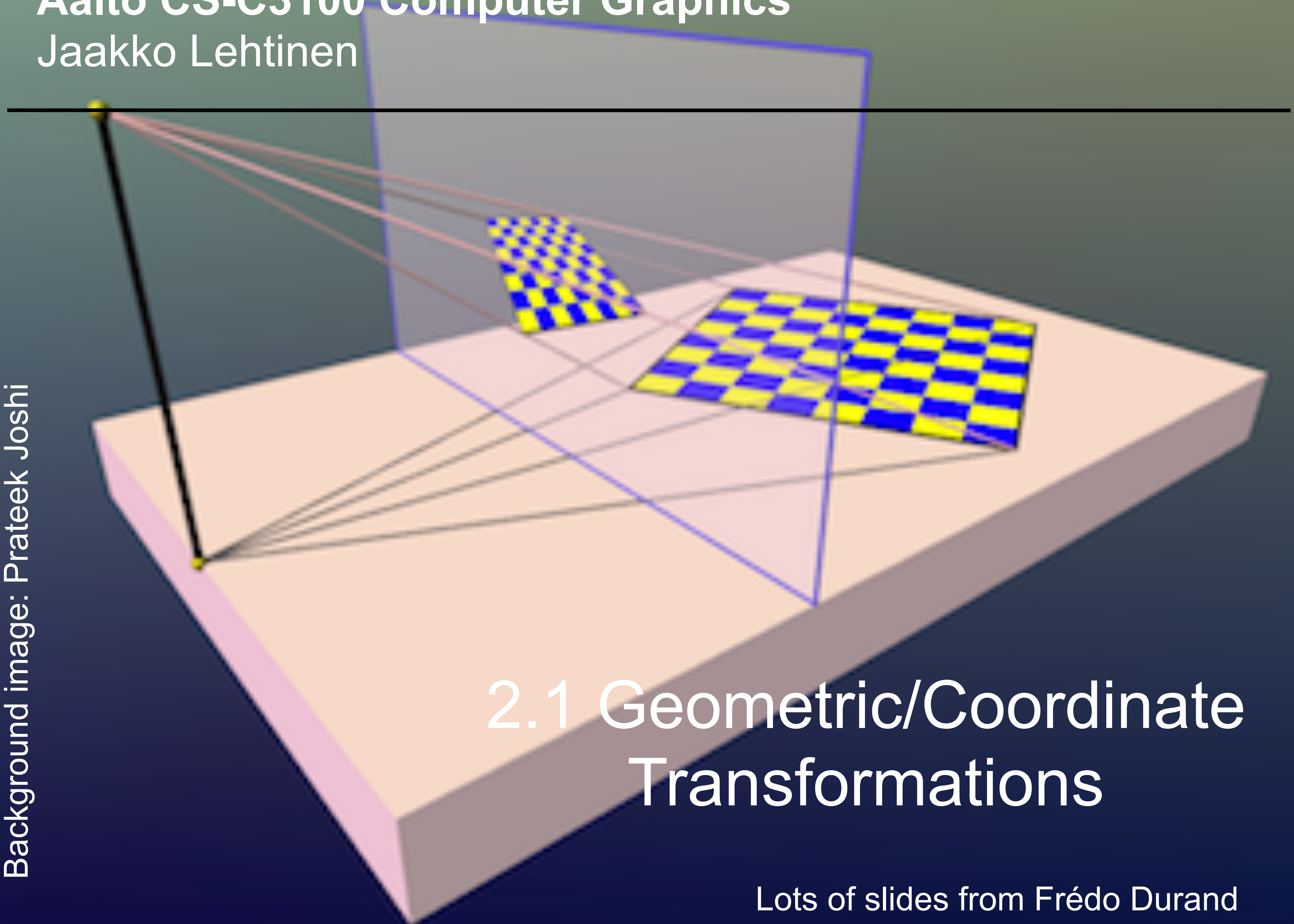
Aalto CS-C3100 Computer Graphics

Jaakko Lehtinen



2.1 Geometric/Coordinate Transformations

Lots of slides from Frédo Durand

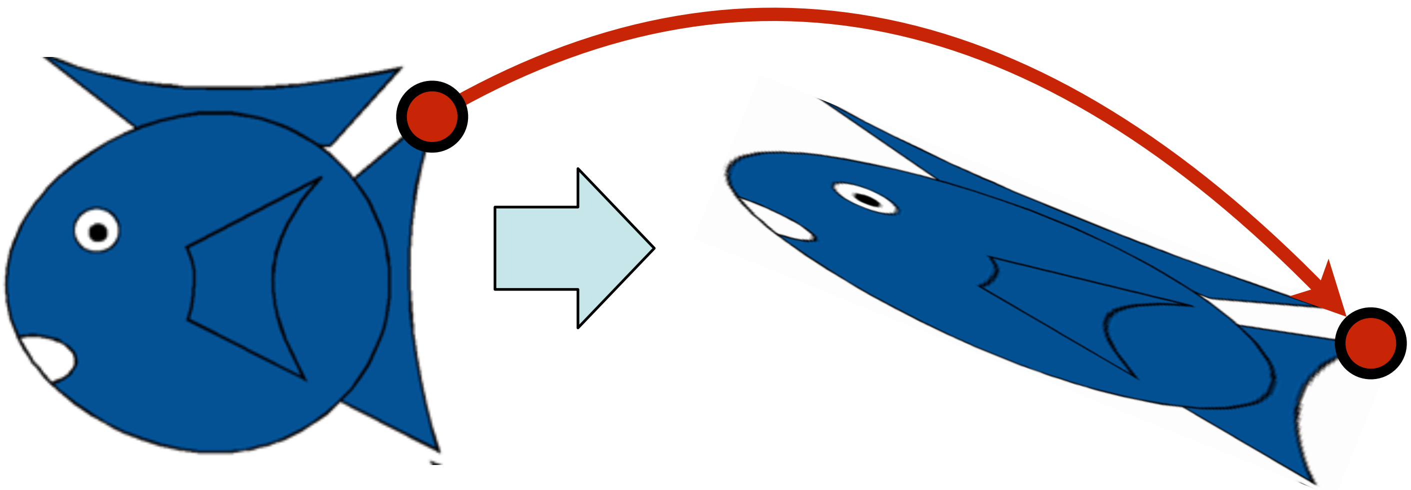
Background image: Prateek Joshi

In This Video

- What are geometric transformations?
- Useful types of transformations
- Transformations as algebraic groups
  - And why it is useful to look at them that way

Two Views on Transformations

- First, the geometric one: a warp of space
  - Focus on how a point x gets transported into x'

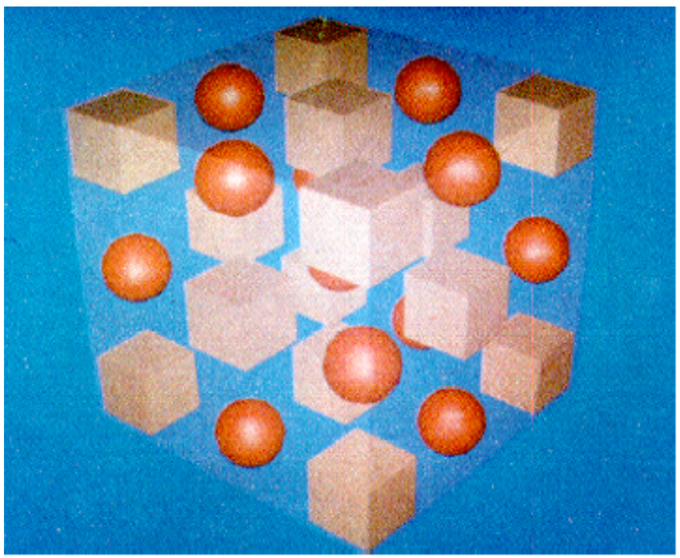


points in original configuration

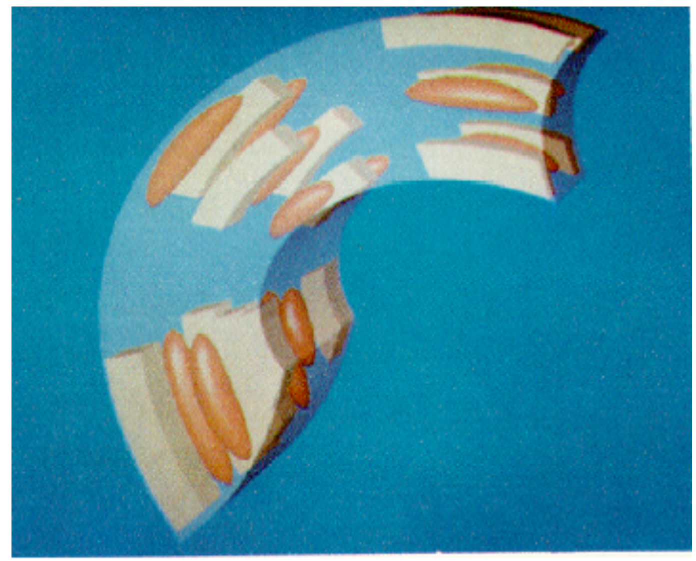
points in warped configuration after transformation

