Transformations

The Ambassadors (Holbein), 1533

Alexei Efros
CS280, Spring 2024

Linear Envy

- (Scaled) Orthographic Projection

$$x = mX, y = mY$$

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

- Perspective Projection

$$x = f\,\frac{X}{Z}, y = f\,\frac{Y}{Z}$$

Homogeneous coordinates

Trick: add one more coordinate!

Converting to *homogeneous* coordinates

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image coordinates

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

homogeneous scene coordinates

Converting *from* homogeneous coordinates

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

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

Perspective Projection Matrix

Projection is a matrix multiplication using homogeneous coordinates

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f \end{bmatrix} \Rightarrow \left(f\frac{x}{z}, f\frac{y}{z}\right)$$

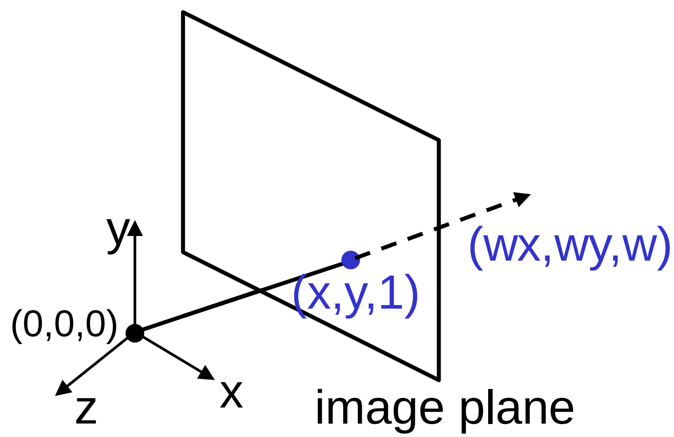
divide by the third coordinate

It's not just some hack…

Projective Geometry: the study of geometric properties that are invariant under projective transformations.

What is the geometric intuition?

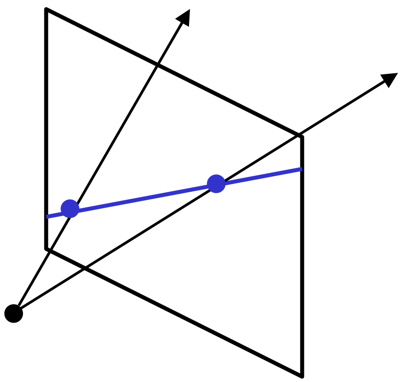
- a point in the image is a *ray* in projective space



- Each *point* (x,y) on the plane is represented by a *ray* (wx,wy,w)
  - all points on the ray are equivalent: (x, y, 1) ≅ (wx, wy, w)

Projective lines

What does a line in the image correspond to in projective space?



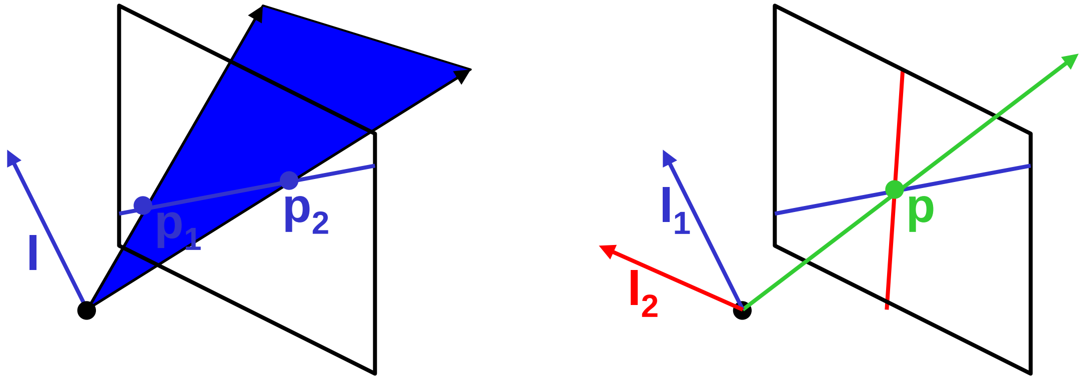
- A line is a *plane* of rays through origin
  - – all rays (x,y,z) satisfying: ax + by + cz = 0

$$\text{in vector notation}: \quad 0 = \underbrace{\begin{bmatrix} a & b & c \end{bmatrix}}_{\mathbf{l}} \underbrace{\begin{bmatrix} x \\ y \\ z \end{bmatrix}}_{\mathbf{p}}$$

- A line is also represented as a homogeneous 3-vector **l**

Point and line duality

- Since **l p**=0
- **l** is $\perp$ to every point (ray) **p** on the line



What is the line **l** spanned by rays $\mathbf{p_1}$ and $\mathbf{p_2}$?

- **l** is $\perp$ to $\mathbf{p_1}$ and $\mathbf{p_2}$ $\Rightarrow$ $\mathbf{l} = \mathbf{p_1} \times \mathbf{p_2}$
- **l** is the plane normal

What is the intersection of two lines $\mathbf{l_1}$ and $\mathbf{l_2}$?

- **p** is $\perp$ to $\mathbf{l_1}$ and $\mathbf{l_2}$ $\Rightarrow$ $\mathbf{p} = \mathbf{l_1} \times \mathbf{l_2}$

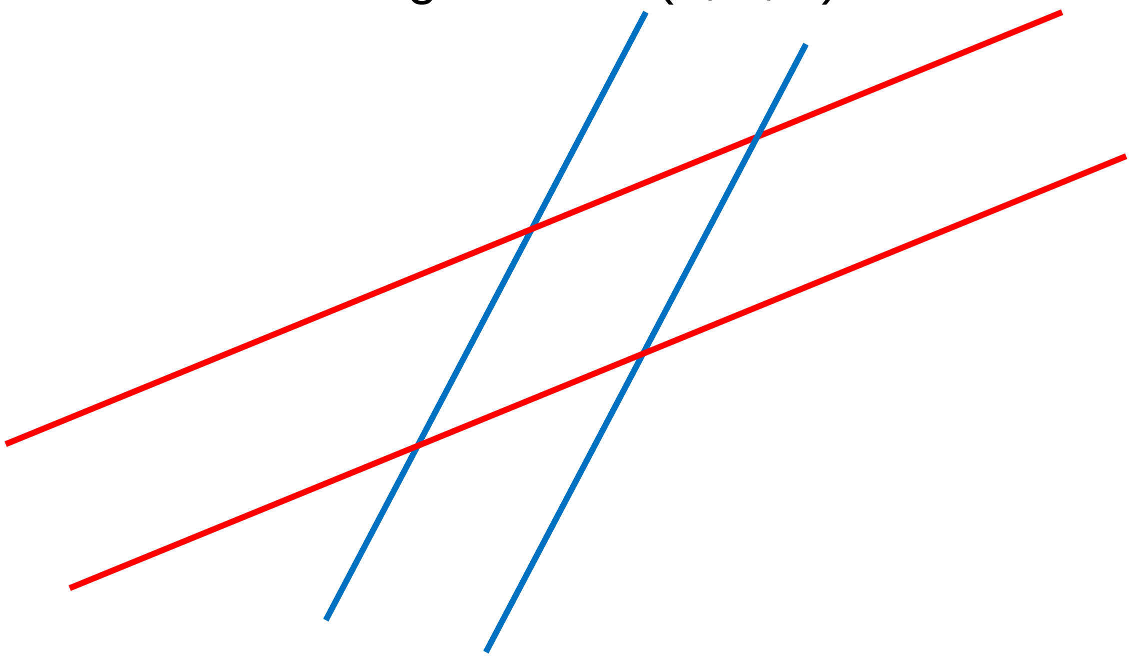
Points and lines are *dual* in projective space!

