Discussion 4

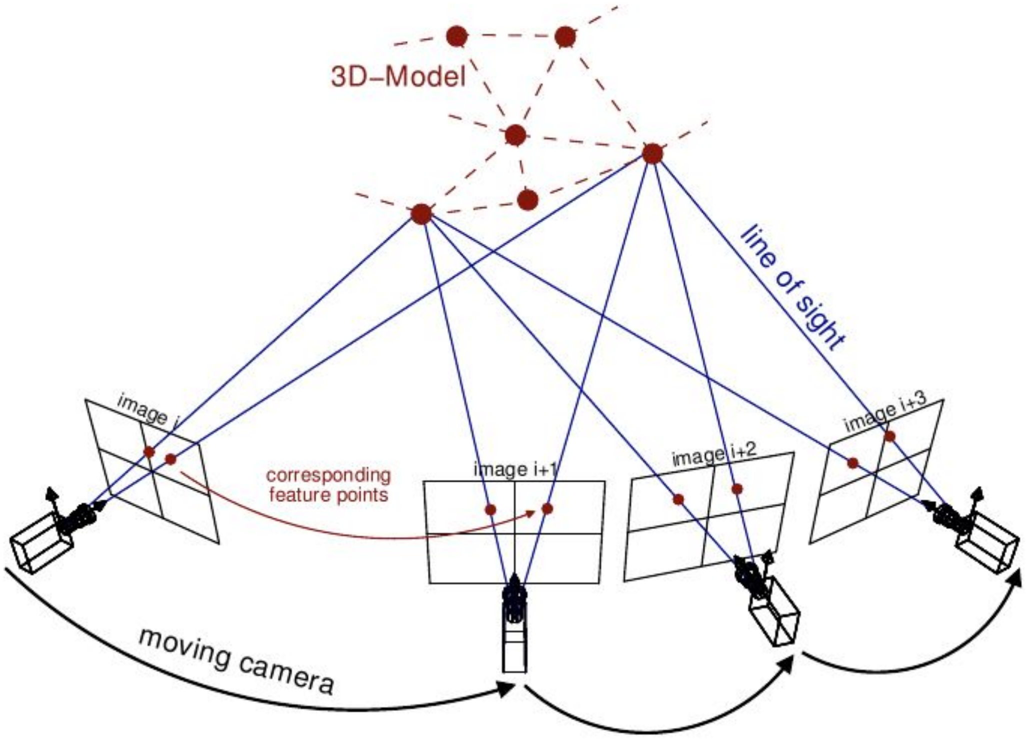
CS188 - Fall 19

Objectives

- What are the past 2 lecture driving towards?
  - Structure from Motion (SfM)
  - Simultaneous Localization and Mapping (SLAM)
  - Depth Prediction
  - Pose Estimation
- A TLDR on Photometry
- Geometry
  - Homogeneous coordinates
  - Affine spaces and the perspective projection
  - Transformations in the image and projective transformations
- Some potential exam problems ...

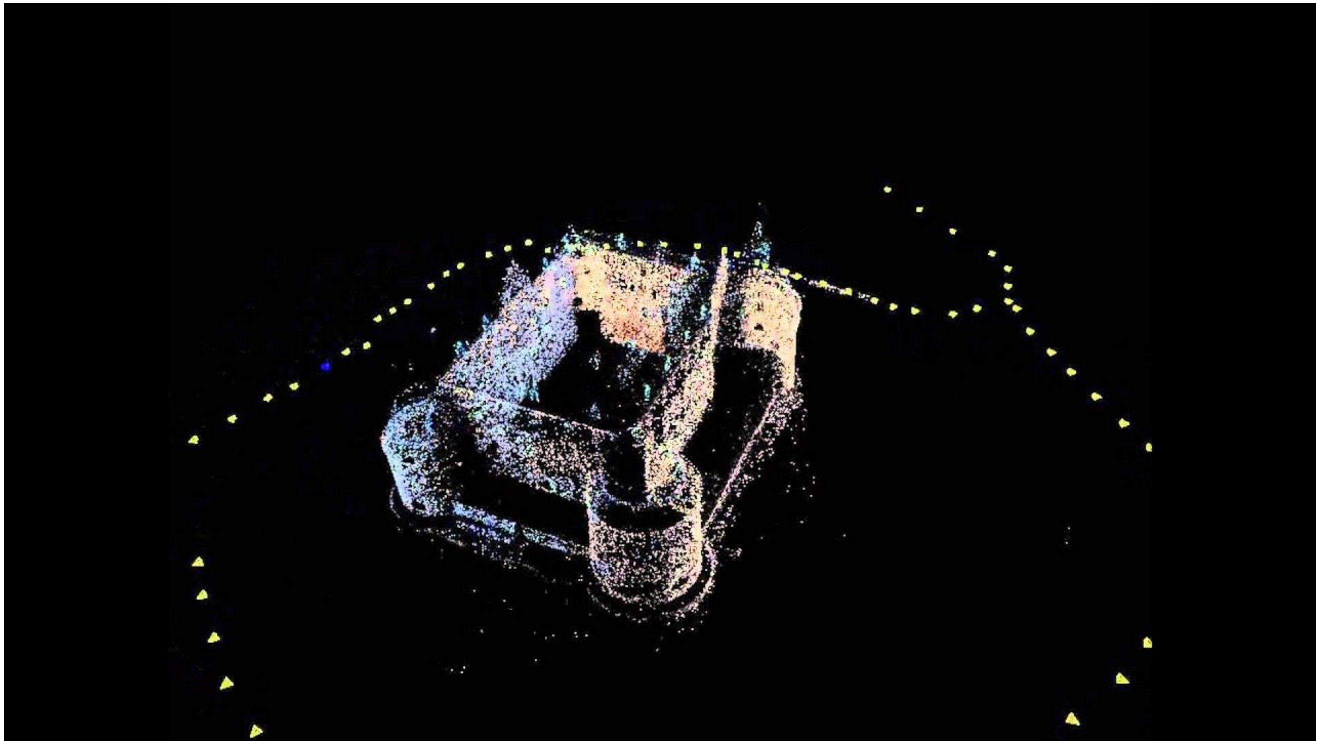
Structure from Motion (SfM)

Given images of the same scene from different angles (obtained through movement), can we reconstruct a 3D model (structure) of the scene?