Two Views on Transformations

- First, the geometric one: a warp of space
  - Focus on how a point x gets transported into x'



points in original configuration

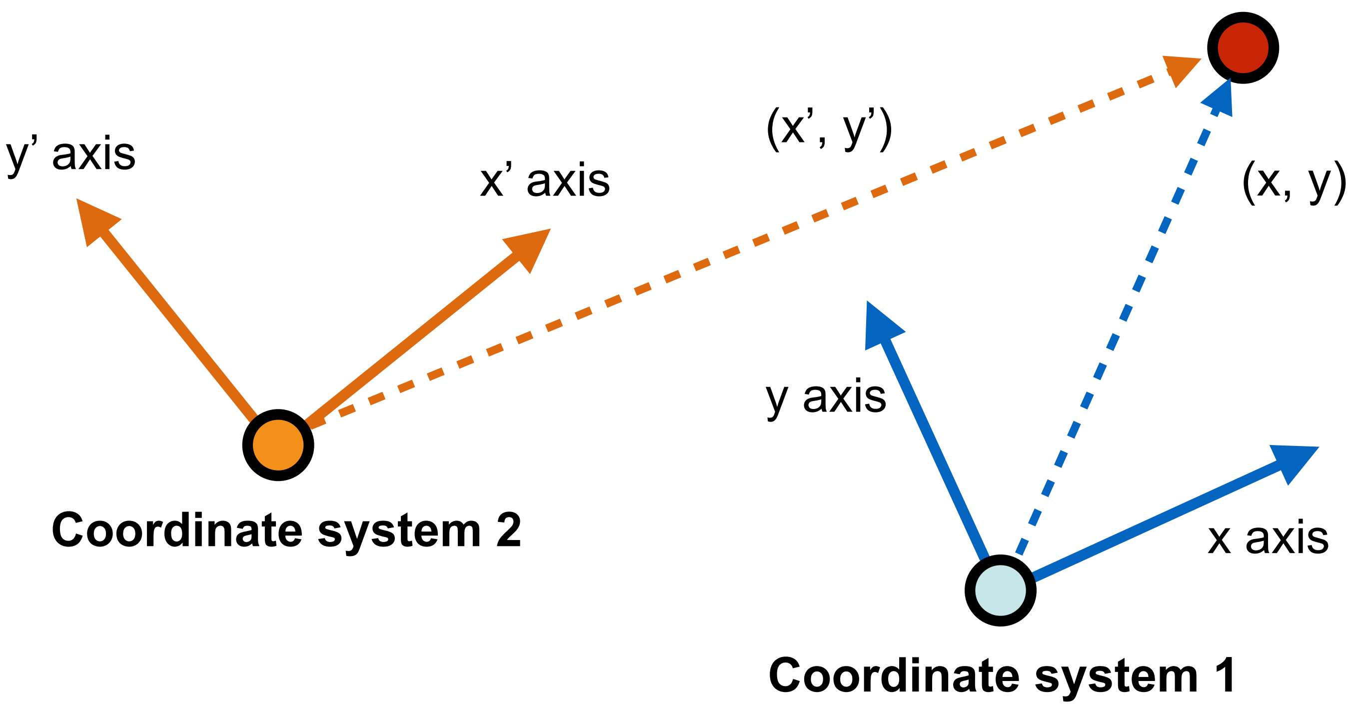


points in warped configuration after transformation

From Sederberg and Parry, Siggraph 1986 link

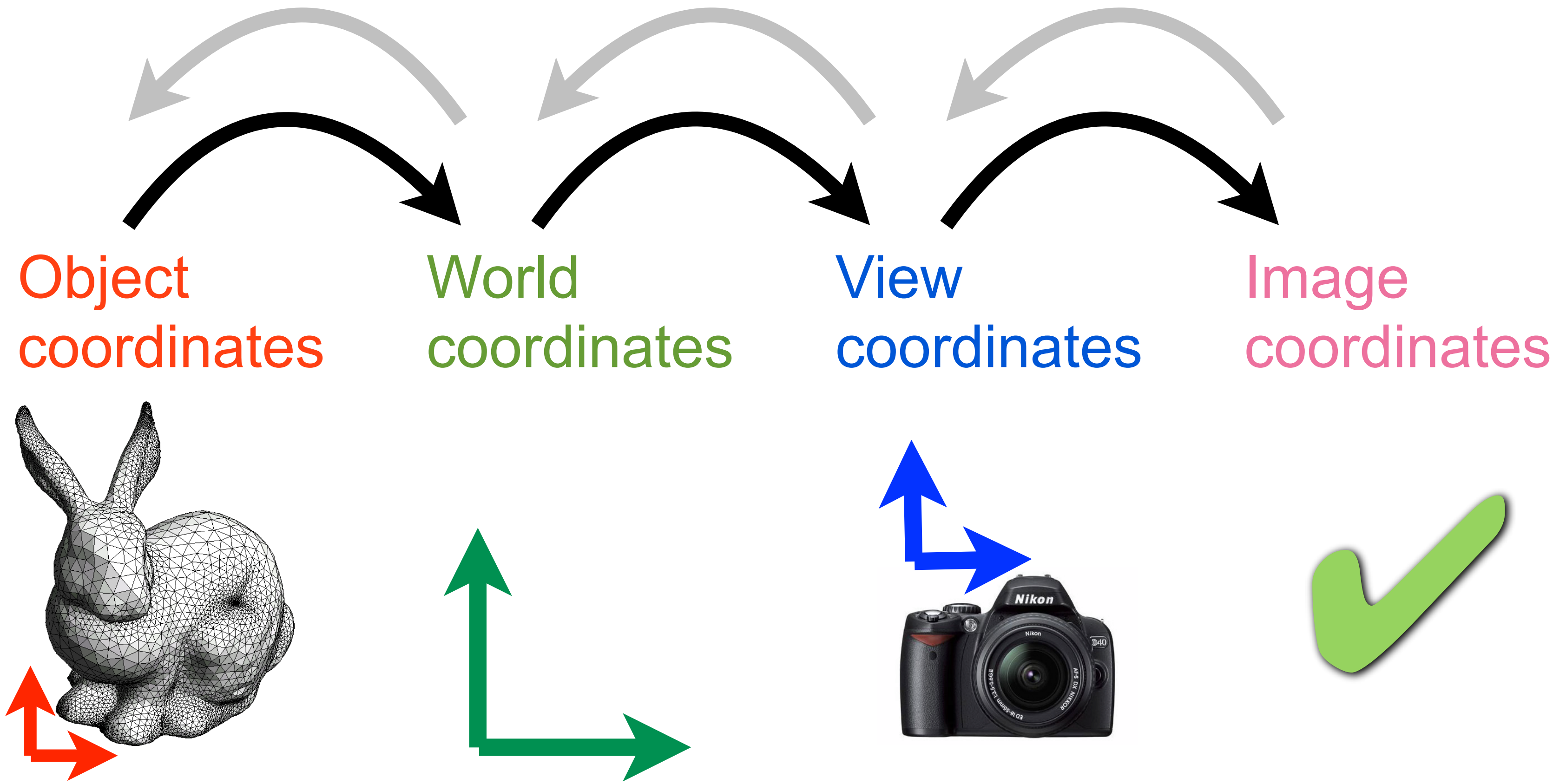
Two Views on Transformations

- Second, the coordinate view
  - Given the coordinates (x, y) of a point in System 1, what are its coordinates (x', y') in System 2?



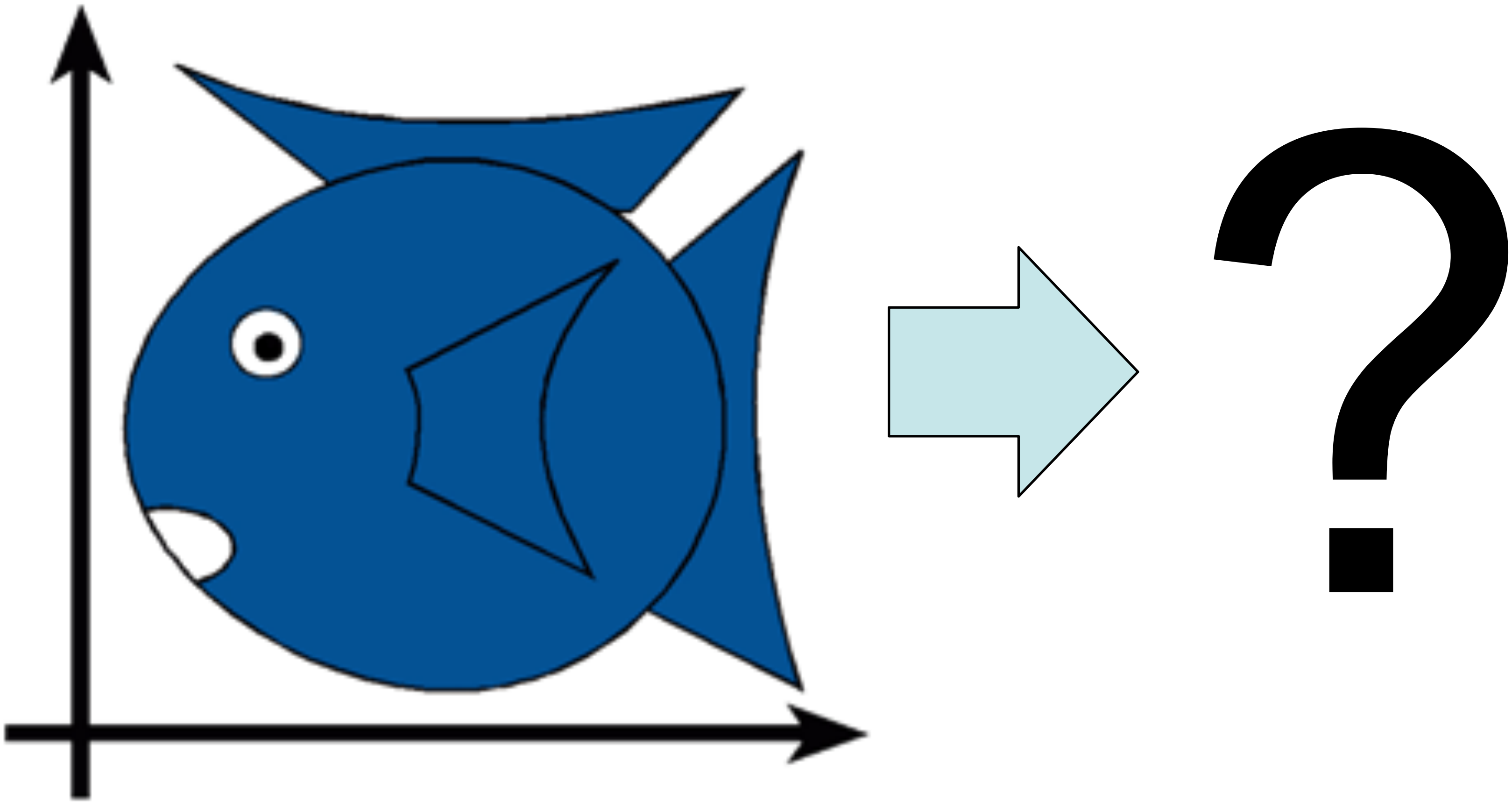
View 2 is Directly Useful Here

Transformations
take us between these coordinates.