Another problem solved

Intersection of parallel lines

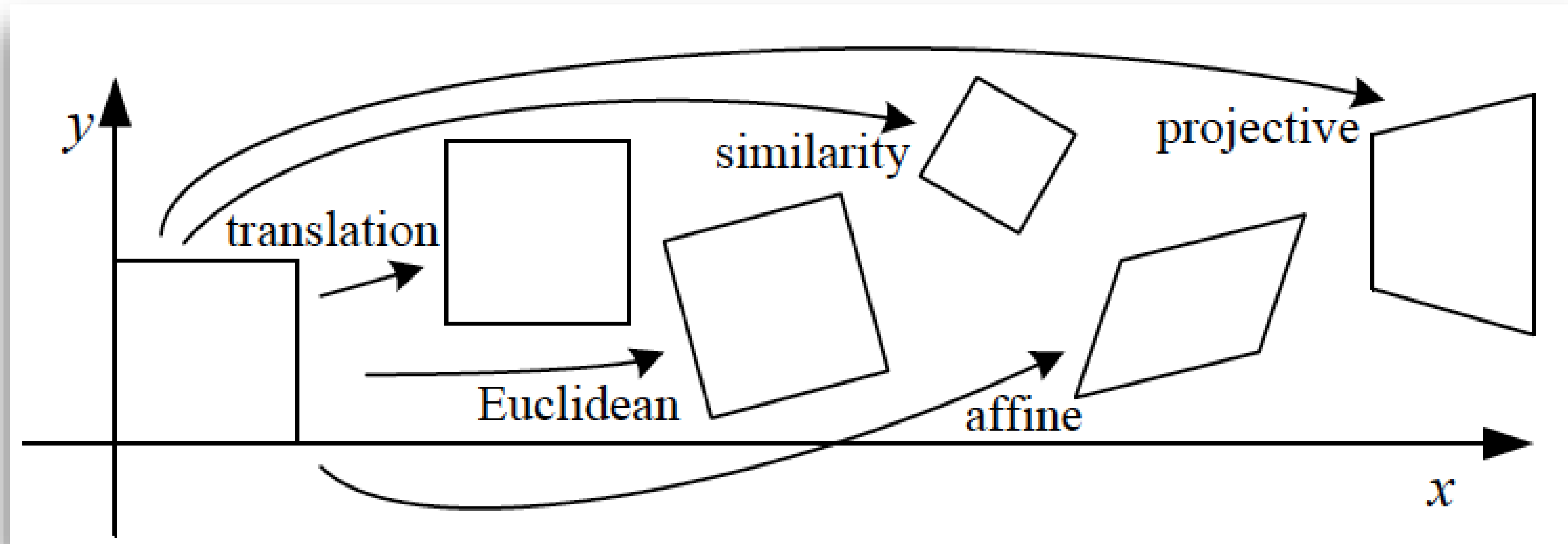
Cartesian: (Inf, Inf)
Homogeneous: (1, 1, 0)

Cartesian: (Inf, Inf)
Homogeneous: (1, 2, 0)



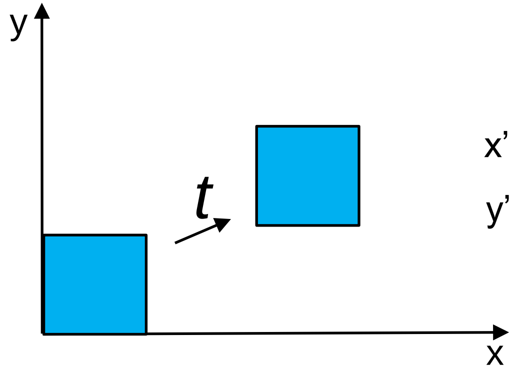
They intersect at <u>different</u> infinities!

These 2D Transformations can all be represented as matrix multiplications in homogeneous coordinates



2D Transformations

Translation

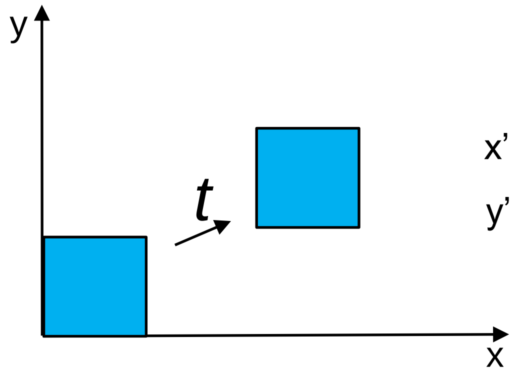


$x' = x + t_x$

$y' = y + t_y$

2D Transformations

Translation



$x' = x + t_x$

$y' = y + t_y$

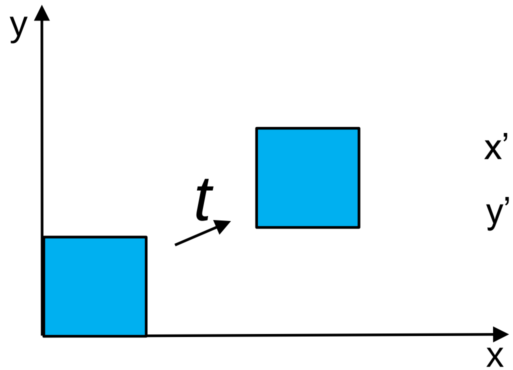
Heterogeneous

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{t}$$

2D Transformations

Translation



$x' = x + t_x$

$y' = y + t_y$

Heterogeneous

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Homogeneous to heterogeneous

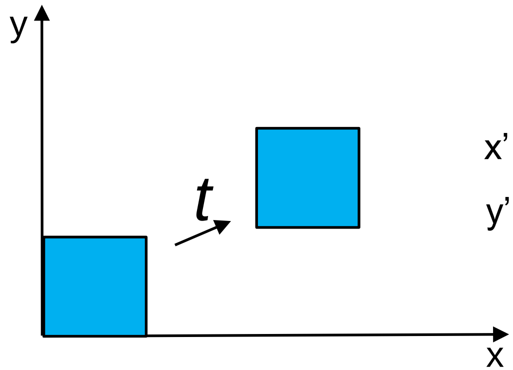
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{t}$$

$$\boldsymbol{x}' = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{t} \end{bmatrix} \bar{\boldsymbol{x}}$$

2D Transformations

Translation



$x' = x + t_x$

$y' = y + t_y$

Heterogeneous

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{t}$$

Homogeneous to heterogeneous

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\boldsymbol{x}' = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{t} \end{bmatrix} \bar{\boldsymbol{x}}$$

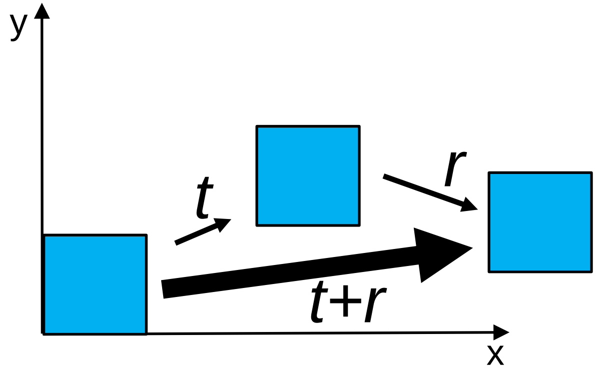
Homogeneous

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\bar{\boldsymbol{x}}' = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{t} \\ 0^T & 1 \end{bmatrix} \bar{\boldsymbol{x}}$$

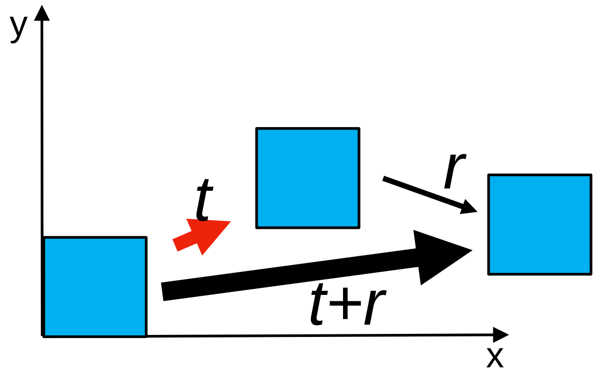
2D Transformations

Translation



2D Transformations

Translation

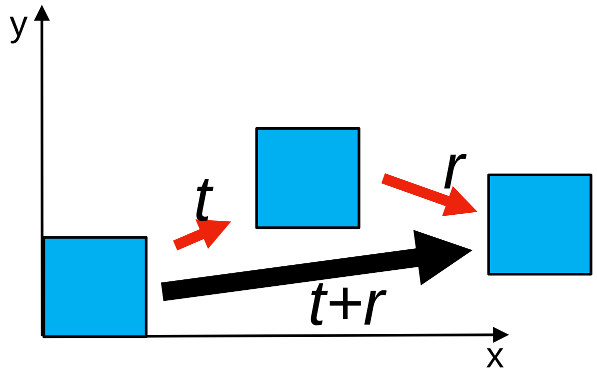


Now we can chain transformations

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2D Transformations

Translation



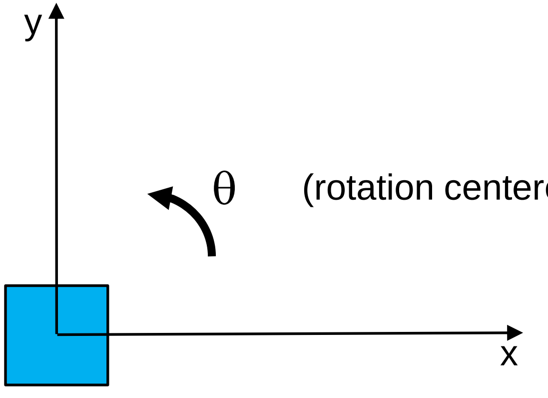
Now we can chain transformations

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & r_x \\ 0 & 1 & r_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$x' = RTx$$