SfM: Example

Yes!

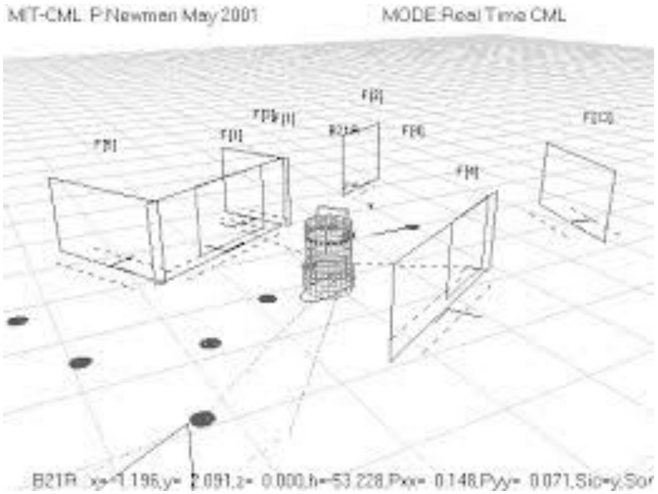


Simultaneous Localization and Mapping

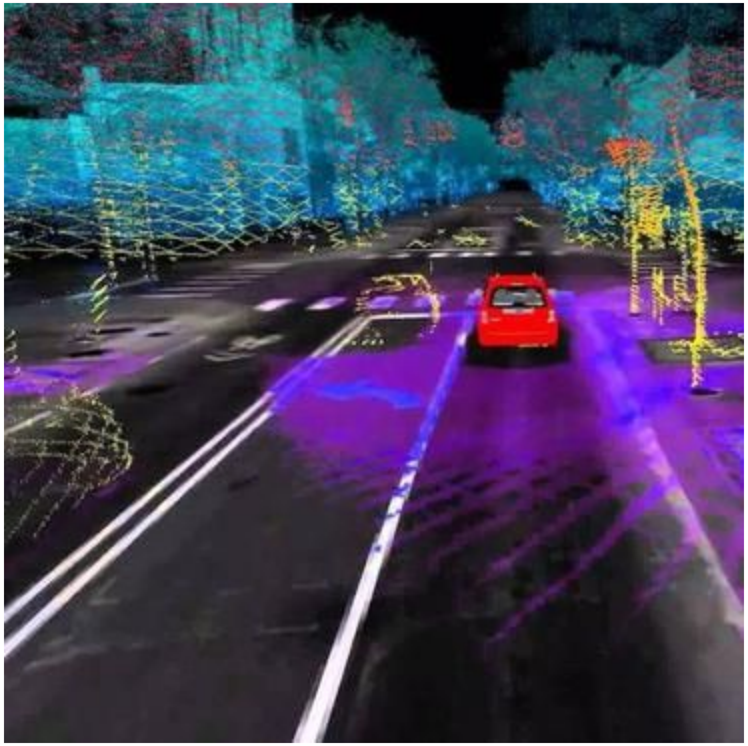
SLAM is extensively used by autonomous vehicles, but also robotics, and is starting to see applications in the medical field.

Given a starting position and a sequence of images taken from that position:

Can I track my trajectory in the scene and create a map?



SLAM: other visualizations



Depth Estimation / Epipolar Geometry

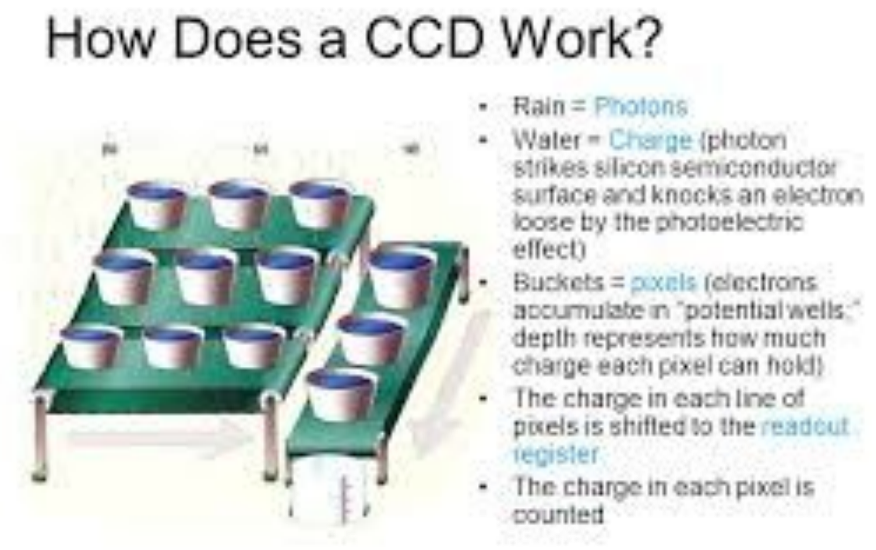
On Google Collab …

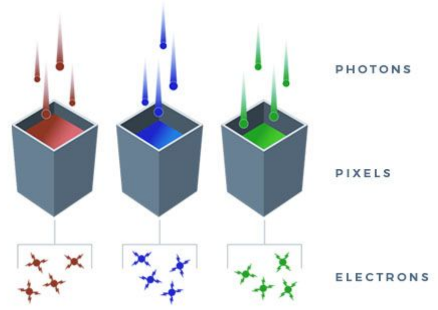
A conclusion

- We need to understand a few things:
  - How the scene 3D is projected onto the image plane
    - Geometry (points => pixels)
    - Photometry (light => intensities)
  - How moving around in the scene affects the images we take of the scene
    - How are surfaces deformed by camera motion?
    - Are any properties of the image preserved?
- To be able to invert the problem
  - Can I deduce **from the transformation in my images** (without any information about the scene itself):
    - What motion induced the transformations? (pose estimation)
    - How far points in the scene are? (depth estimation)
    - What the geometry of the scene is? (3D reconstruction)