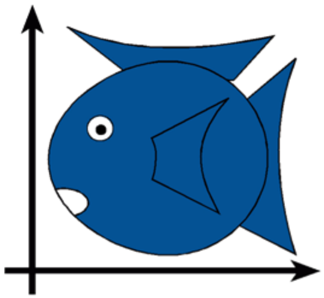


The two views are not contradictory

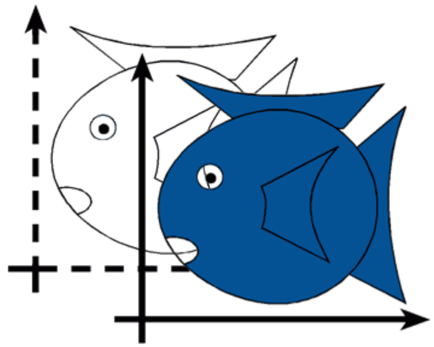
Simple Transformations



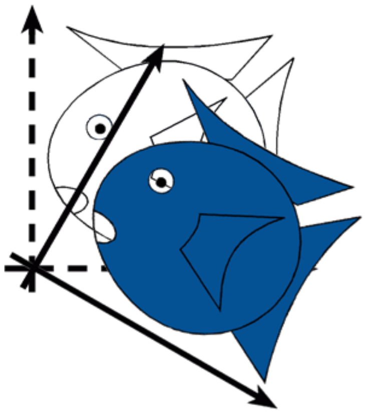
Some Simple Transformations



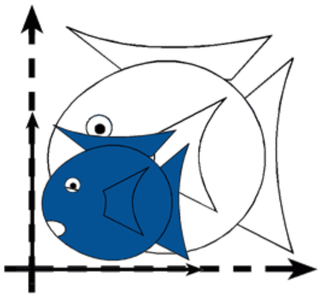
Identity



Translation



Rotation



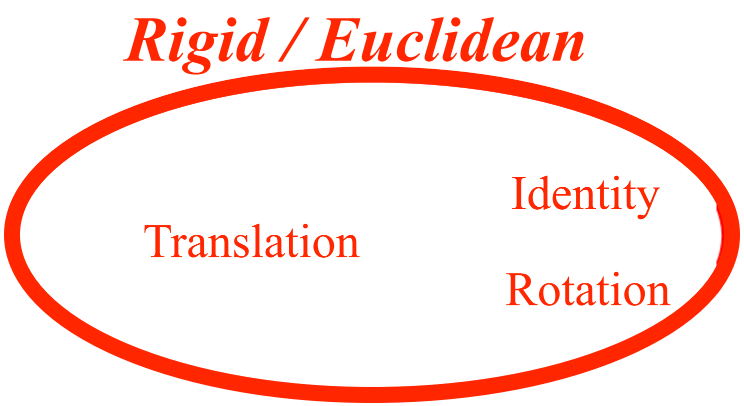
Isotropic
(Uniform)
Scaling

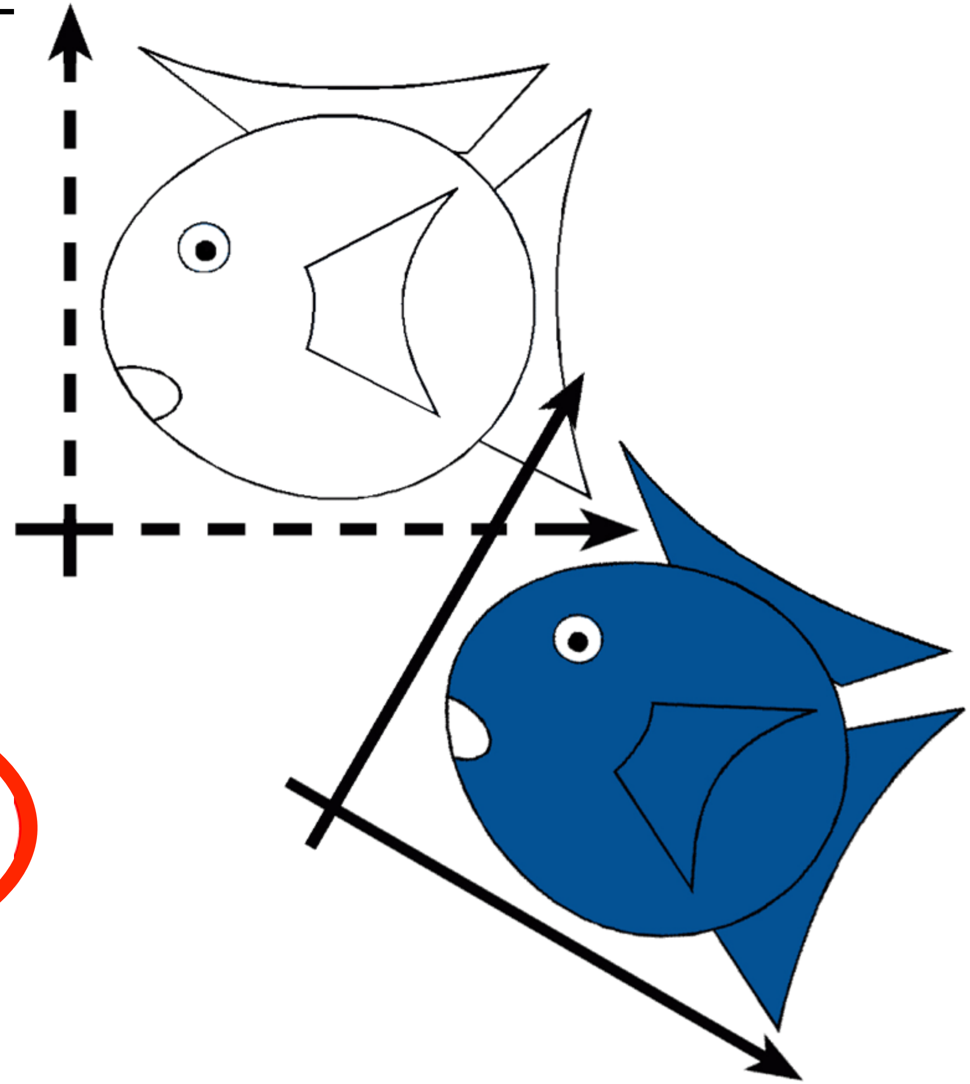
- Can be combined
- Are these operations invertible?

  Yes, except scale = 0

Rigid-Body / Euclidean Transforms

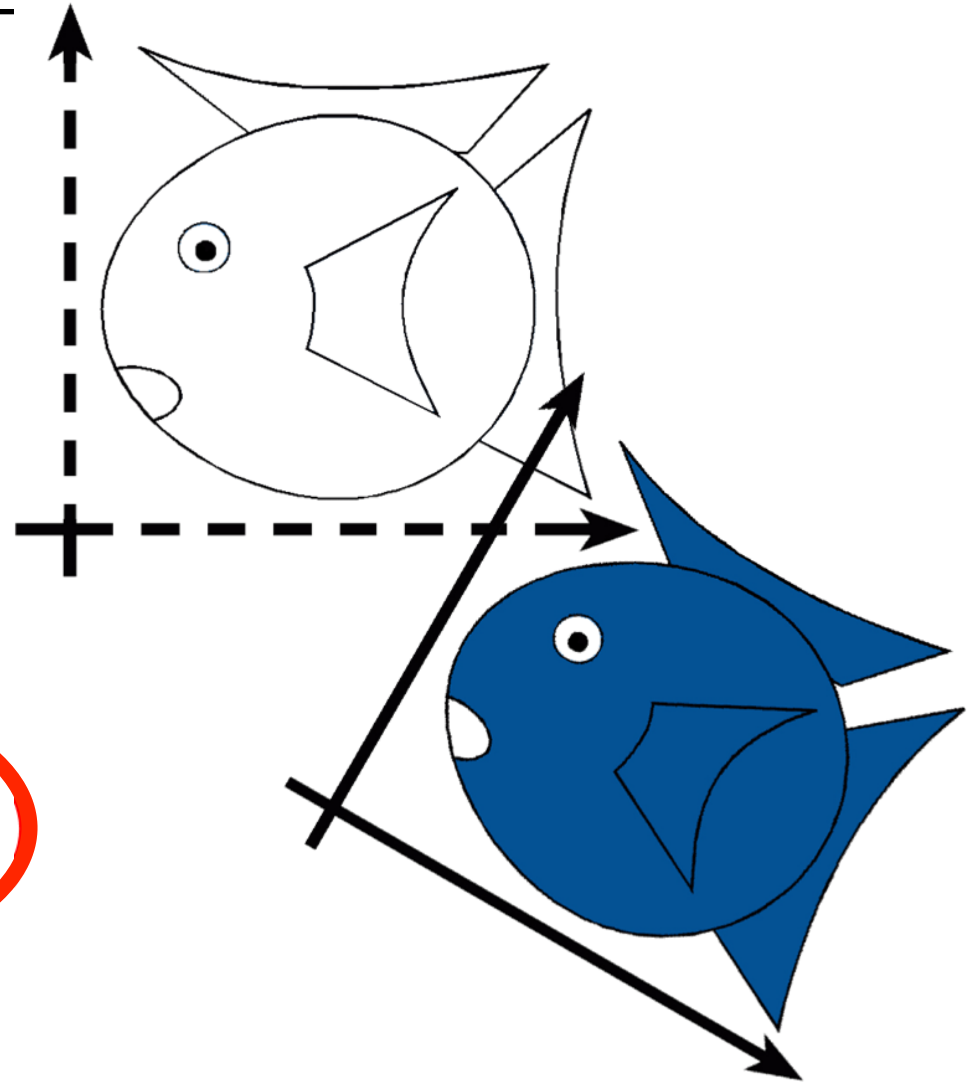
- What properties are preserved?