2D Transformations

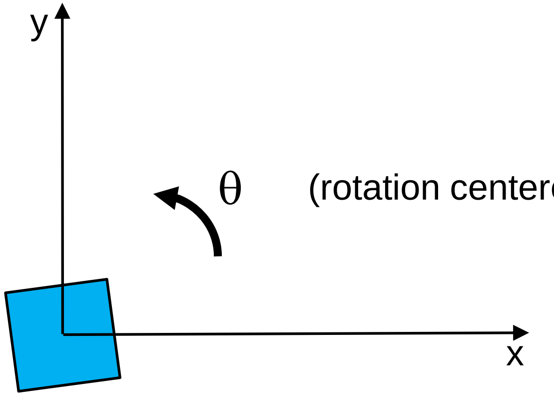
Rotation



(rotation centered on the origin)

2D Transformations

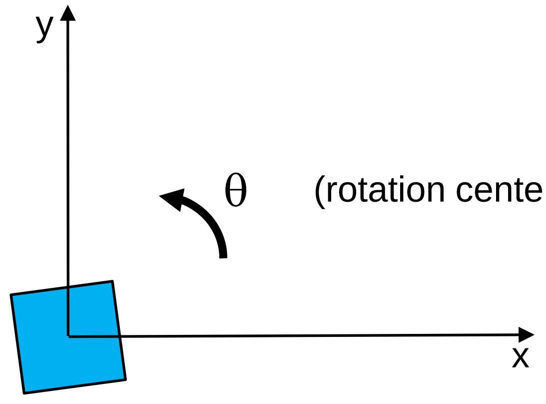
Rotation



(rotation centered on the origin)

2D Transformations

Rotation

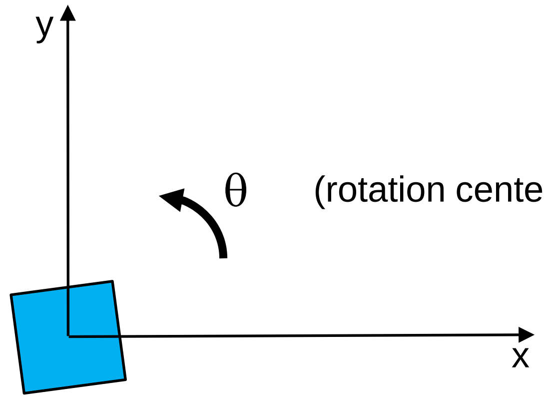


(rotation centered on the origin)

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2D Transformations

Rotation

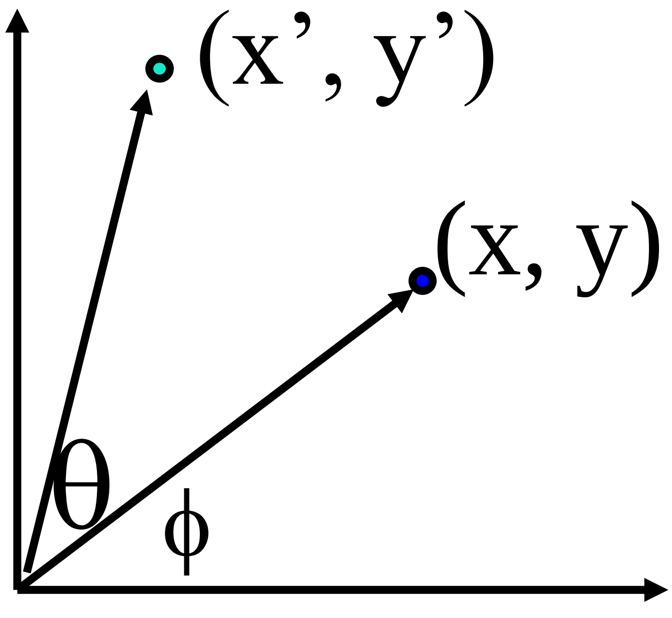


(rotation centered on the origin)

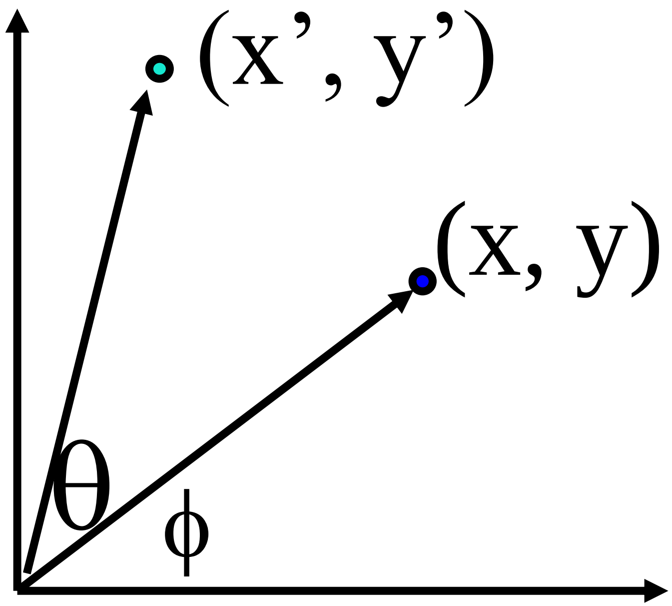
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2-D Rotation

$x = r \cos(\phi)$
$y = r \sin(\phi)$
$x' = r \cos(\phi + \theta)$
$y' = r \sin(\phi + \theta)$



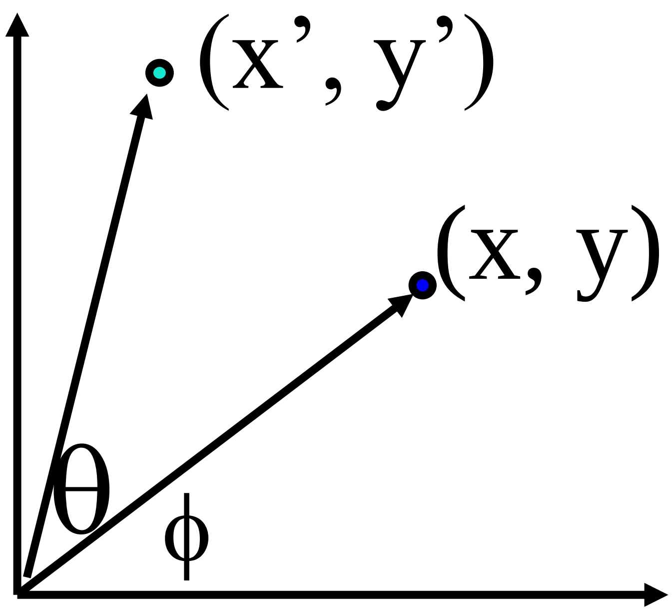
2-D Rotation



$x = r\cos(\phi)$
$y = r\sin(\phi)$
$x' = r\cos(\phi + \theta)$
$y' = r\sin(\phi + \theta)$

Trig Identity…
$x' = r\cos(\phi)\cos(\theta) - r\sin(\phi)\sin(\theta)$
$y' = r\sin(\phi)\cos(\theta) + r\cos(\phi)\sin(\theta)$

2-D Rotation



$x = r \cos(\phi)$
$y = r \sin(\phi)$
$x' = r \cos(\phi + \theta)$
$y' = r \sin(\phi + \theta)$

Trig Identity…

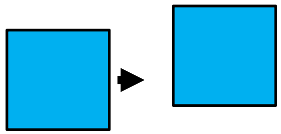
$x' = r \cos(\phi) \cos(\theta) - r \sin(\phi) \sin(\theta)$
$y' = r \sin(\phi) \cos(\theta) + r \cos(\phi) \sin(\theta)$

Substitute…

$x' = x\, \mathbf{cos}(\theta) - y\, \mathbf{sin}(\theta)$
$y' = x\, \mathbf{sin}(\theta) + y\, \mathbf{cos}(\theta)$