Photometry: A TLDR

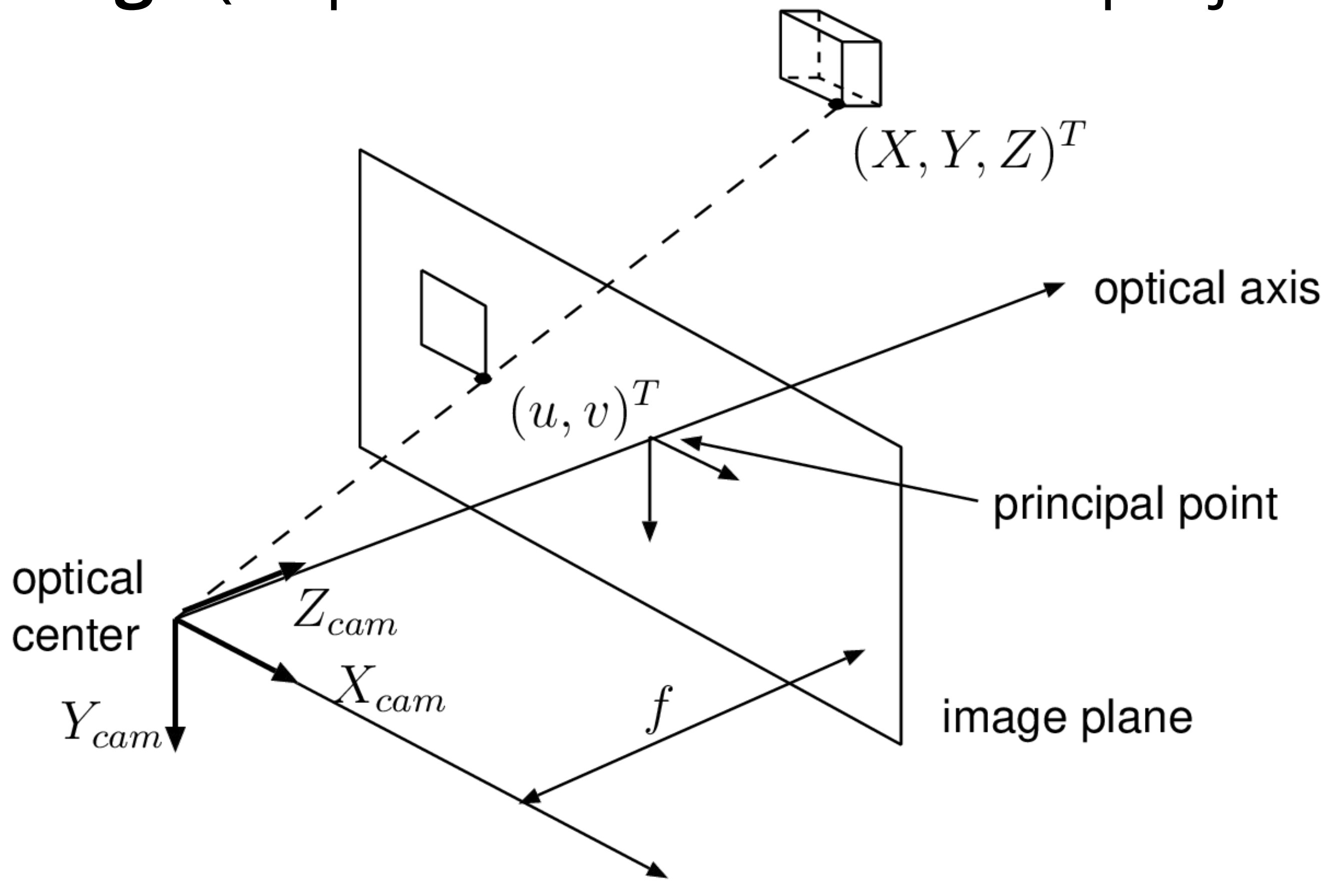
- The equations of light propagating in the scene are extremely complicated. Instead





Determining a pixel color

A pixel averages intensity values coming from points in its **preimage** (all points in the scene that project on that pixel)



The problem of infinity

- Fact: parallel lines in the scene (3D) cross at infinity in the image (2D)
  - Euclidean geometry cannot explain this => 5th axiom
  - We **need** something else, otherwise we can't understand how points in the scene map to pixels in our images
  - What we'll use instead is projective geometry
- Projective geometry 'just' replaces the 5th axiom
  - Euclidean => parallel lines don't cross
  - Projective => parallel lines cross exactly once
- This small change induces a lot of non intuitive behavior
  - But you'll make sense of it

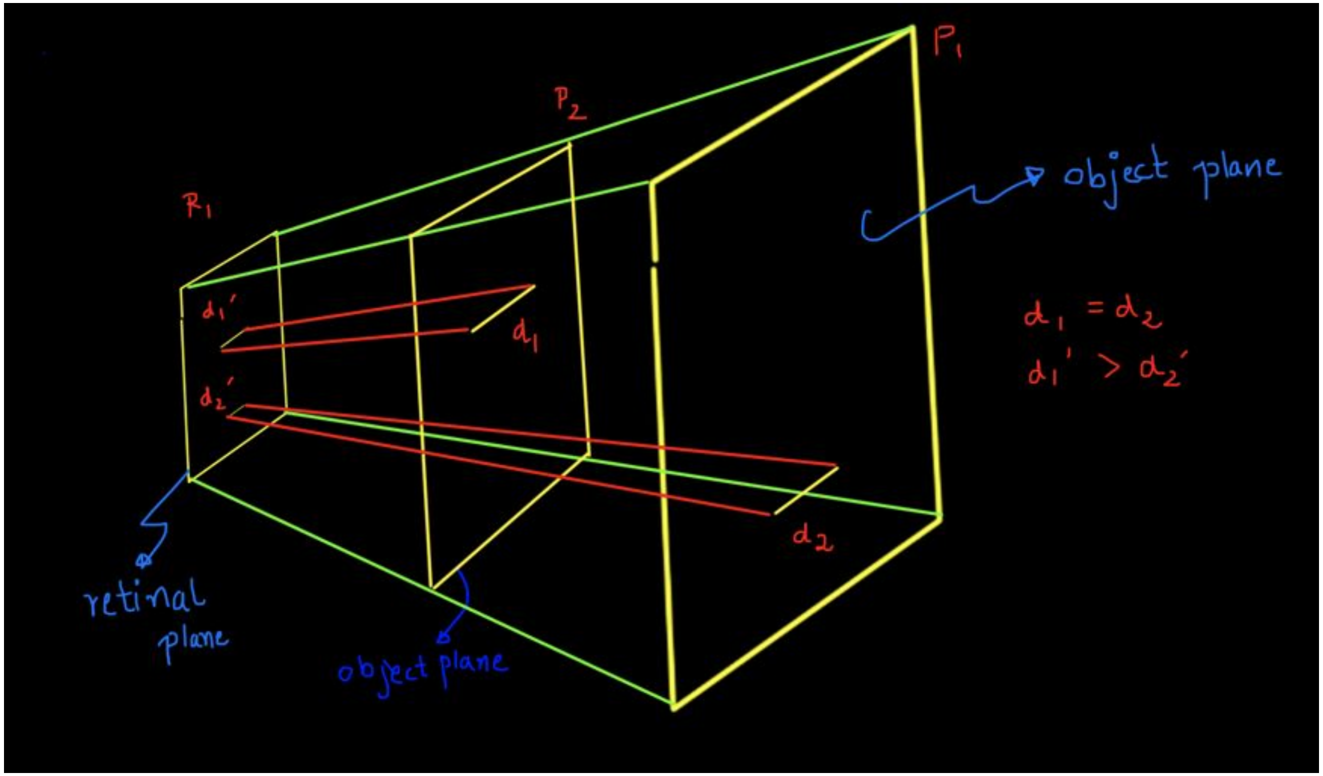
Forgetting Euclidean geometry?