Rigid-Body / Euclidean Transforms

- Preserves distances
- Preserves angles



Rigid / Euclidean

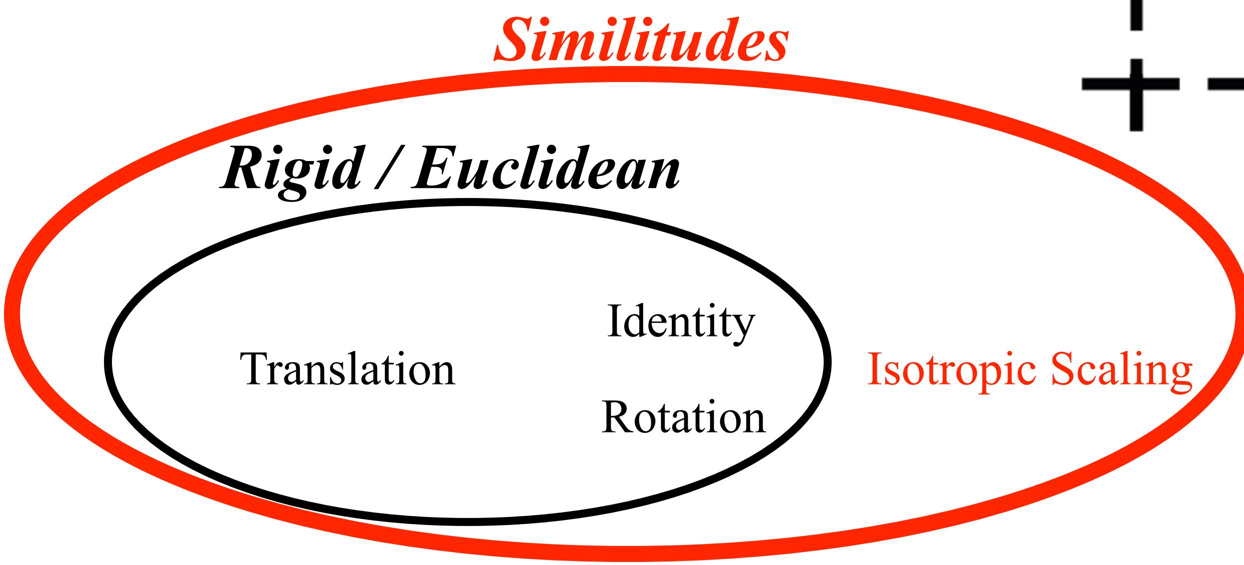
Identity

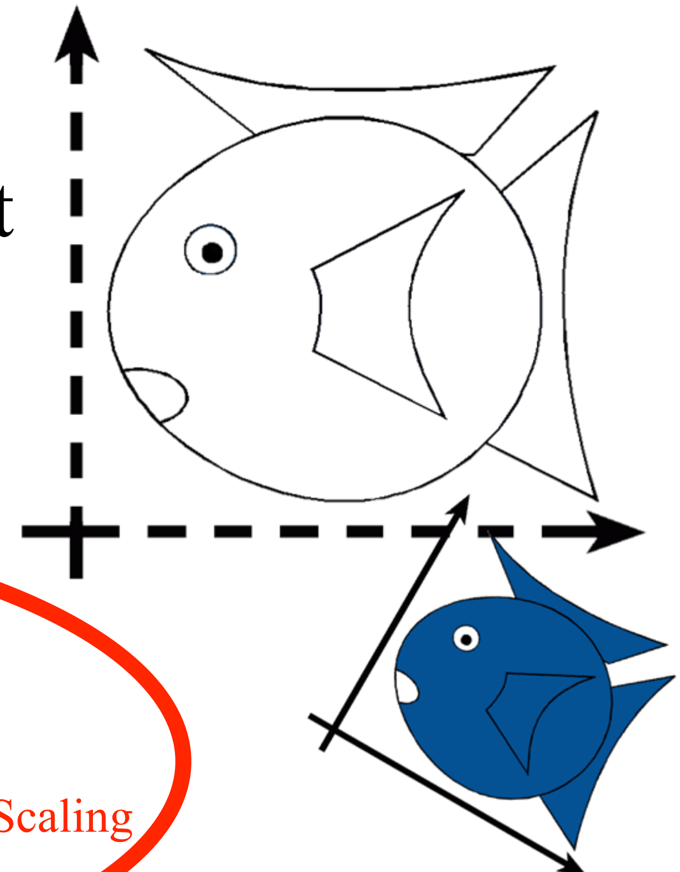
Translation

Rotation

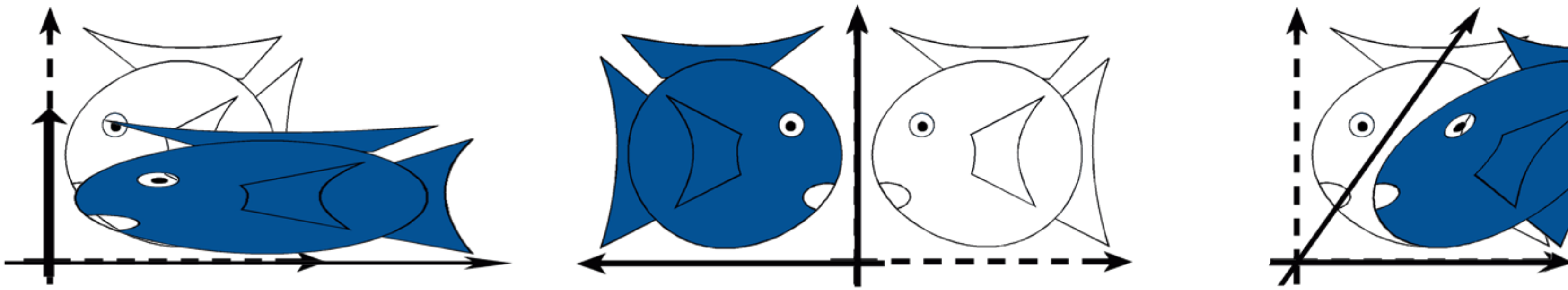
Similitudes / Similarity Transforms

- Preserves angles
- "The shapes are the same", just at different scale, orientation and location