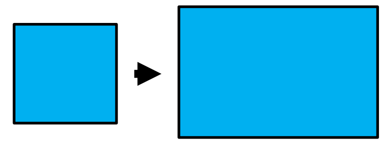
2D Transformations

Translation

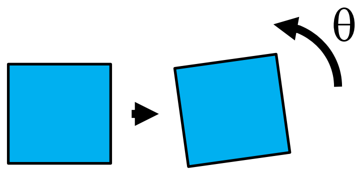


$$\begin{bmatrix}x'\\ y'\\ 1\end{bmatrix} = \begin{bmatrix}1 & 0 & t_x\\ 0 & 1 & t_y\\ 0 & 0 & 1\end{bmatrix} \cdot \begin{bmatrix}x\\ y\\ 1\end{bmatrix}$$

Scaling

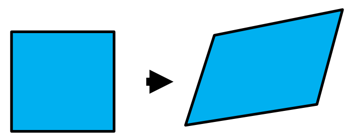


Rotation



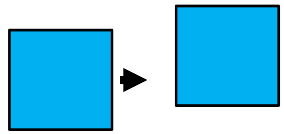
$$\begin{bmatrix}x'\\ y'\\ 1\end{bmatrix} = \begin{bmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix} \cdot \begin{bmatrix}x\\ y\\ 1\end{bmatrix}$$

Shearing



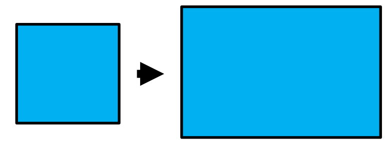
2D Transformations

Translation



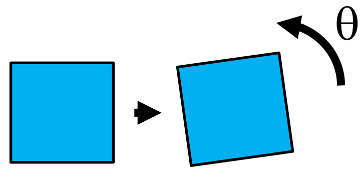
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Scaling



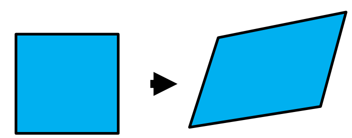
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rotation



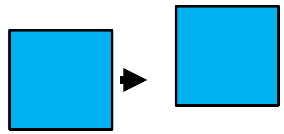
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Shearing



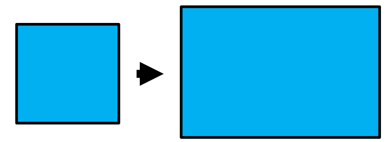
2D Transformations

Translation



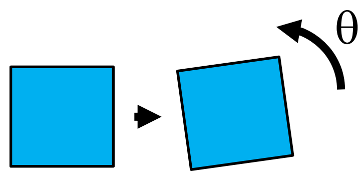
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Scaling



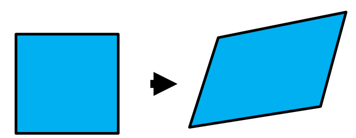
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rotation



$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

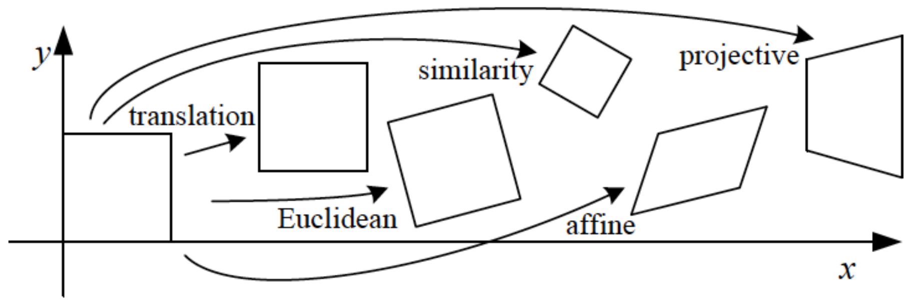
Shearing



$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2D Transformations

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & a_x & 0 \\ a_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



Euclidean = rotation and translation

Similarity = rotation, translation and uniform scaling ($s_x = s_y$)

Affine = rotation, translation, uniform scaling ($s_x = s_y$), and shearing

2D Transformations

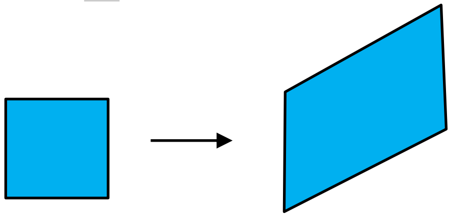
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & a_x & 0 \\ a_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Affine = rotation, translation, shearing and uniform scaling ($s_x = s_y$)

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

Properties:

- 6 degrees of freedom
- Parallel lines remain parallel



Homogeneous coordinates

2D

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

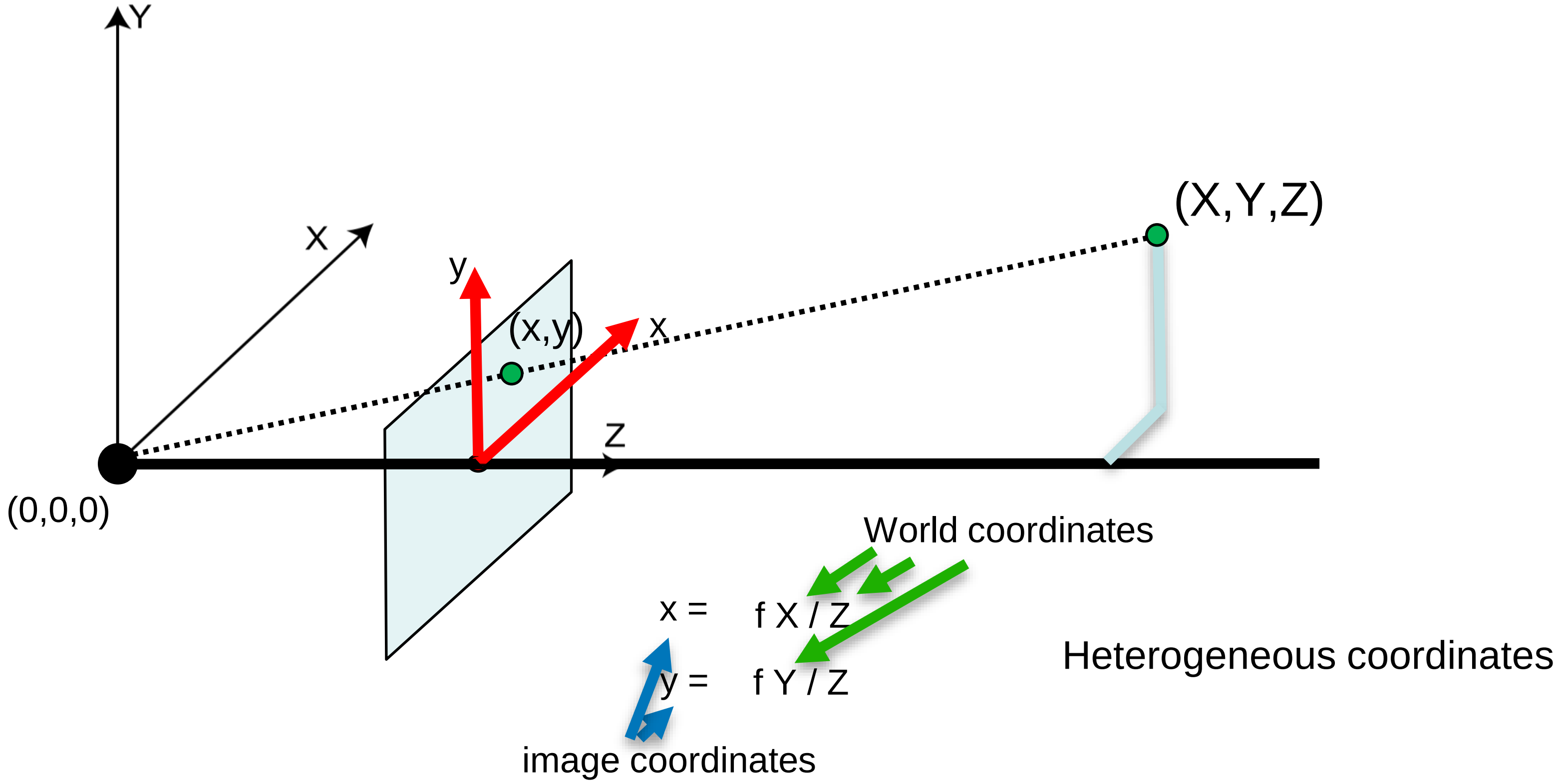
$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

3D

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

Perspective projection



Perspective projection

Heterogeneous coordinates

World coordinates

$$x = f X / Z$$

$$y = f Y / Z$$

image coordinates

Homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \overset{?}{\longleftrightarrow} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Perspective projection