- The image plane still follows the rules of Euclidean geometry
  - Two parallel lines **in the image** do not intersect
  - So once an image is obtained, we still very much depend on Euclidean geometry

- Where we use something else is to translate 3D geometry to 2D
  - Parallel lines in space **will** intersect in the image at infinity
  - **Infinity is not present in the image**, but the crossing point is
  - We need a way to **express infinity with finite coordinates**
  - This is where homogeneous coordinates come in

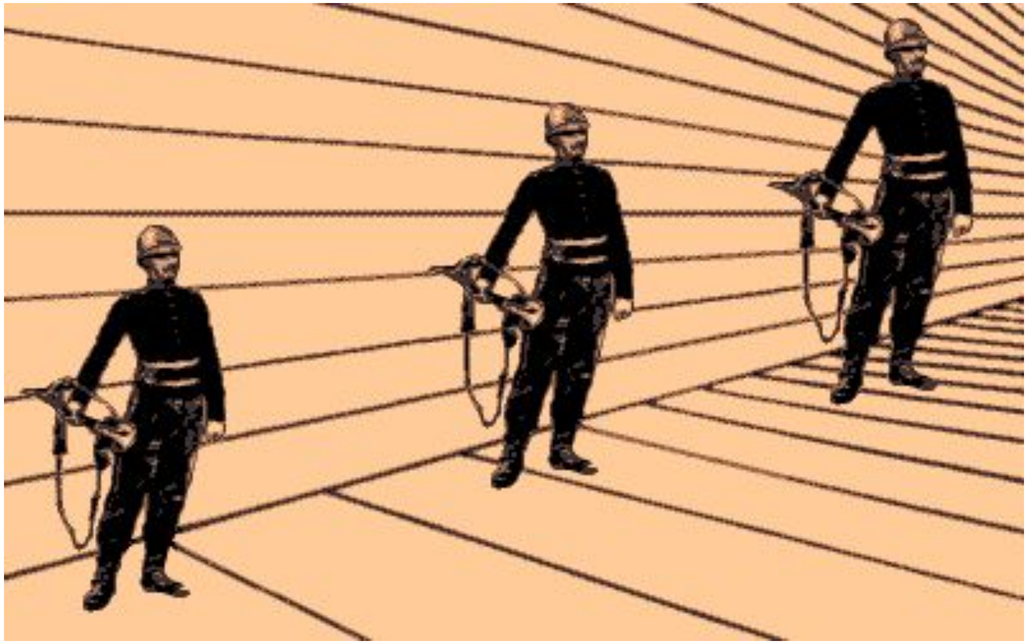
Projective geometry

Remember, we care about how pixels in the scene are **projected** on pixels



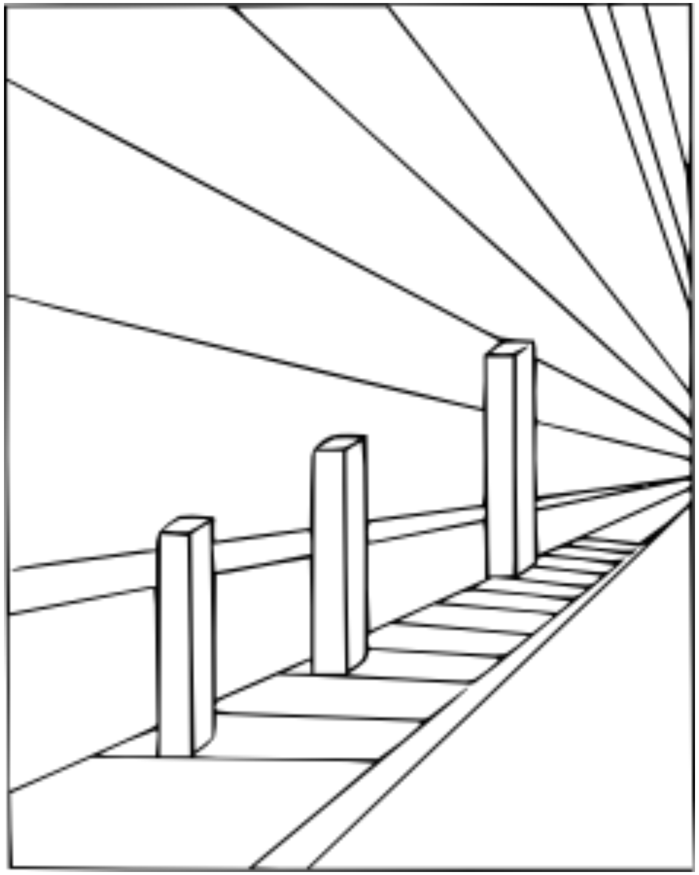
The Ponzo effect (a detour)

Which soldier is bigger?