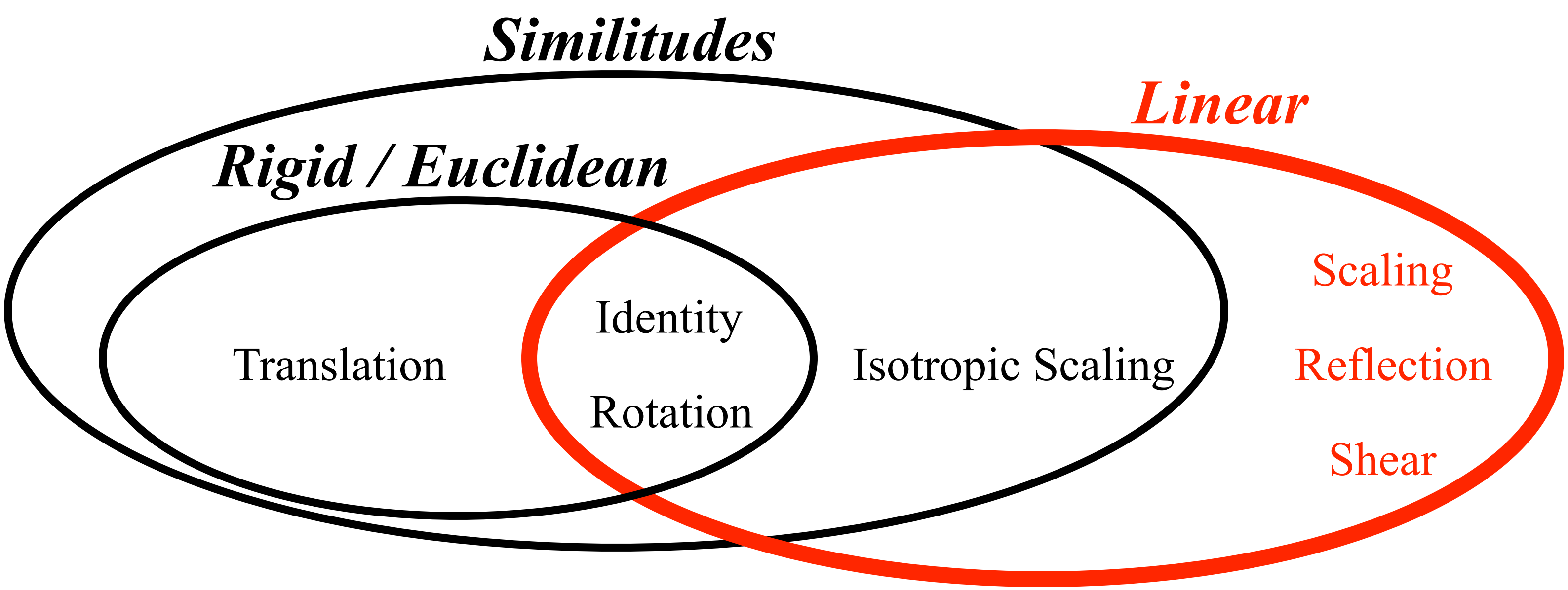




Linear Transformations

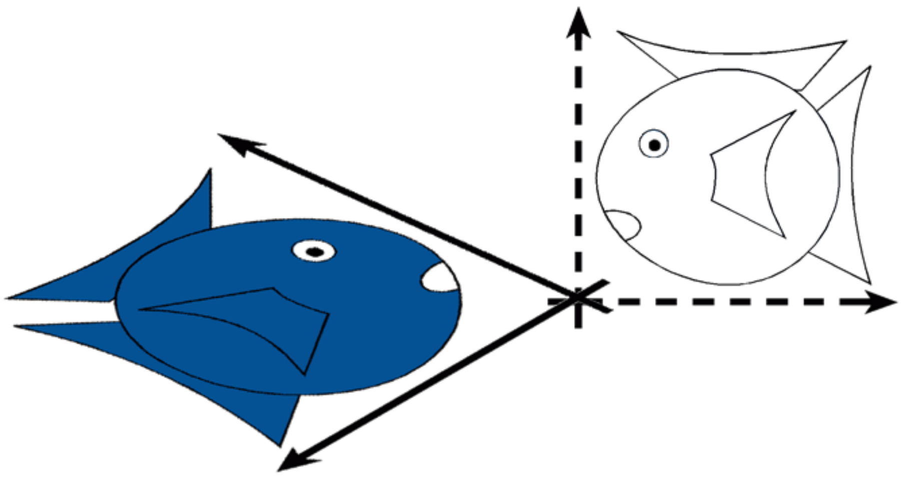


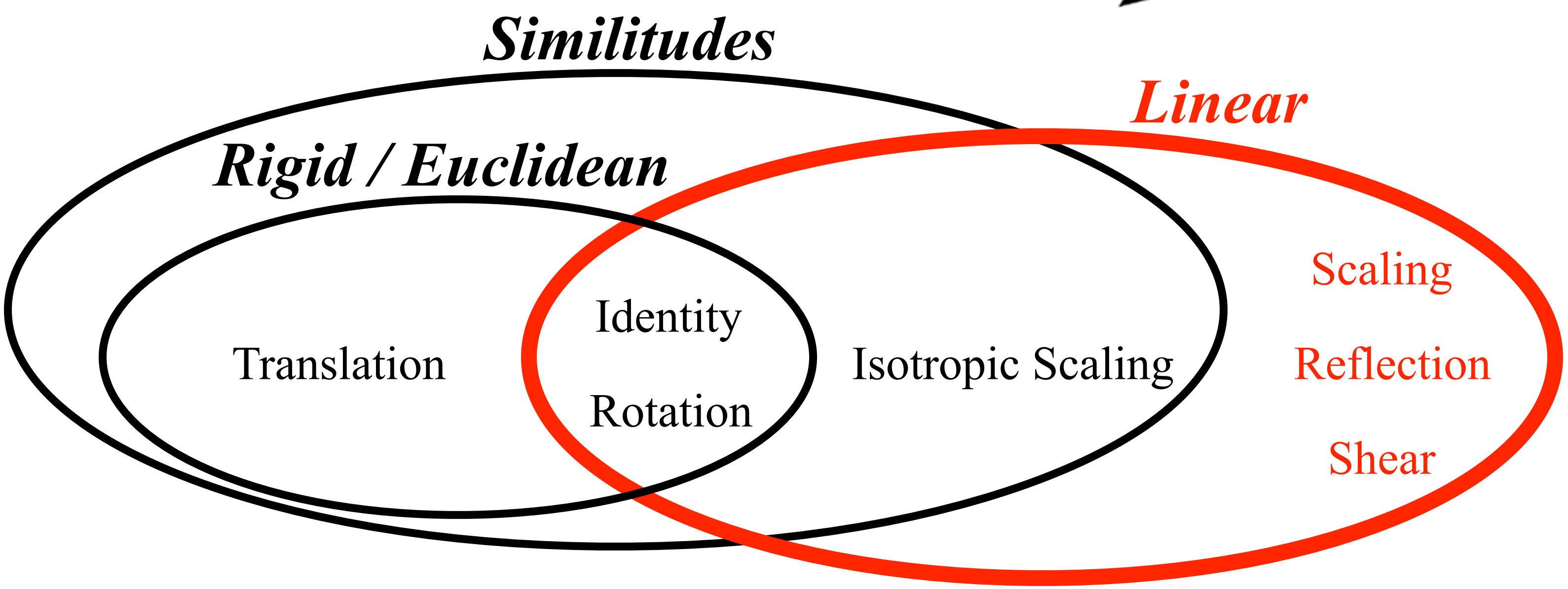
Scaling Reflection Shear



Linear Transformations

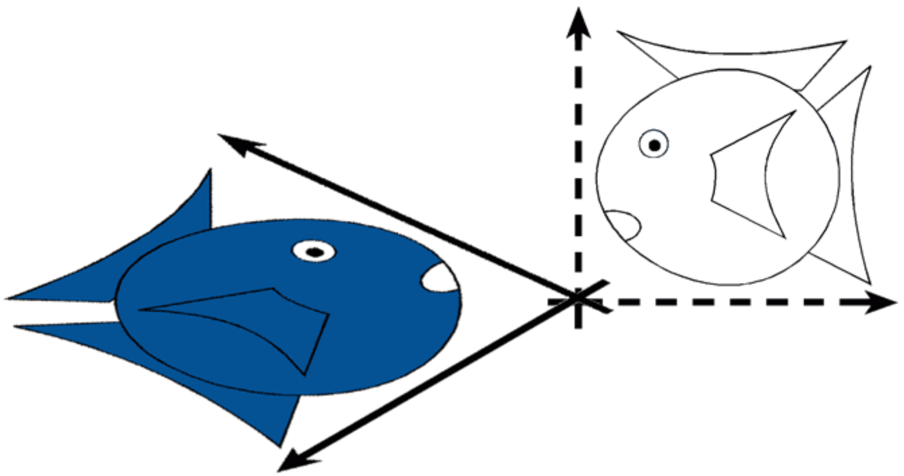
- $L(\mathbf{p} + \mathbf{q}) = L(\mathbf{p}) + L(\mathbf{q})$
- $L(a\mathbf{p}) = a\, L(\mathbf{p})$

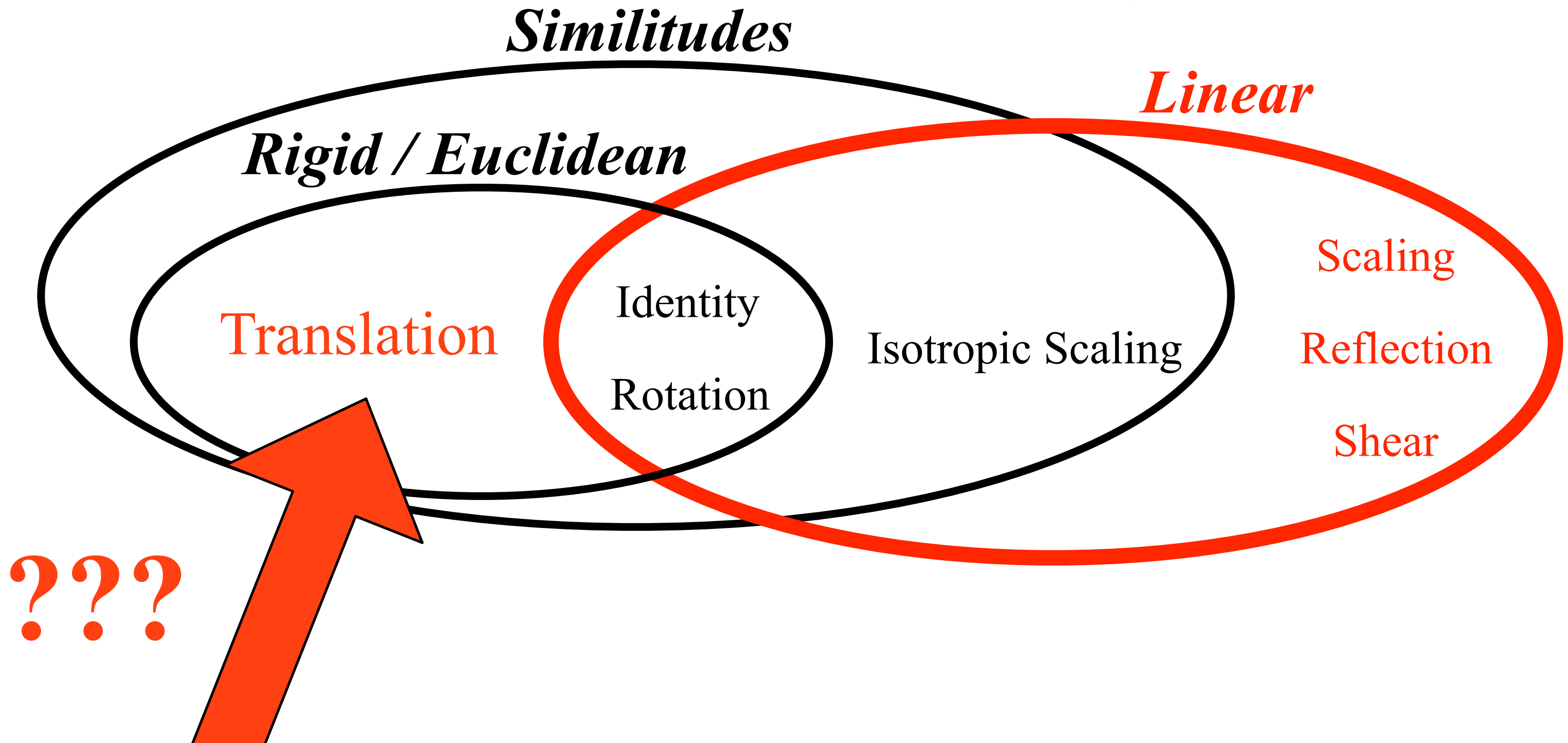




Linear Transformations

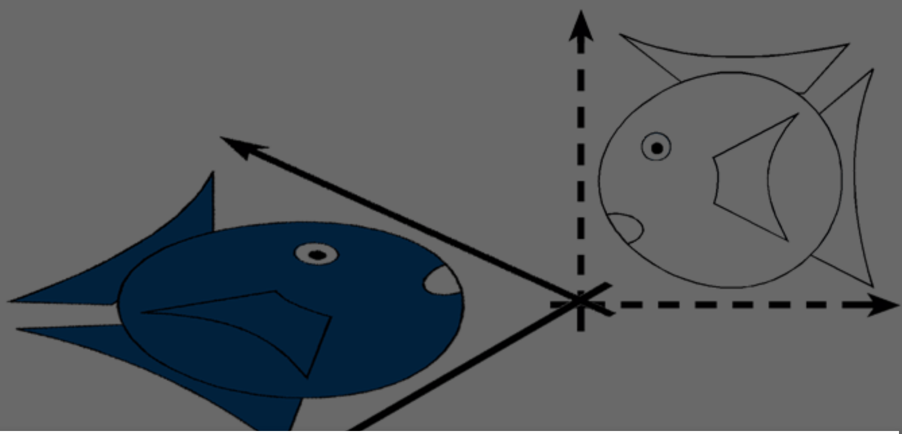
- $L(p + q) = L(p) + L(q)$
- $L(ap) = a\,L(p)$





Linear Transformations

- $L(p + q) = L(p) + L(q)$
- $L(ap) = a\,L(p)$



Translation is not linear:

$$f(\mathbf{p}) = \mathbf{p}+\mathbf{t}$$

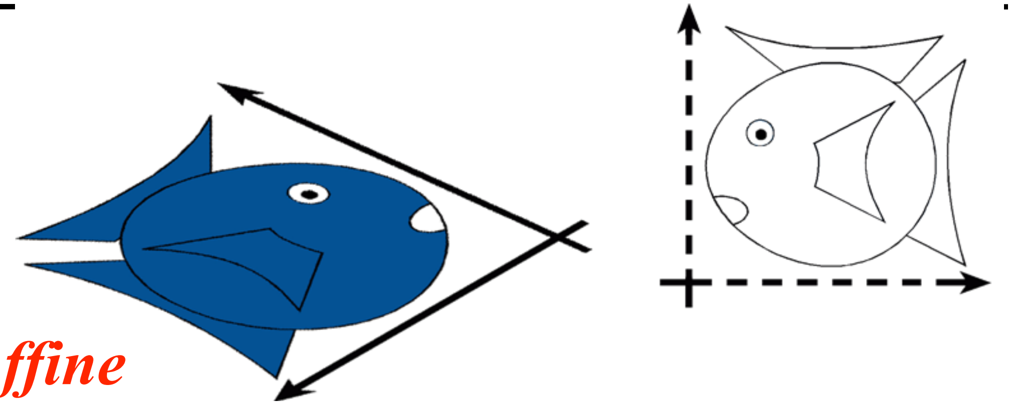
$$f(a\mathbf{p}) = a\mathbf{p}+\mathbf{t} \neq a(\mathbf{p}+\mathbf{t}) = a\,f(\mathbf{p})$$