Heterogeneous coordinates

world coordinates

$x = f X / Z$

$y = f Y / Z$

image coordinates

Homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Perspective projection

Heterogeneous coordinates

World coordinates

$$x = f X / Z$$

$$y = f Y / Z$$

image coordinates

Homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Perspective projection

Heterogeneous coordinates

World coordinates

$x = \quad f X / Z$

$y = \quad f Y / Z$

image coordinates

Homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z/f \end{bmatrix}$$

Perspective projection

Heterogeneous coordinates

World coordinates

$x = \quad f\,X / Z$

$y = \quad f\,Y / Z$

image coordinates

Homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z/f \end{bmatrix}$$

Going back to heterogeneous coordinates:

$$\begin{bmatrix} X \\ Y \\ Z/f \end{bmatrix} \longrightarrow (f\,X/Z,\ f\,Y/Z)$$

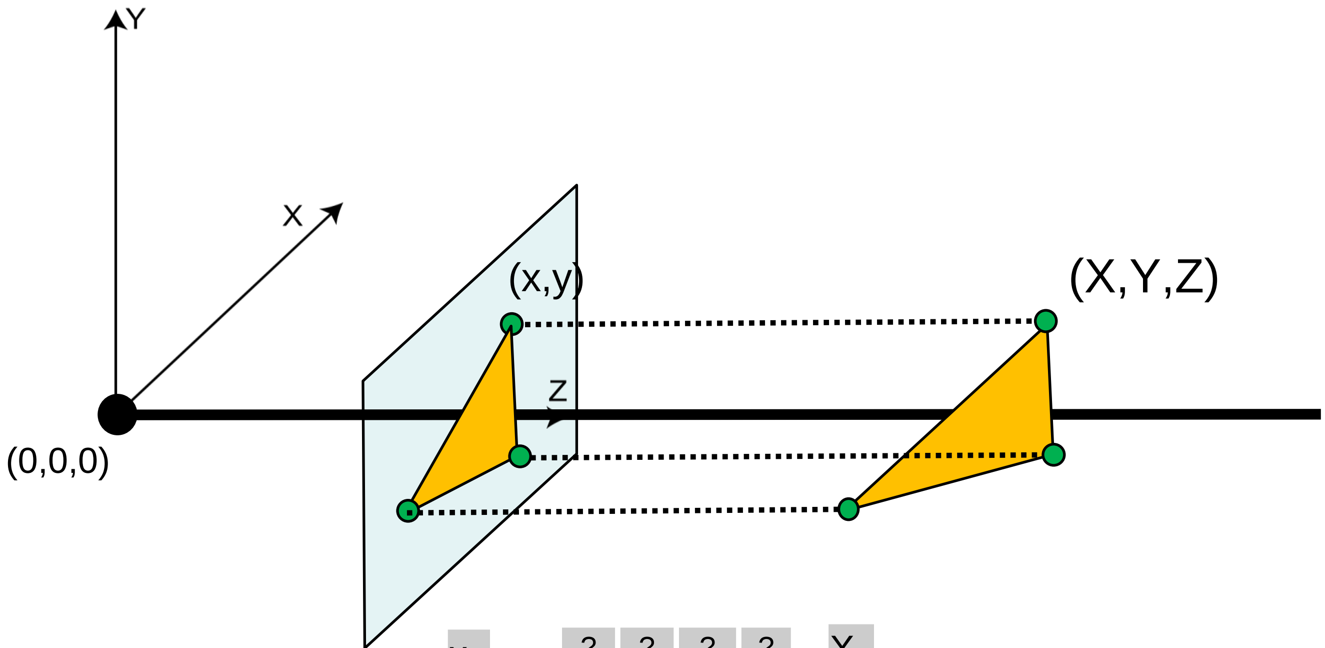
Perspective projection

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z/f \end{bmatrix}$$

(f X/Z, f Y/Z)

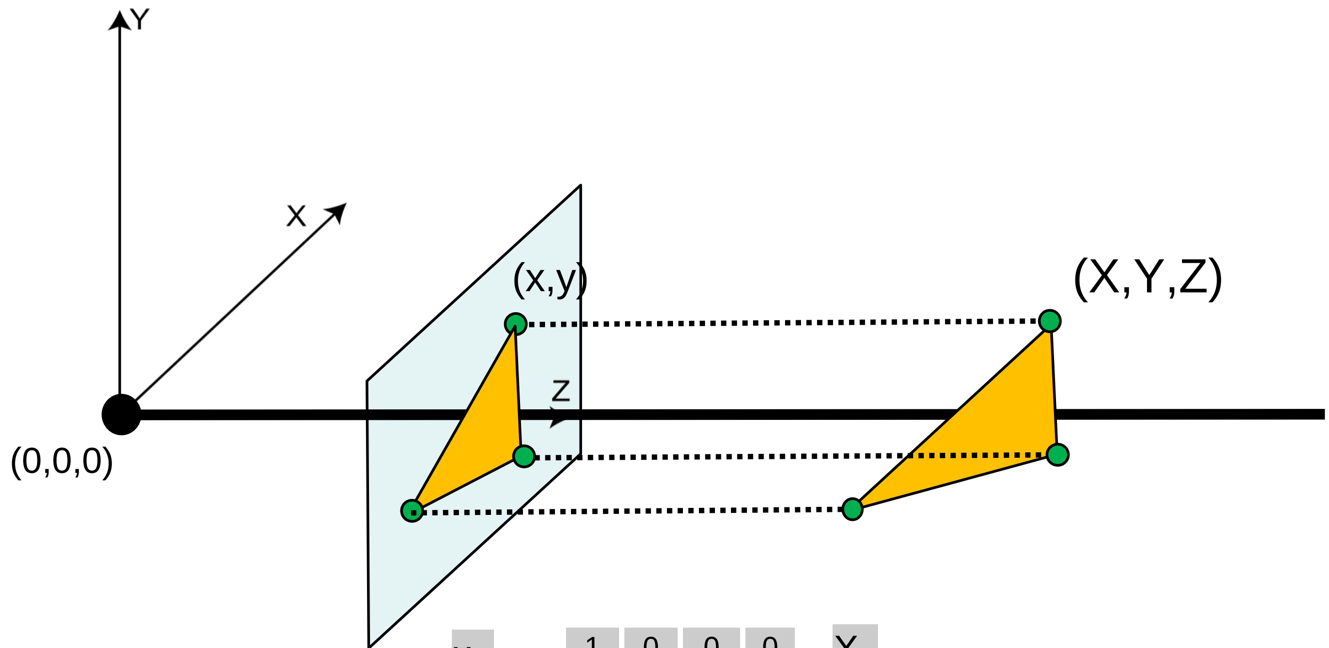
$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix}$$

Orthographic (parallel) projection



$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Orthographic (parallel) projection

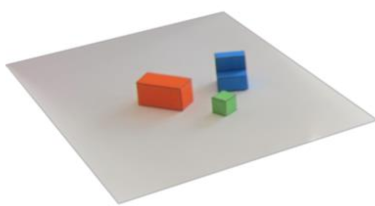


$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Camera parameters

3D world

(X, Y, Z)



?

(x, y)

2D image

$$\underbrace{\begin{bmatrix} x \\ y \\ w \end{bmatrix}}_{\text{Pixels}} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix} \cdot \underbrace{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}_{\text{World coordinates}}$$

Perspective Projection Matrix

- Projection is a matrix multiplication using homogeneous coordinates

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f \end{bmatrix} \Rightarrow \left(f\frac{x}{z}, f\frac{y}{z}\right)$$

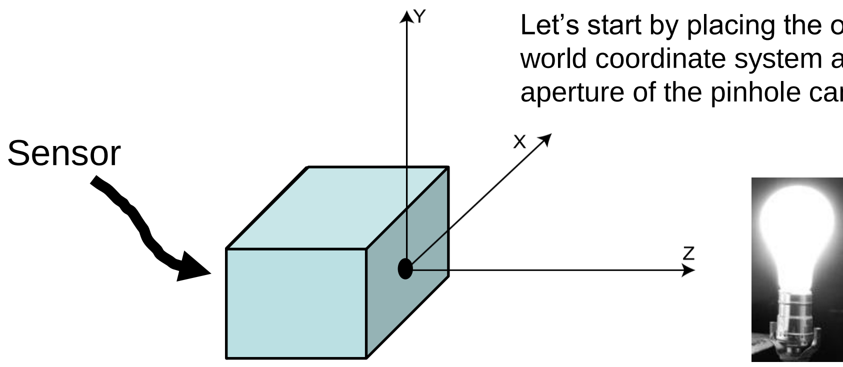
divide by the third coordinate

In practice: lots of coordinate transformations…

$$\begin{pmatrix} \text{2D} \\ \text{point} \\ \text{(3x1)} \end{pmatrix} = \begin{pmatrix} \text{Camera to} \\ \text{pixel coord.} \\ \text{trans. matrix} \\ \text{(3x3)} \end{pmatrix} \begin{pmatrix} \text{Perspective} \\ \text{projection matrix} \\ \text{(3x4)} \end{pmatrix} \begin{pmatrix} \text{World to} \\ \text{camera coord.} \\ \text{trans. matrix} \\ \text{(4x4)} \end{pmatrix} \begin{pmatrix} \text{3D} \\ \text{point} \\ \text{(4x1)} \end{pmatrix}$$