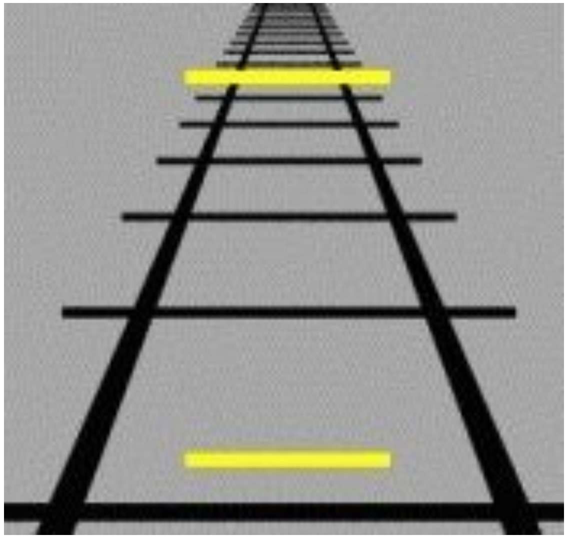
The Ponzo effect (a detour)

Which rectangle is bigger ?



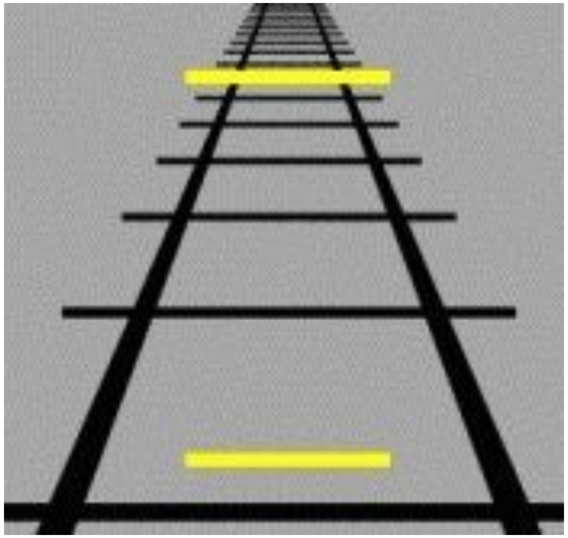
The Ponzo effect (a detour)

Which yellow line is longer ?



The Ponzo effect (a detour)

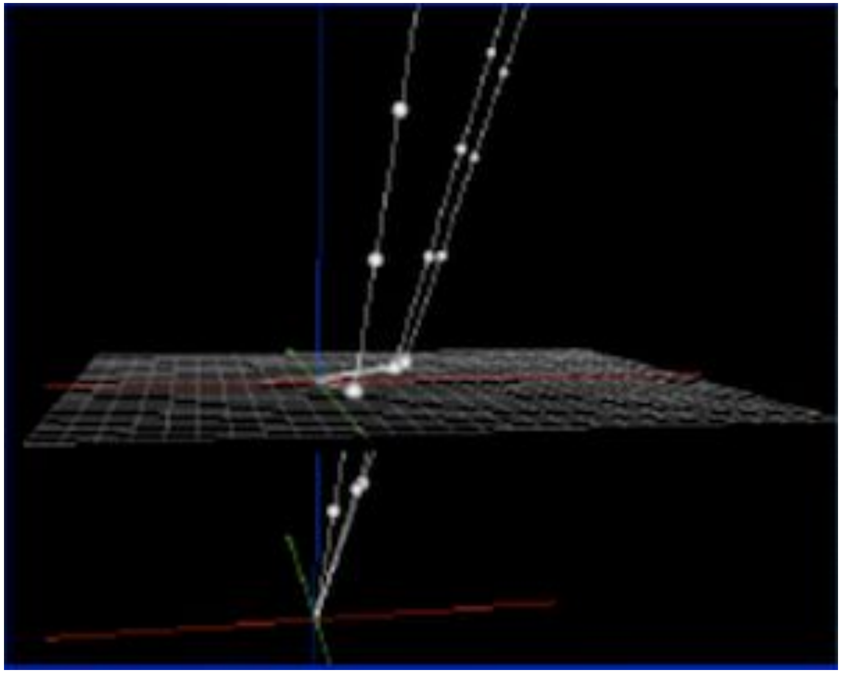
Which yellow line is longer ?



In fact, all soldiers, rectangles and lines are exactly the same. The Ponzo effect shows that the human brain uses background to determine object size: you are amazing at projective geometry already !

Homogeneous coordinates

- You can think of it as ‘adding a dimension’
  - Points in 2D become lines (through the origin) in 3D
- How?
  - All points on the same line =>
  - Map to one grid point =>



We only care about where the 3D points fall on the 2D grid!

In that sense, they’re all the same

Homogeneous vs Cartesian

You've seen this before, we add a dimension:

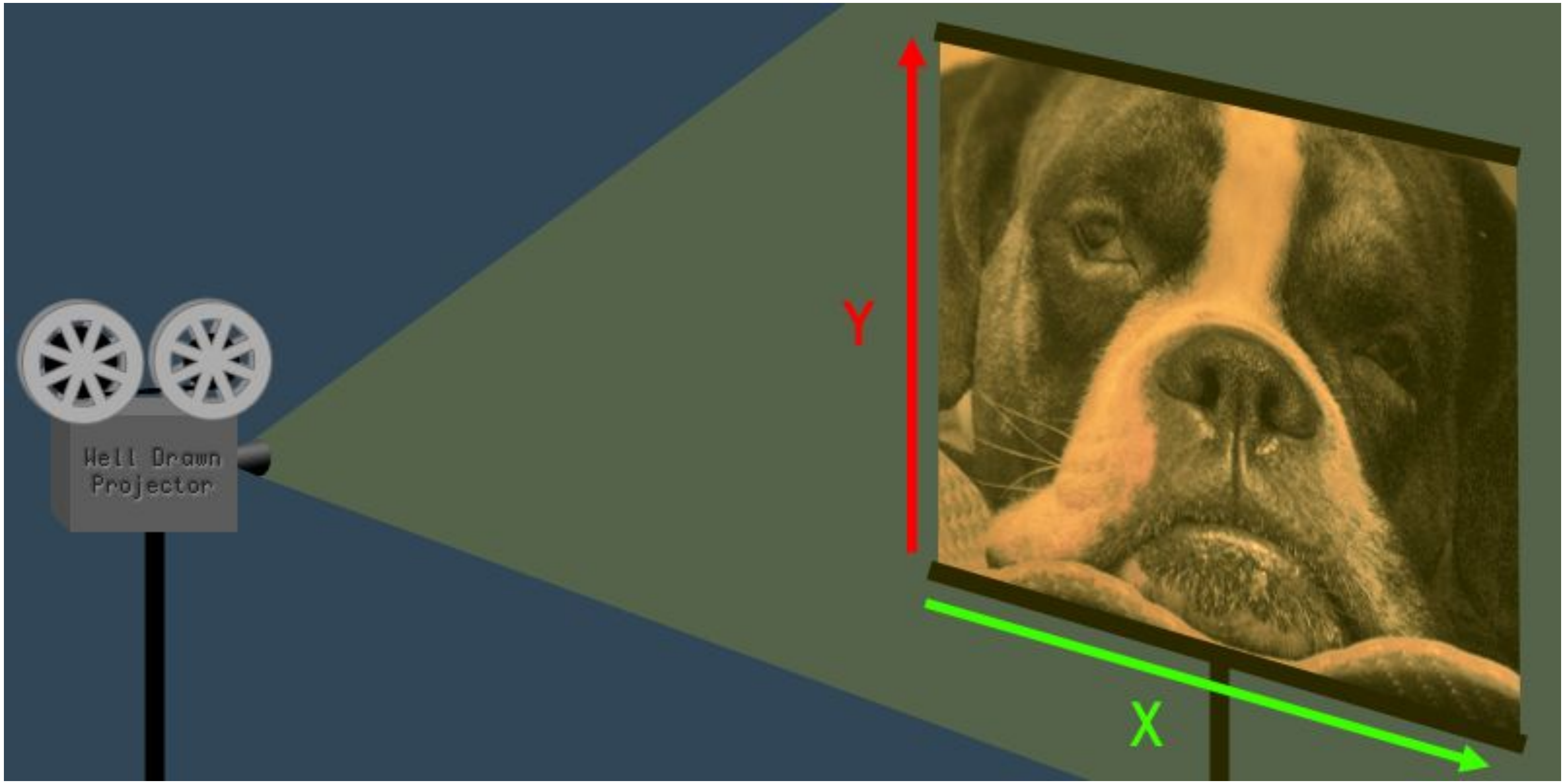
$$\underset{\text{Homogeneous}}{(x, y, w)} \quad \Leftrightarrow \quad \underset{\text{Cartesian}}{\left(\frac{x}{w}, \frac{y}{w}\right)}$$

If we set w = 0, we managed to represent infinity in cartesian coordinates with 3 finite homogeneous coordinates! This is a neat trick, but what is w really?

Scale as an extra dimension

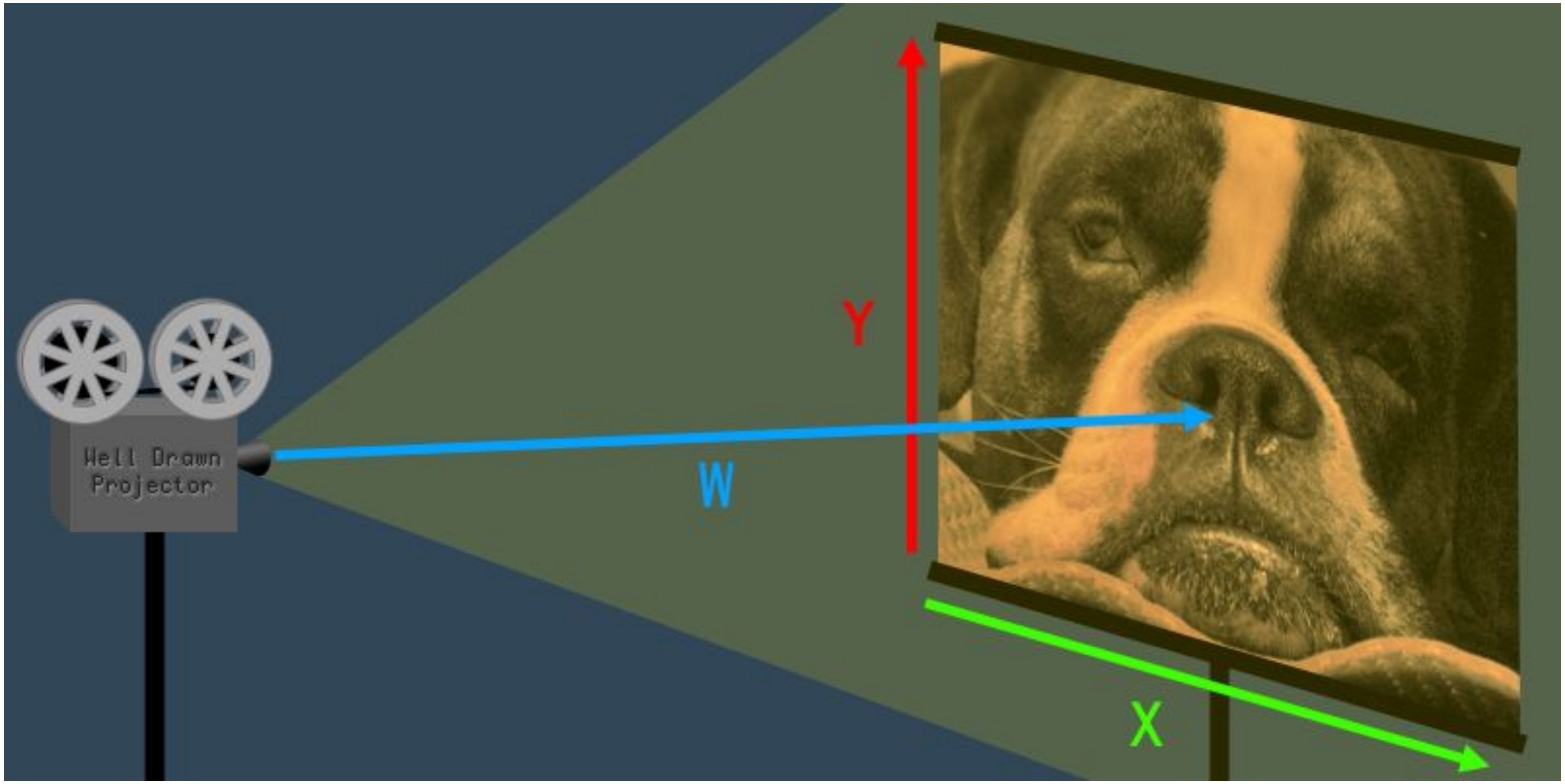
A projector is displaying an image on the screen.