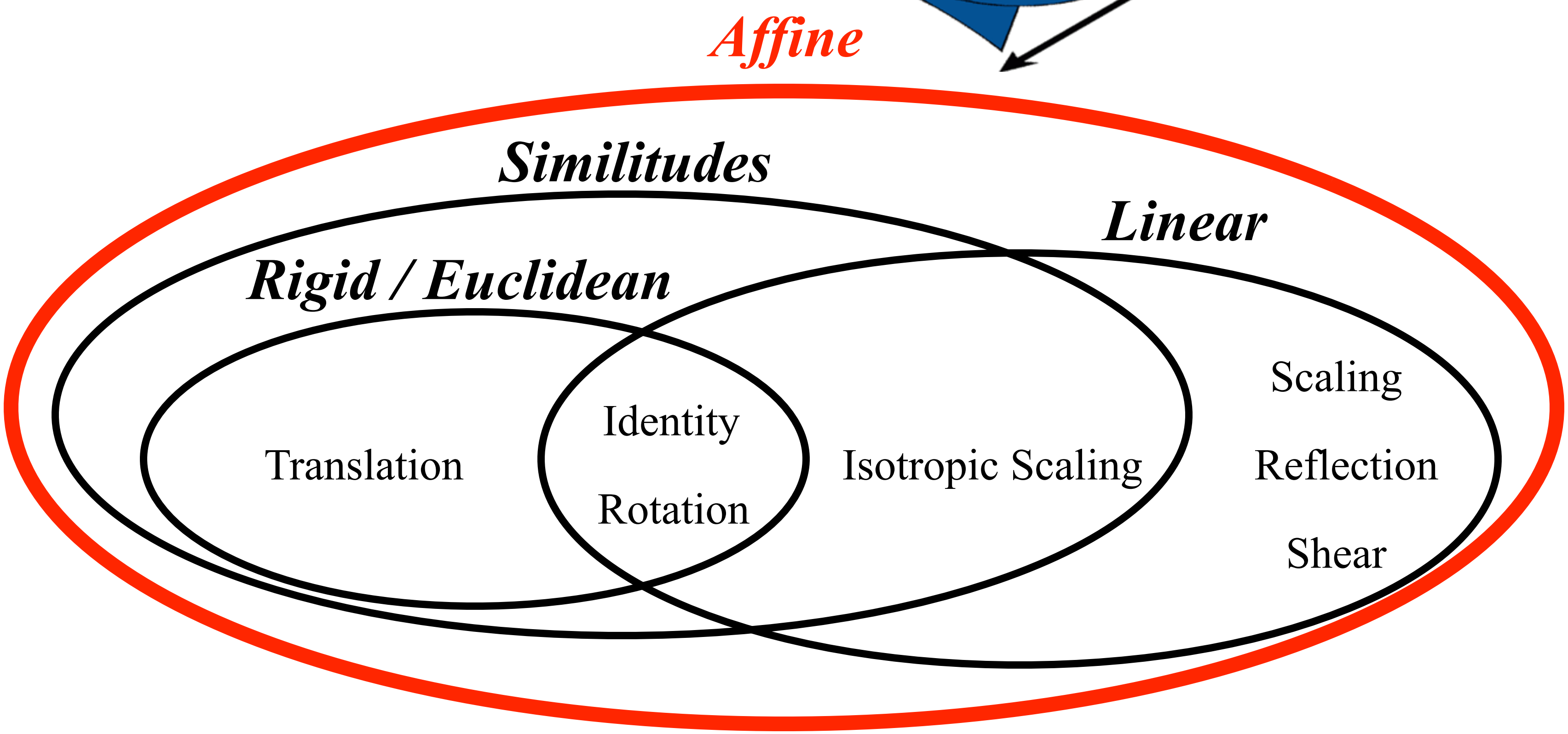
$$f(\mathbf{p}+\mathbf{q}) = \mathbf{p}+\mathbf{q}+\mathbf{t} \neq (\mathbf{p}+\mathbf{t})+(\mathbf{q}+\mathbf{t}) = f(\mathbf{p}) + f(\mathbf{q})$$

???

Affine Transformations

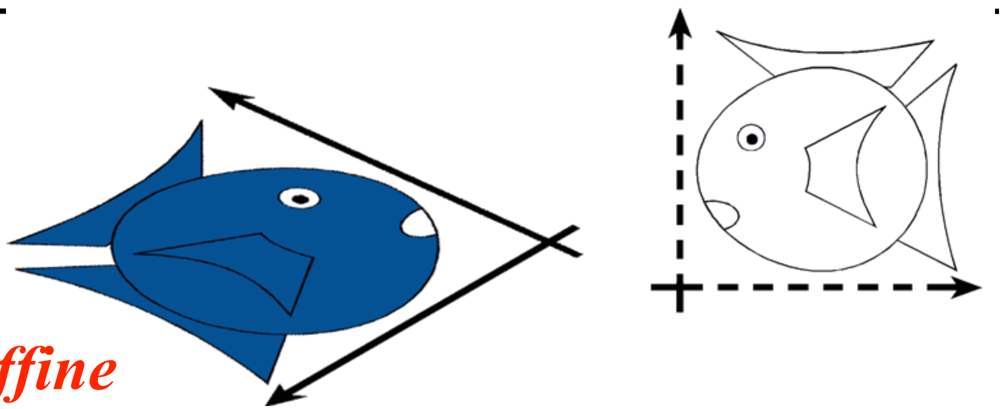
- What is preserved..?