Camera parameters

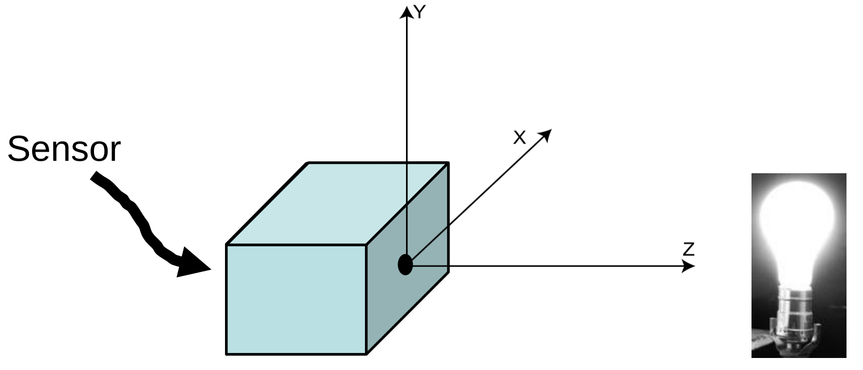
Let’s start by placing the origin of the world coordinate system at the aperture of the pinhole camera



For the pinhole camera:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} \longrightarrow (f\,X/Z,\ f\,Y/Z)$$

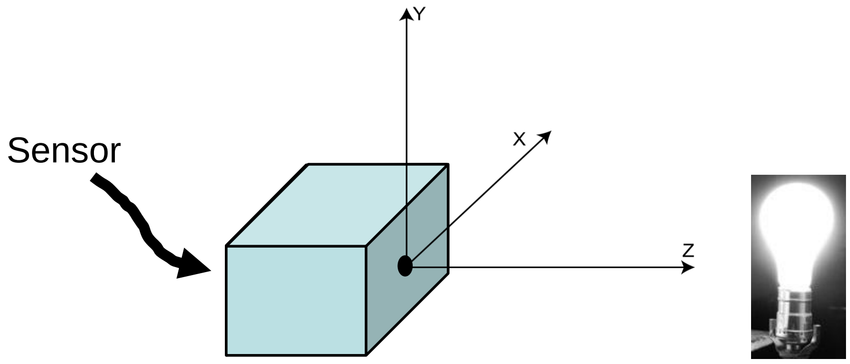
Camera parameters



When changing to pixels, there will be an arbitrary scaling:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & a & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} aX \\ aY \\ Z \end{bmatrix} \longrightarrow (a\ X/Z,\ a\ Y/Z)$$

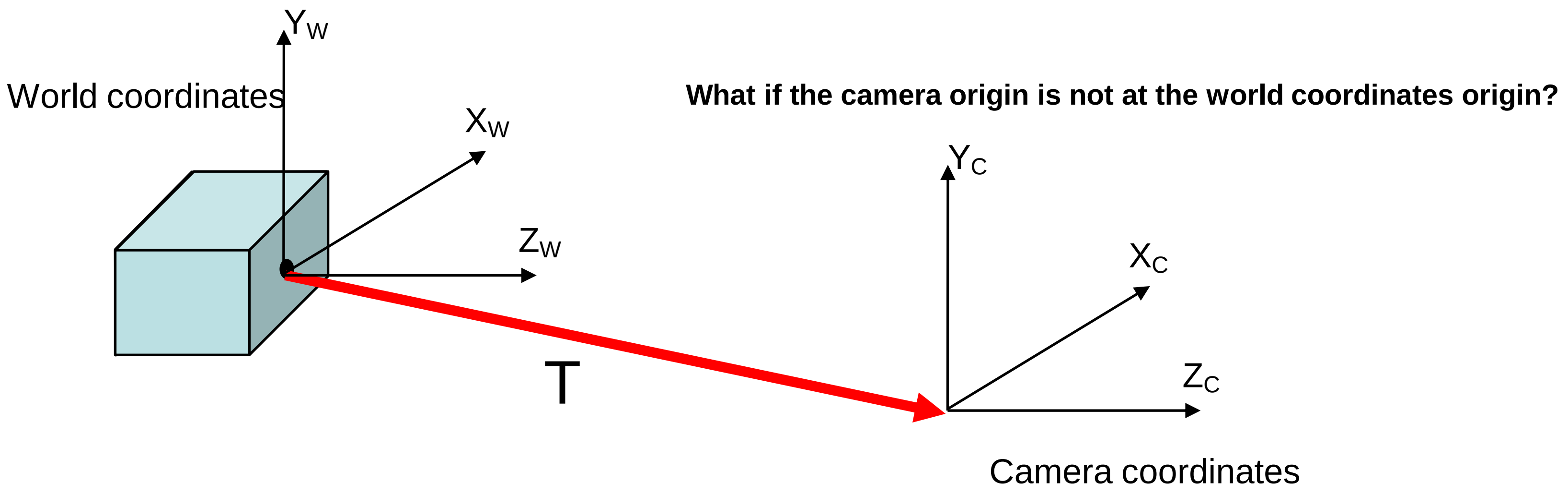
Camera parameters



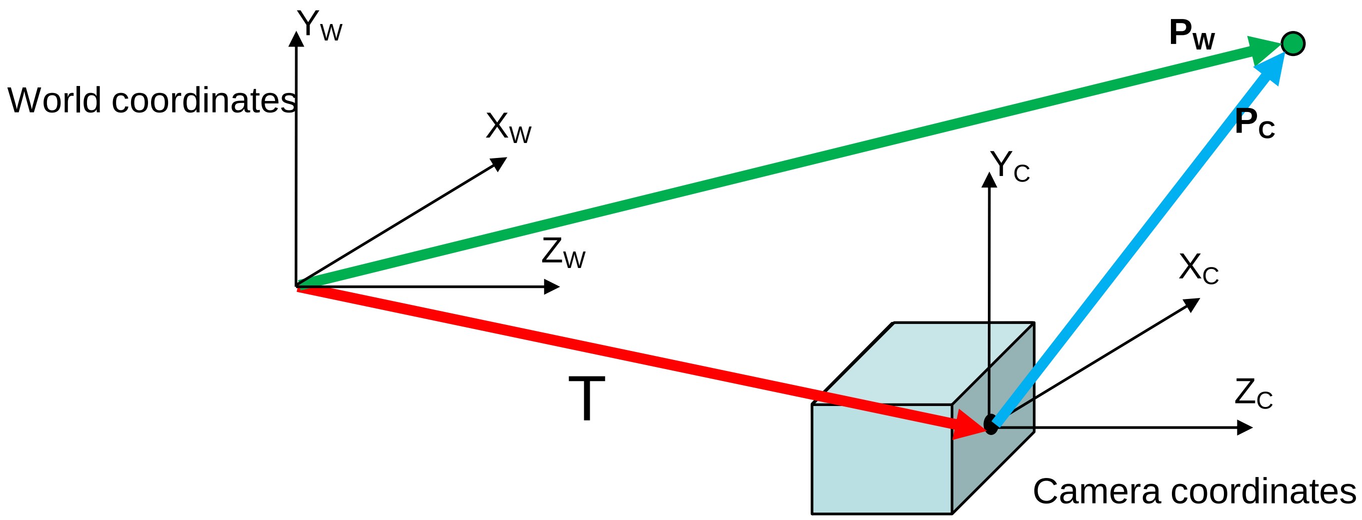
if pixels are rectangular

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} aX \\ bY \\ Z \end{bmatrix} \longrightarrow (a\ X/Z,\ b\ Y/Z)$$