The image on the lens of the projector is fixed



Scale as an extra dimension

You can picture w as the distance between the projector and the screen: a ‘scale factor’

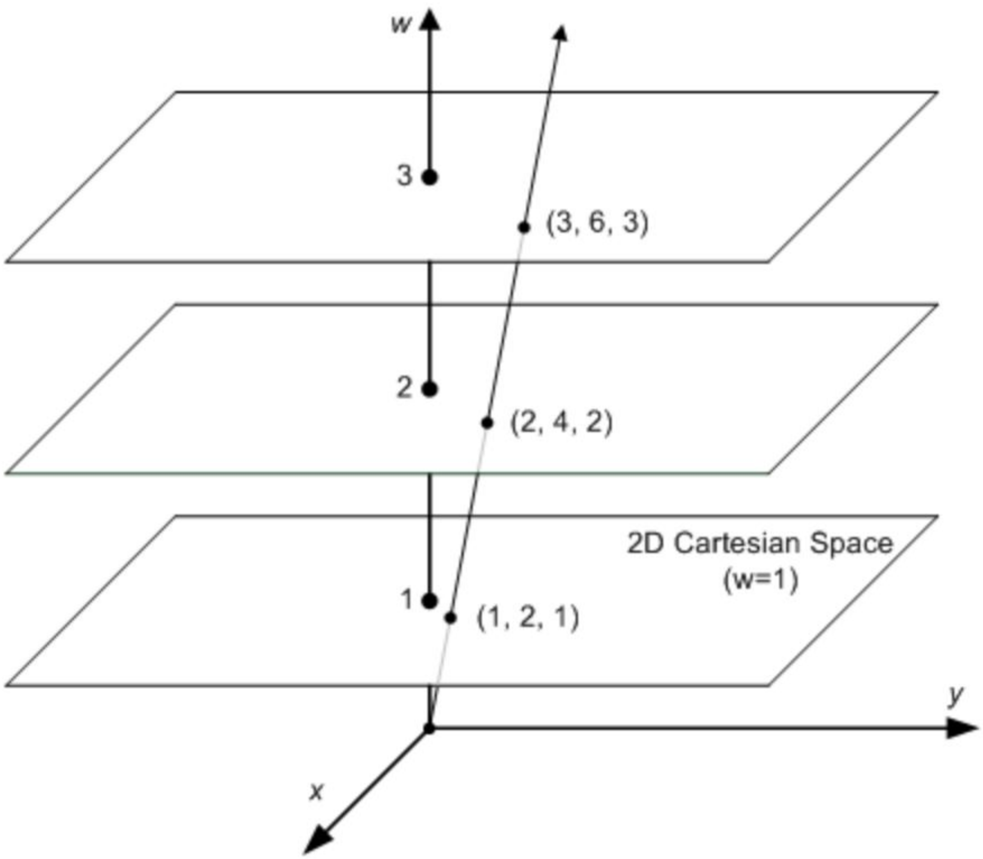


Scale as an extra dimension

If the projector moves closer, the image on the lens is still the same, but it gets smaller on the screen

How can we see this mathematically?

All points in the planes project on the same point



How can we see this mathematically?

All points in the planes project on the same point

$$
\begin{array}{ccccc}
\text{Homogeneous} & & \text{Cartesian} & & \\
(1,2,3) & \Rightarrow & \left(\frac{1}{3},\frac{2}{3}\right) & & \\
(2,4,6) & \Rightarrow & \left(\frac{2}{6},\frac{4}{6}\right) & = & \left(\frac{1}{3},\frac{2}{3}\right) \\
(4,8,12) & \Rightarrow & \left(\frac{4}{12},\frac{8}{12}\right) & = & \left(\frac{1}{3},\frac{2}{3}\right) \\
\vdots & & \vdots & & \\
(1a,2a,3a) & \Rightarrow & \left(\frac{1a}{3a},\frac{2a}{3a}\right) & = & \left(\frac{1}{3},\frac{2}{3}\right)
\end{array}
$$

The intersection of parallel lines

Cartesian:

$$\begin{cases} Ax + By + C = 0 \\ Ax + By + D = 0 \end{cases}$$

Need C = D => same lines (A = B because lines are //)

Homogeneous:

$$\begin{cases} A\dfrac{x}{w} + B\dfrac{y}{w} + C = 0 \\ \\ A\dfrac{x}{w} + B\dfrac{y}{w} + D = 0 \end{cases} \Rightarrow \begin{cases} Ax + By + Cw = 0 \\ \\ Ax + By + Dw = 0 \end{cases}$$

Vanishing points

Homogeneous:

$$\begin{cases} A\dfrac{x}{w} + B\dfrac{y}{w} + C = 0 \\ \\ A\dfrac{x}{w} + B\dfrac{y}{w} + D = 0 \end{cases} \Rightarrow \begin{cases} Ax + By + Cw = 0 \\ \\ Ax + By + Dw = 0 \end{cases}$$

w = 0 works!

This is exactly the point at infinity.

This point is called a **vanishing point**

A conclusion

- We need to understand how the geometry of the scene (which is a collection of 3D surfaces) is represented onto the image plane
- The 3D scene is projected onto the image plane
  - Which creates funky things, like parallel lines crossing on the finite image coordinates
- To explain this, we had to move away from Euclidean geometry
  - Using projective geometry calls for a new system of coordinate, the homogeneous coordinates
  - This allows to represent points at infinity in finite coordinates
  - This allows to find the intersection of parallel lines