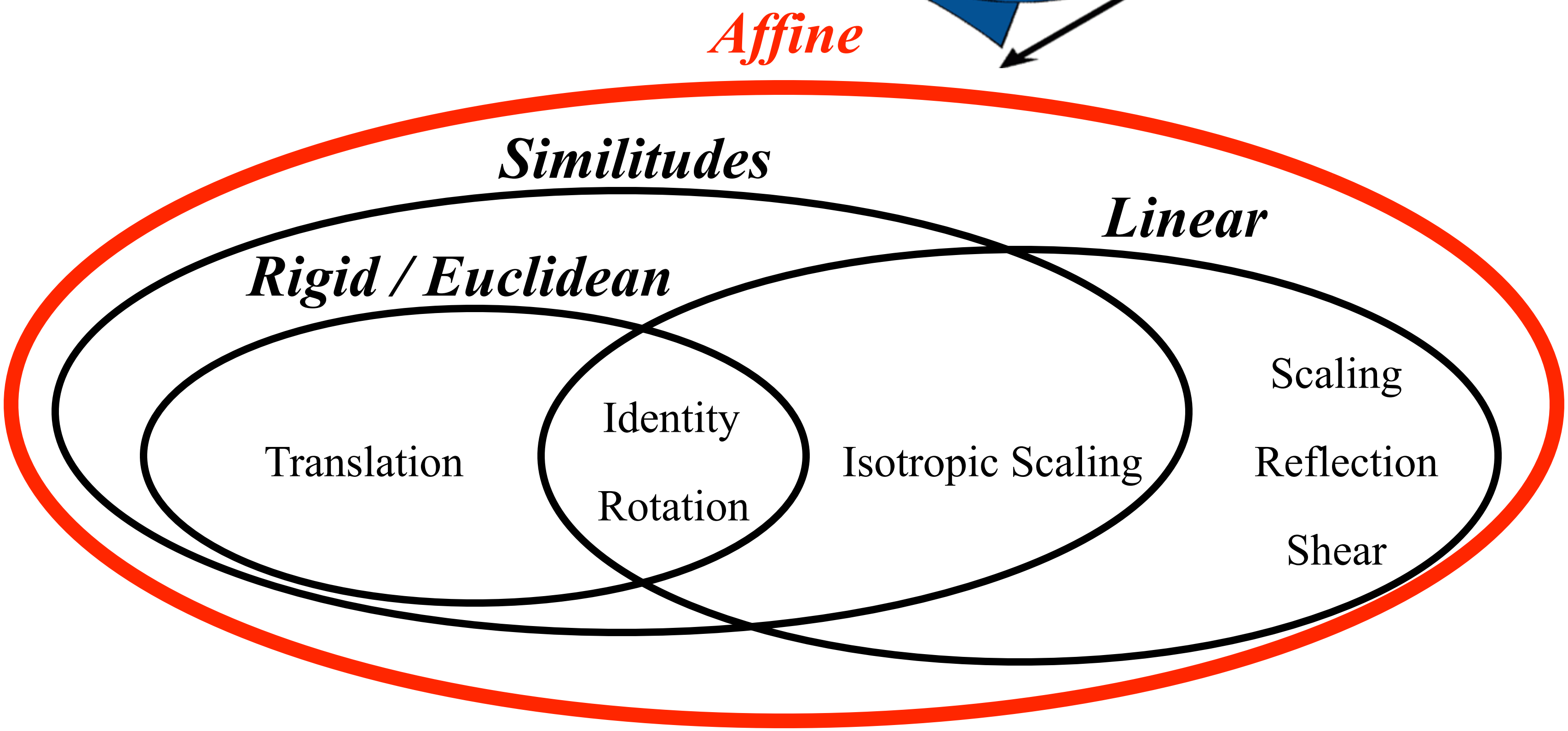




Affine Transformations

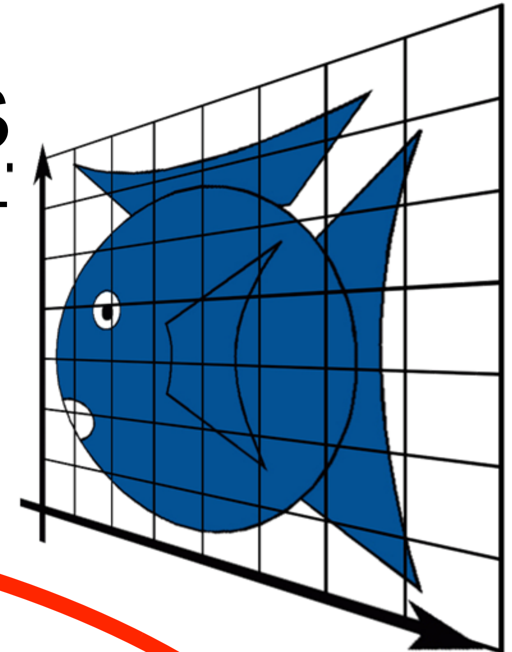
- Preserves parallel lines

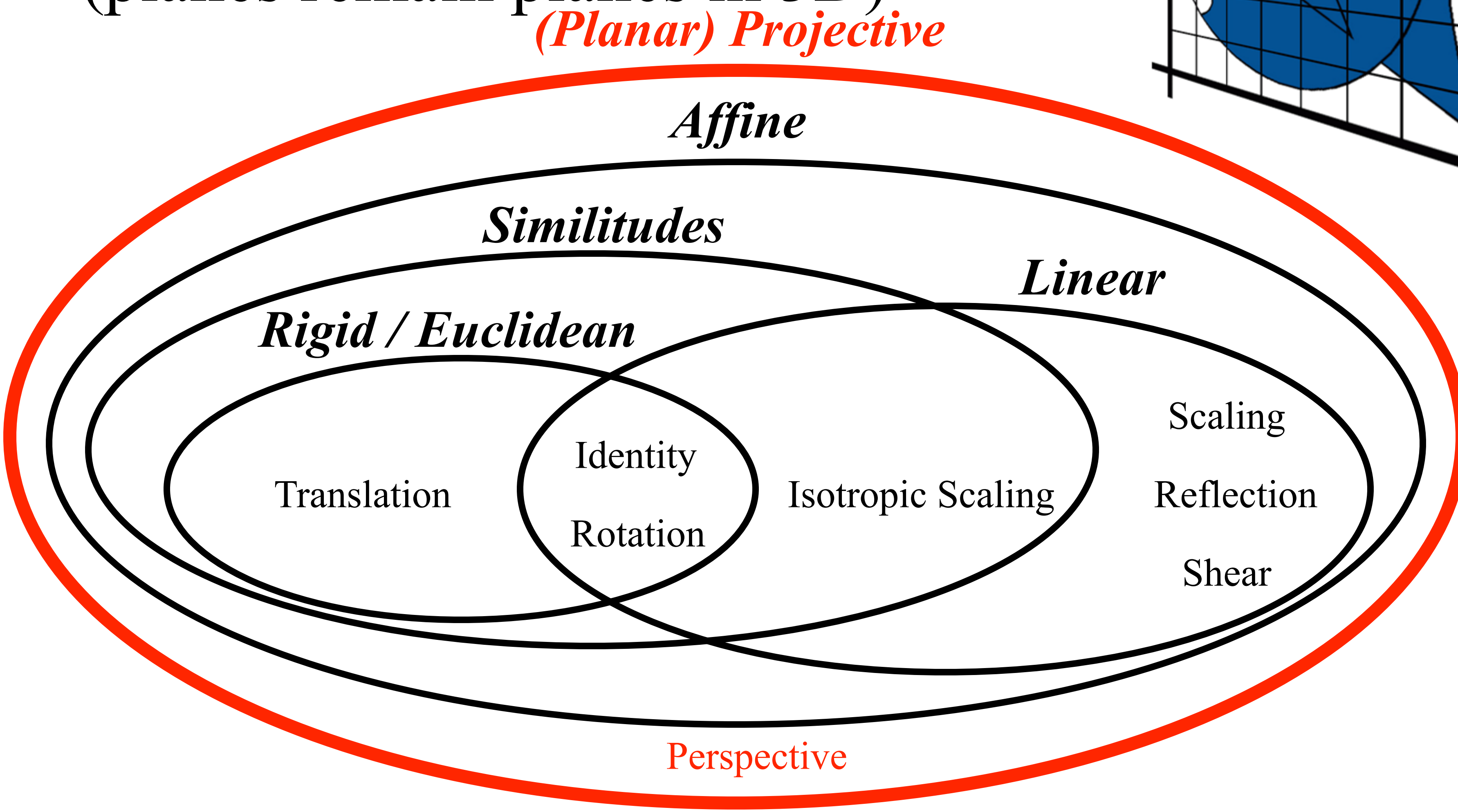




Projective Transformations

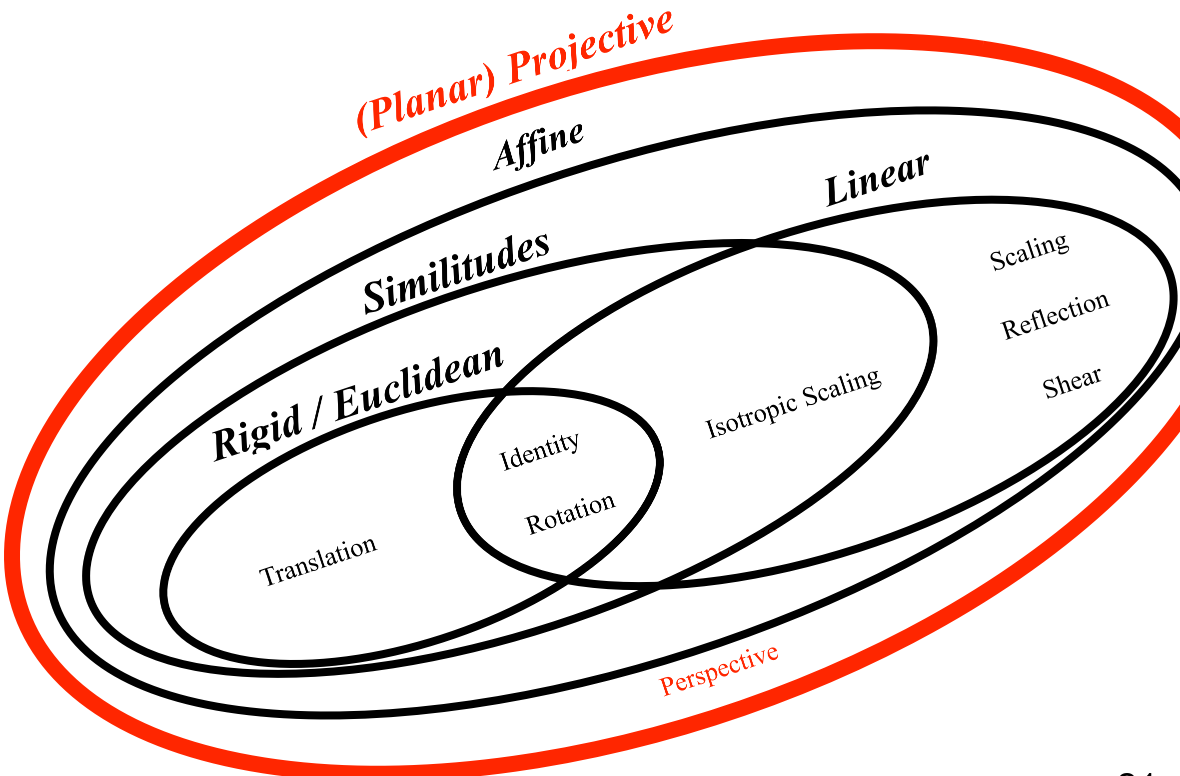
- Preserves lines: lines remain lines (planes remain planes in 3D)





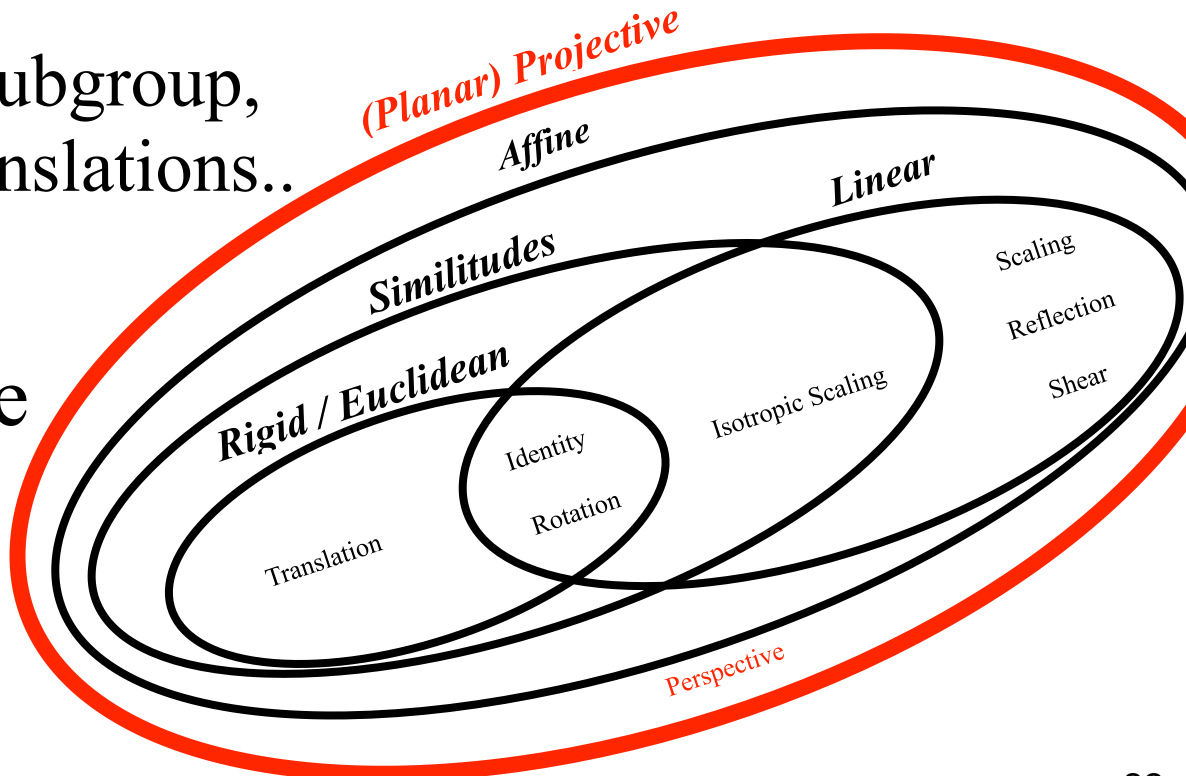
What's so nice about these?

- What's with the hierarchy? Why have we grouped types of transformations with and within each other?



What's so nice about these?

- They are **closed** under concatenation
  - Means e.g. that an affine transformation followed by another affine transformation is still an affine transformation
  - Same for every subgroup, e.g. rotations, translations..
- Very convenient!
- Projections are the most general
  - Others are its special cases



Name-dropping

- Fancy name: Group Theory
- Remember algebra?
  - A group is a set S with an operation f that takes two elements of S and produces a third:

$$s, t \in S,\ f(s,t) = u \ \Rightarrow\ u \in S$$

  (and some other axioms)

- These transformations are group(s) and subgroups
  - The transformations are the set S, concatenation of transformations is f