Camera parameters



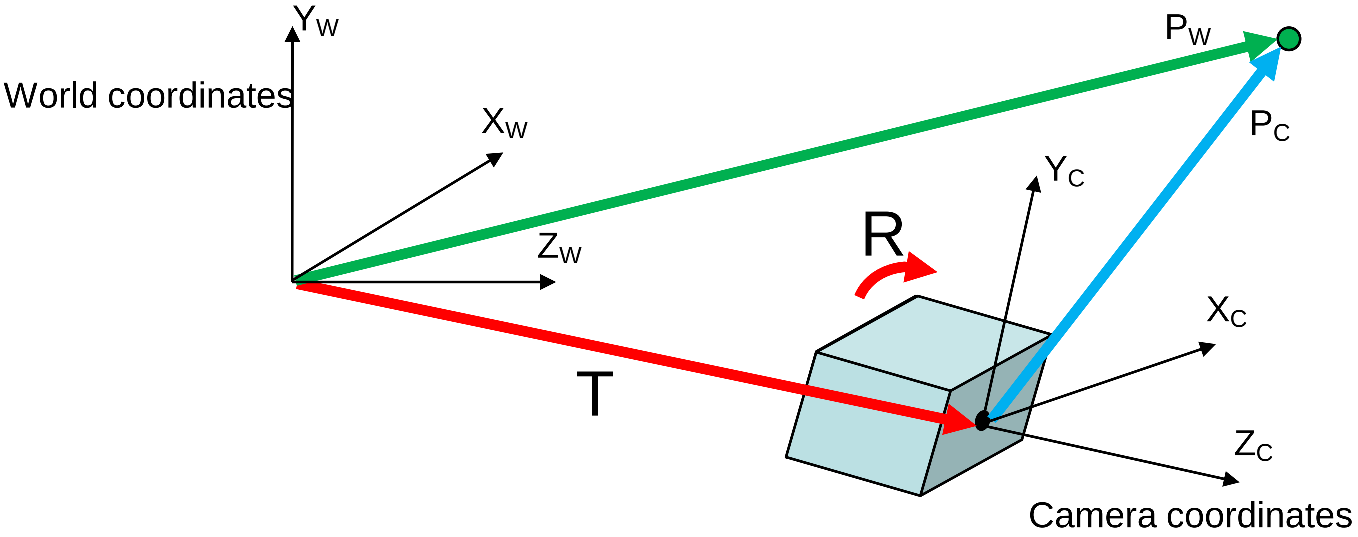
Camera parameters



In heterogeneous coordinates:

$$P_C = P_W - T$$

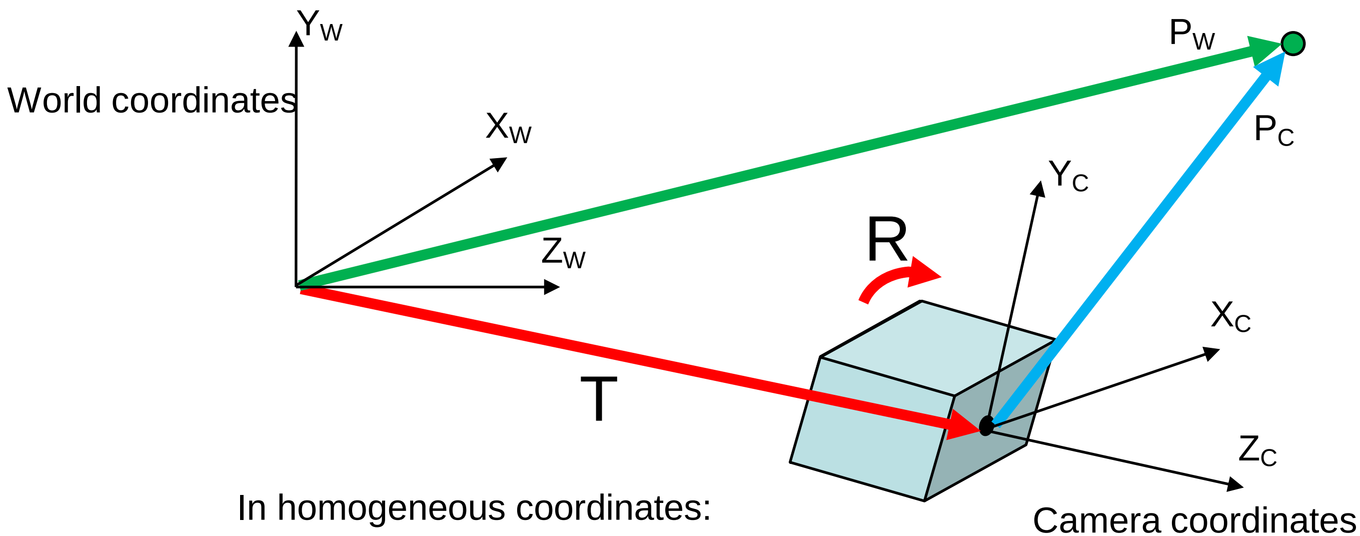
Camera parameters



In heterogeneous coordinates:

$$P_C = R(P_W - T)$$

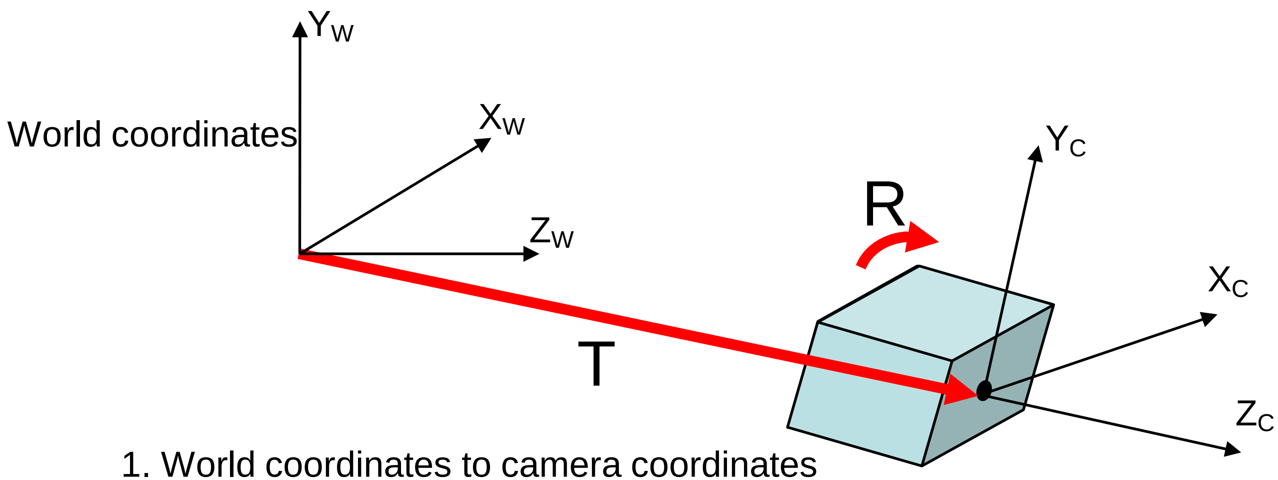
Camera parameters



In homogeneous coordinates:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \overset{[3\text{x}3]}{\mathbf{R}} & \overset{[3\text{x}1]}{\mathbf{-RT}} \\ \underset{[1\text{x}3]}{\mathbf{0}} & \underset{[1\text{x}1]}{1} \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Camera parameters



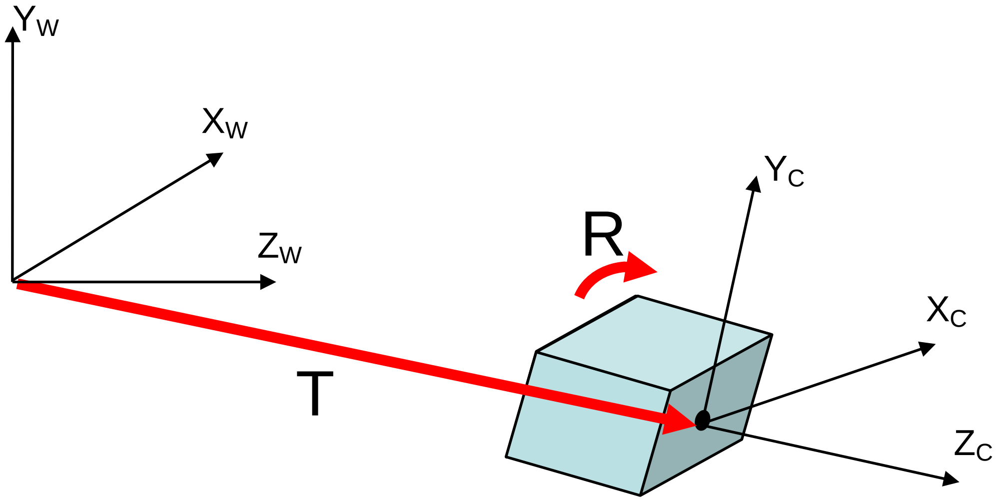
1. World coordinates to camera coordinates

