the complex number $a + b\mathbf{i}$

- Quaternion is a more generalized complex number:

  $$q = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$$

  - w is the real part and $\overrightarrow{v} = (x, y, z)$ is the imaginary part
  - Imaginary: $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$
  - anti-commutative :
    $\mathbf{ij} = \mathbf{k} = -\mathbf{ji},\ \mathbf{jk} = \mathbf{i} = -\mathbf{kj},\ \mathbf{ki} = \mathbf{j} = -\mathbf{ik}$

Slides credit: Hao Su

Properties of General Quaternions

- In vector-form, the product of two quaternions:
  For $q_1 = (w_1, \vec{v}_1)$ and $q_2 = (w_2, \vec{v}_2)$

$$q_1 q_2 = (w_1 w_2 - \vec{v}_1^T \vec{v}_2, w_1 \vec{v}_2 + w_2 \vec{v}_1 + \vec{v}_1 \times \vec{v}_2)$$

- Conjugate: $q^* = (w, -\vec{v})$
- Norm: $\|q\|^2 = w^2 + \vec{v}^T \vec{v} = qq^* = q^* q$
- Inverse: $q^{-1} := \dfrac{q^*}{\|q\|^2}$

Slides credit: Hao Su

Unit Quaternion as Rotation

- A **unit** quaternion $\|q\| = 1$ can represent a rotation
  - Four numbers plus one constraint $\rightarrow$ 3 DoF

- Geometrically, the shell of a 4D sphere

Slides credit: Hao Su

Unit Quaternion as Rotation

- Rotate a vector $\overrightarrow{x}$ by quaternion q:
  1. Augment $\overrightarrow{x}$ to $x = (0, \overrightarrow{x})$
  2. $x' = qxq^{-1}$
- Compose rotations by quaternion:
  - $(q_2(q_1 x q_1^*)q_2^*)$: first rotate by q_1 and then by q_2
  - Since $(q_2(q_1 x q_1^*)q_2^*) = (q_2 q_1)x(q_1^* q_2^*)$, we conclude that composing rotations is as simple as multiplying quaternions!

Slides credit: Hao Su

Conversation between Quaternions and Angle-Axis

- Exponential coordinate → Quaternion:

$$q = [\cos(\theta/2), \sin(\theta/2)\hat{\omega}]$$

Quaternion is very close to angle-axis representation!

- Exponential coordinate ← Quaternion:

$$\theta = 2\arccos(w), \qquad \hat{\omega} = \begin{cases} \dfrac{1}{\sin(\theta/2)}\overrightarrow{v} & \theta \neq 0 \\ 0 & \theta = 0 \end{cases}$$

Slides credit: Hao Su

Conversation between Quaternion and Rotation Matrix

- Rotation ← Quaternion

$$R(q) = E(q)G(q)^T$$

$$\text{where } E(q) = [-\vec{v}, wI + [\vec{v}]] \text{ and}$$

$$G(q) = [-\vec{v}, wI - [\vec{v}]]$$

- Rotation → Quaternion
  - Rotation → Angle-Axis → Quaternion

Slides credit: Hao Su

Inspection from Learning Perspective

- Each rotation corresponds to two quaternions (“double-covering”)
- Need to normalize to unit length in networks. This normalization may cause big/small gradients in practice

Slides credit: Hao Su

More about Quaternion

- Quaternion is computationally cheap:
  - Internal representation of Physical Engine and Robot
  - Pay attention to convention (w, x, y, z) or (x, y, z, w)?
  - (w, x, y, z): SAPIEN, transforms3d, Eigen, blender, MuJoCo, V-Rep
  - (x, y, z, w): ROS, PhysX, PyBullet

Slides credit: Hao Su

Summary of Quaternion

- Very useful and popular in practice
- 4D parameterization, compact and efficient to compute
- Good numerical properties in general

Summary of Rotation Representations

	Inverse?	Composing?	Any local movement in SO(3) can be achieved by local movement in the domain?
Rotation Matrix	✔	✔	N/A
Euler Angle	Complicated	Complicated	No
Angle-axis	✔	Complicated	?
Skew-symmetrical Matrix	✔	Complicated	?
Quaternion	✔	✔	✔

? means no singularity with single exceptions

Slides credit: Hao Su

Useful Resources

- On the Continuity of Rotation Representations in Neural Networks
- A useful torch library that you can play with is “kornia”
- Use with cautious to its numerical properties
- “ceres” is a C++ library that is quite useful

Next Time: Learning-based SFM and SLAM