Transforms are Groups

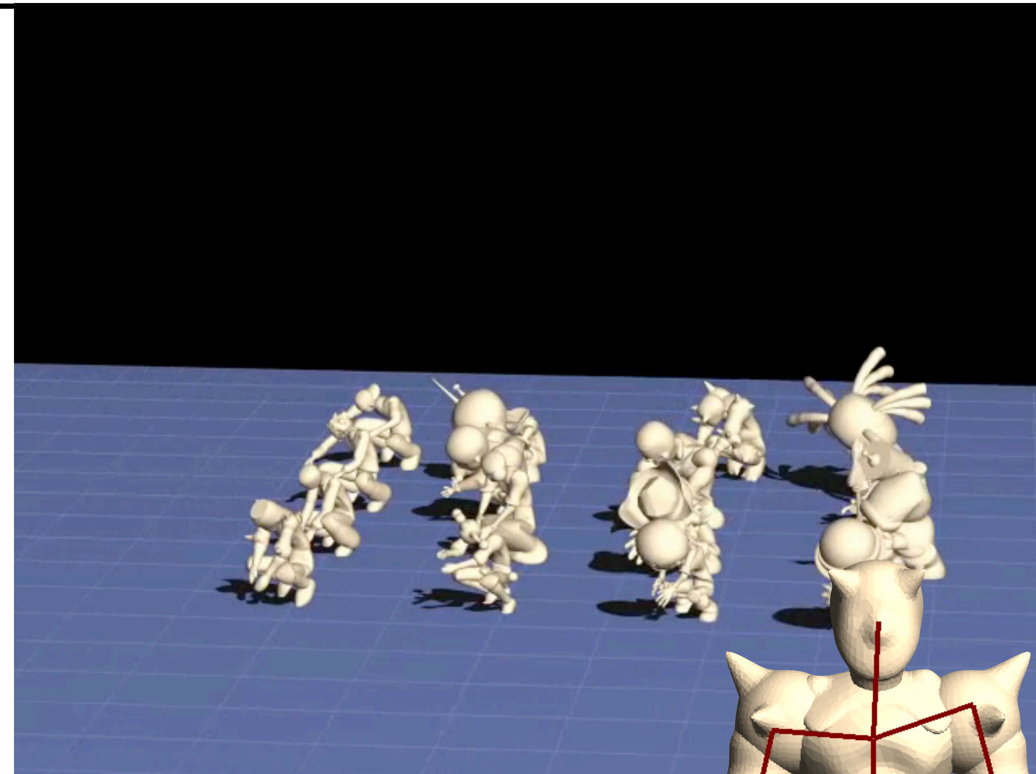
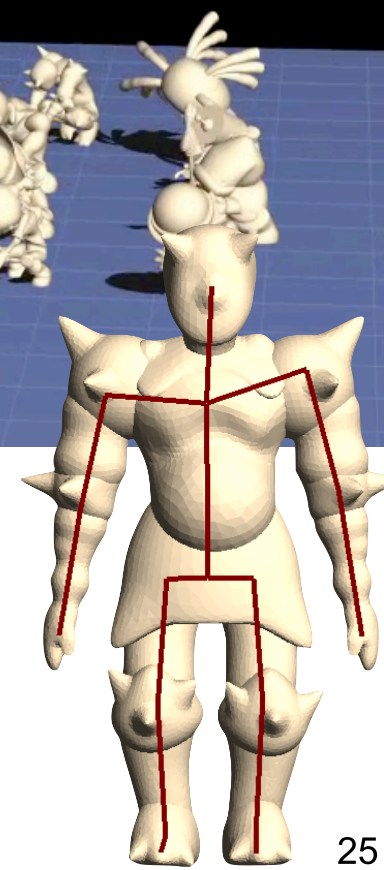
- Why is this useful?
  - **You can represent any number of successive transformations by a single compound transformation**
- Example
  - The object-to-world transformation, the world-to-view transformation, and the perspective projection (view-to-image) can all be folded into a single projective object-to-image transformation
  - (OpenGL: Modelview, projection)

Disclaimer: Not ANY transformation, but the types just introduced

Object coordinates
World coordinates
View coordinates
Image coordinates

More Complex Transformations..

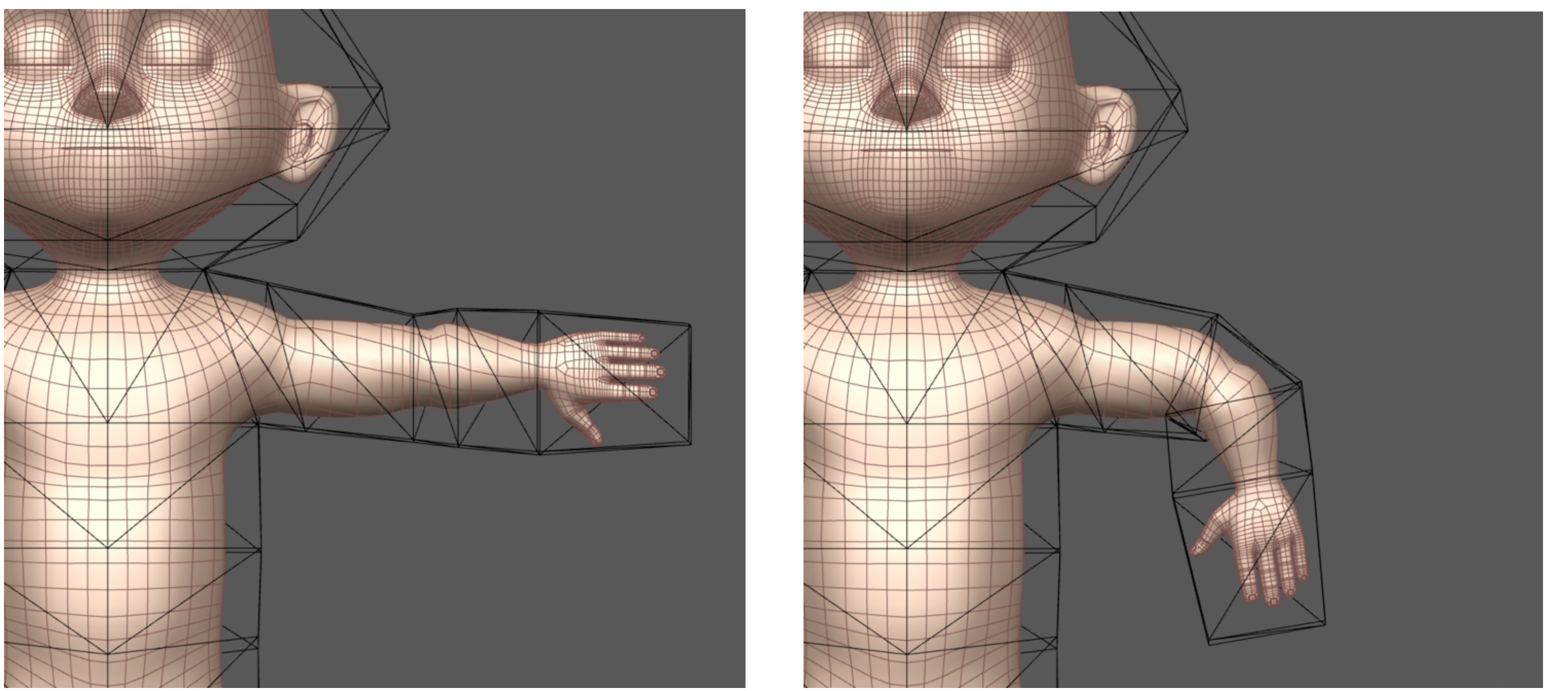
- ...can be built out of these, e.g.
- "Skinning"
  - Blending of affine transformations
  - We'll do this later.. and you will code it up! :)

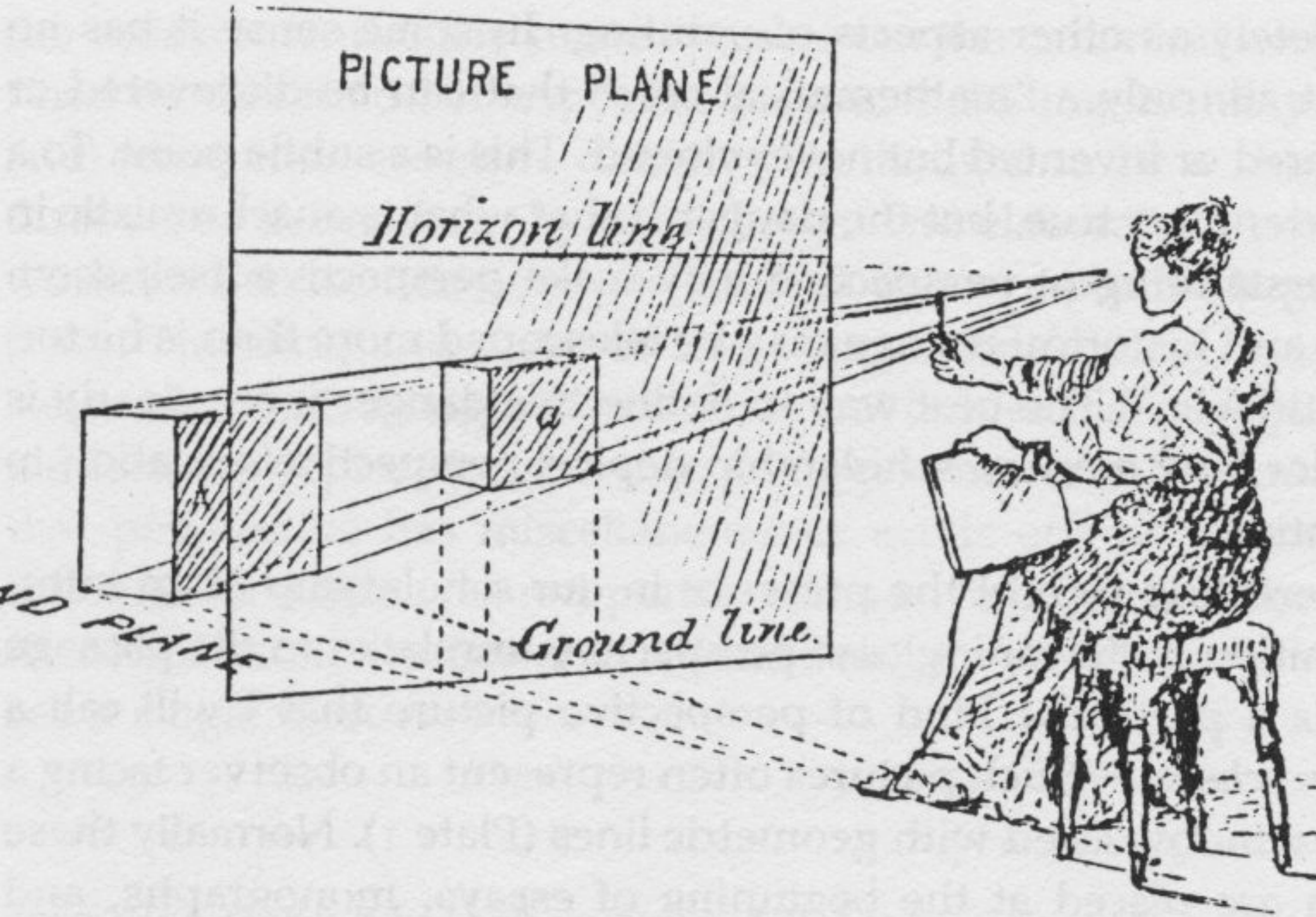
Ilya Baran

More Complex Transformations

- Harmonic coordinates (link to paper)
  - Object enclosed in simple "cage", each object point knows the influence each cage vertex has on it
  - Deform the cage, and the object moves!



Pixar


PICTURE PLANE
Horizon line.
Ground line.
a
A
ND PLANE

Key Concepts

- Geometric transformations change the positions/ coordinates of points in space
- Translation, scaling, rotation, shearing, reflection, and planar perspective transformations are the building blocks of graphics
  - And, as you will see in the next two videos, they can all be represented using matrices
- More complex ones can be built out of them