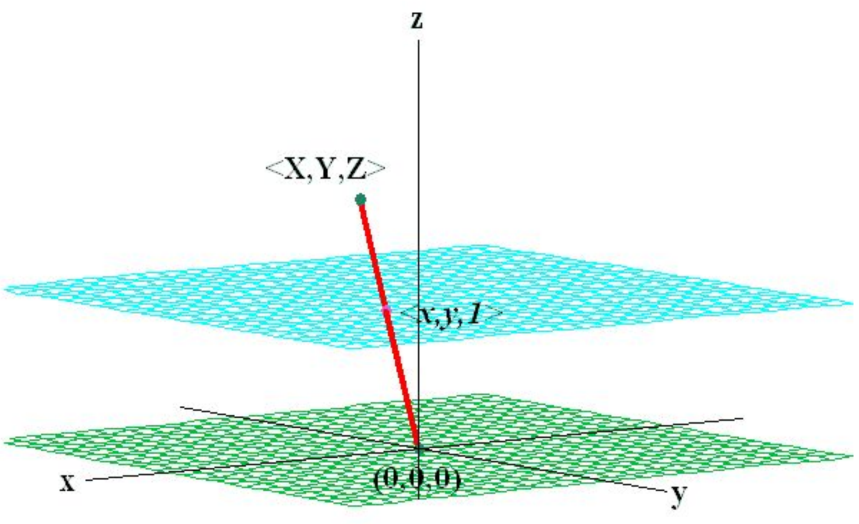
The problem

- Projections are not linear transformations
  - We cannot leverage linear algebra
- The scale factor we added doesn't **fully** add a dimension
  - Changing w doesn't change the point we're considering
  - We cannot use w=0 in space, this is reserved for points at infinity
- By doing this trick, we moved from a **vector space** to an **affine space**
  - The sum of 2 elements in a vector space is a vector in that space
  - A vector space is **closed** over addition
  - This doesn't hold for an affine space
  - Which implies that an affine space loses the notion of origin
  - It also loses the notion of an inverse

Vector space vs affine space

The affine space (in blue) is still 2 dimensional, but embedded in 3D.

Points are no longer vectors without the origin!



The solution

- Homogeneous coordinates allow us to ‘force’ projections in the linear algebra realm and express them with matrices
- **BUT** your original space is affine, watch out …
  - We are expressing a projection with a matrix, it does not make a projection linear
- We call this projection a **perspective projection**
  - It maps a point in 3D space to a point in the image plane

What did we do so far

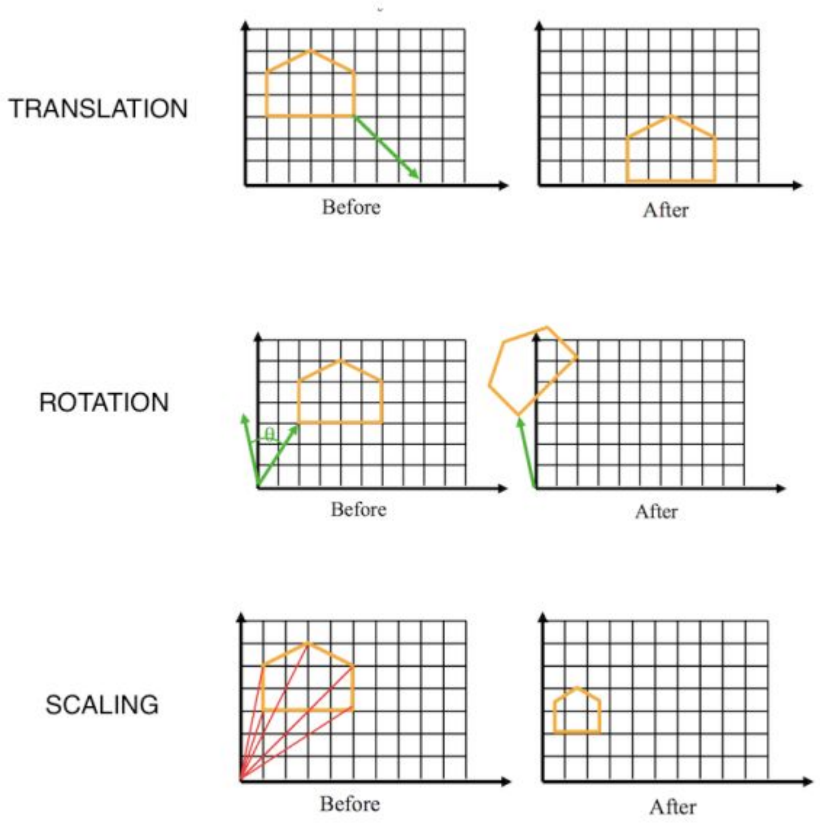
- We built a framework to understand how points in the scene will be projected on the image
- Homogeneous coordinates allow us to write this projection from 3D to 2D in matrix form
    - I won't go into that today, the next lectures will

- This was one of the things we needed to do to be able to perform SfM, SLAM, …

- The other was to understand how a change of perspective modifies the image

A review of transformations in 2D

- They will be composed of :



A review of transformations in 2D

- Isometric transformations (or Euclidean isometries)
  -

A review of transformations in 2D

- Isometric transformations (or Euclidean isometries)
  - Preserve distances (iso = same)
  - Can be written as a rotation and translation
- Similarity transformation
  -

A review of transformations in 2D

- Isometric transformations (or Euclidean isometries)
  - Preserve distances (iso = same)
  - Can be written as a rotation and translation
- Similarity transformation
  - Preserves shapes (and hence ratio of lengths and angles)
  - Can be seen as an isometric transformation + scaling
  - For a scale of 1, a similarity is an isometry
- Affine transformation
  -

A review of transformations in 2D

- Isometric transformations (or Euclidean isometries)
  - Preserve distances (iso = same)
  - Can be written as a rotation and translation
- Similarity transformation
  - Preserves shapes (and hence ratio of lengths and angles)
  - Can be seen as an isometric transformation + scaling
  - For a scale of 1, a similarity is an isometry
- Affine transformation
  - Preserves points, straight lines, and parallelism
  - Isometries and similarities are special cases of affinities
- Projective transformation of homographies
  -

A review of transformations in 2D

- Isometric transformations (or Euclidean isometries)
  - Preserve distances (iso = same)
  - Can be written as a rotation and translation
- Similarity transformation
  - Preserves shapes (and hence ratio of lengths and angles)
  - Can be seen as an isometric transformation + scaling
  - For a scale of 1, a similarity is an isometry
- Affine transformation
  - Preserves points, straight lines, and parallelism
  - Isometries and similarities are special cases of affinities
- Projective transformation of homographies
  - Any transformation that maps lines to lines (doesn't need to preserve parallelism)
  - Generalizes affine transformations

Transformations in 2D: a little math

Isometry:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ v & b \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Similarity:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Affinity:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homography:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} SR & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad S = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix}$$

Transformations in 2D: a little math

Isometry:

Similarity:

Affinity:

Homography:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ v & b \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} SR & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad S = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix}$$