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{-RT} \\ \mathbf{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

2. Camera coordinates to image coordinates (square pixels)

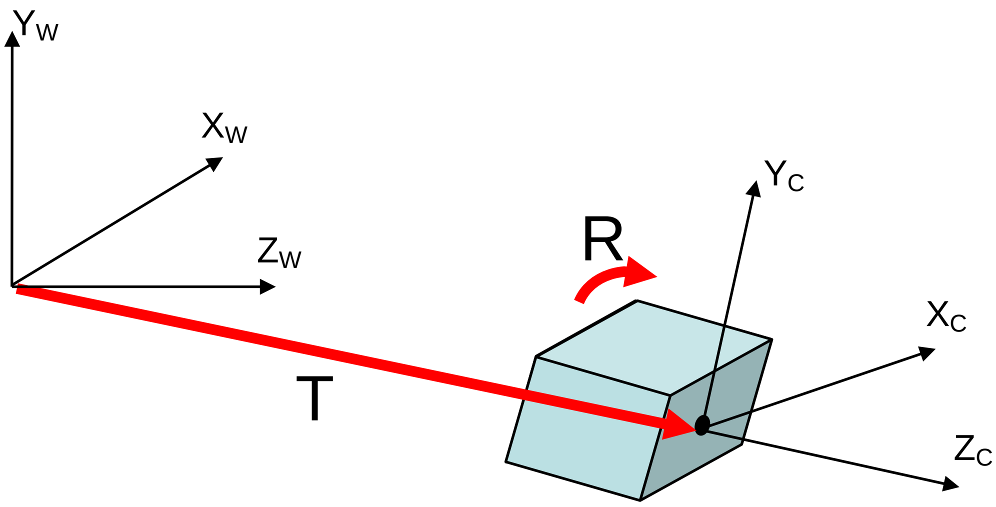
$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & a & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

Camera parameters



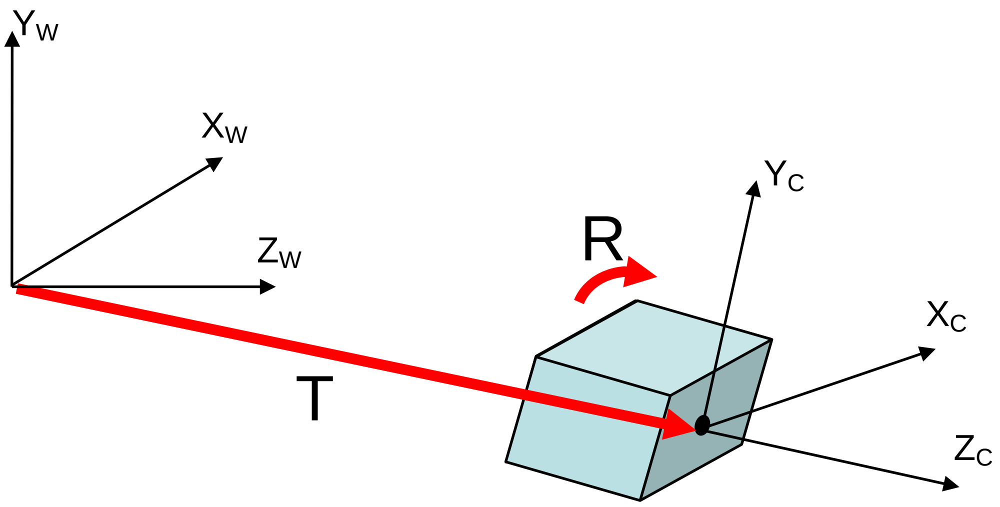
$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Camera parameters



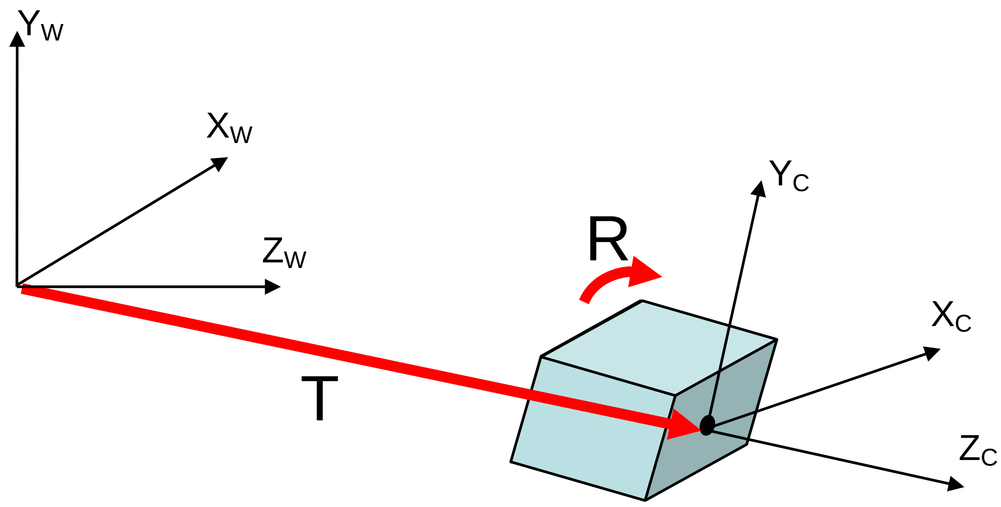
$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \underbrace{\begin{bmatrix} a & 0 & 0 & \cancel{0} \\ 0 & a & 0 & \cancel{0} \\ 0 & 0 & 1 & \cancel{0} \end{bmatrix}}_{[3x4]} \cdot \underbrace{\begin{bmatrix} \mathbf{R} & \mathbf{-RT} \\ \cancel{\mathbf{0}} & \cancel{1} \end{bmatrix}}_{[4x4]} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Camera parameters



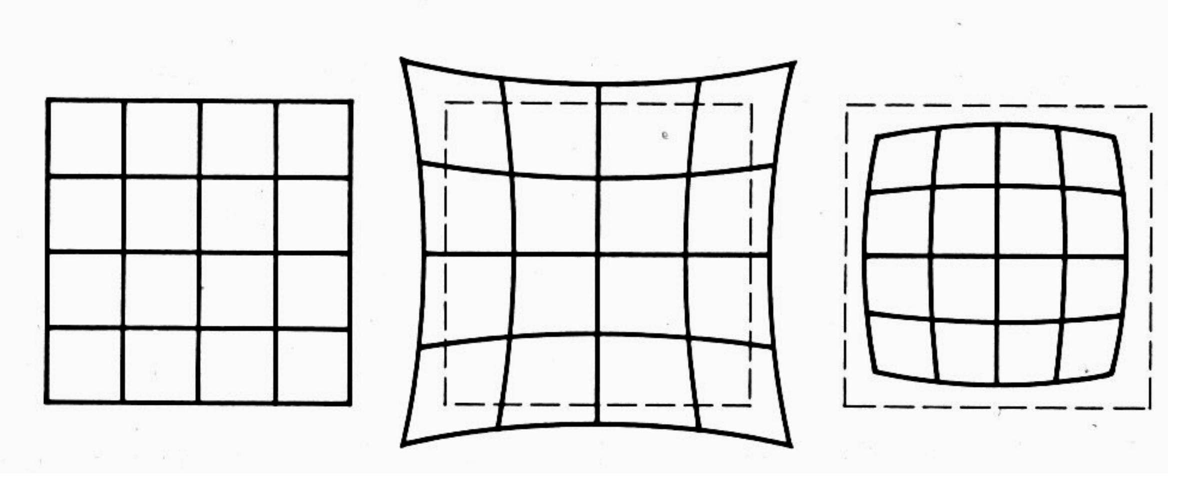
$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \overset{[3x3]}{\begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 1 \end{bmatrix}} \cdot \overset{[3x4]}{\begin{bmatrix} \mathbf{R} & \mathbf{-RT} \end{bmatrix}} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Camera parameters



$$\begin{bmatrix}x\\y\\w\end{bmatrix} = \underbrace{\overset{[3x3]}{\begin{bmatrix}a&0&0\\0&a&0\\0&0&1\end{bmatrix}}}_{\text{Intrinsic parameters}} \cdot \underbrace{\overset{[3x3]}{\mathbf{R}} \cdot \overset{[3x4]}{\begin{bmatrix}\mathbf{I} & -\mathbf{T}\end{bmatrix}}}_{\text{Extrinsic parameters}} \cdot \begin{bmatrix}X_w\\Y_w\\Z_w\\1\end{bmatrix}$$

Radial Distortion



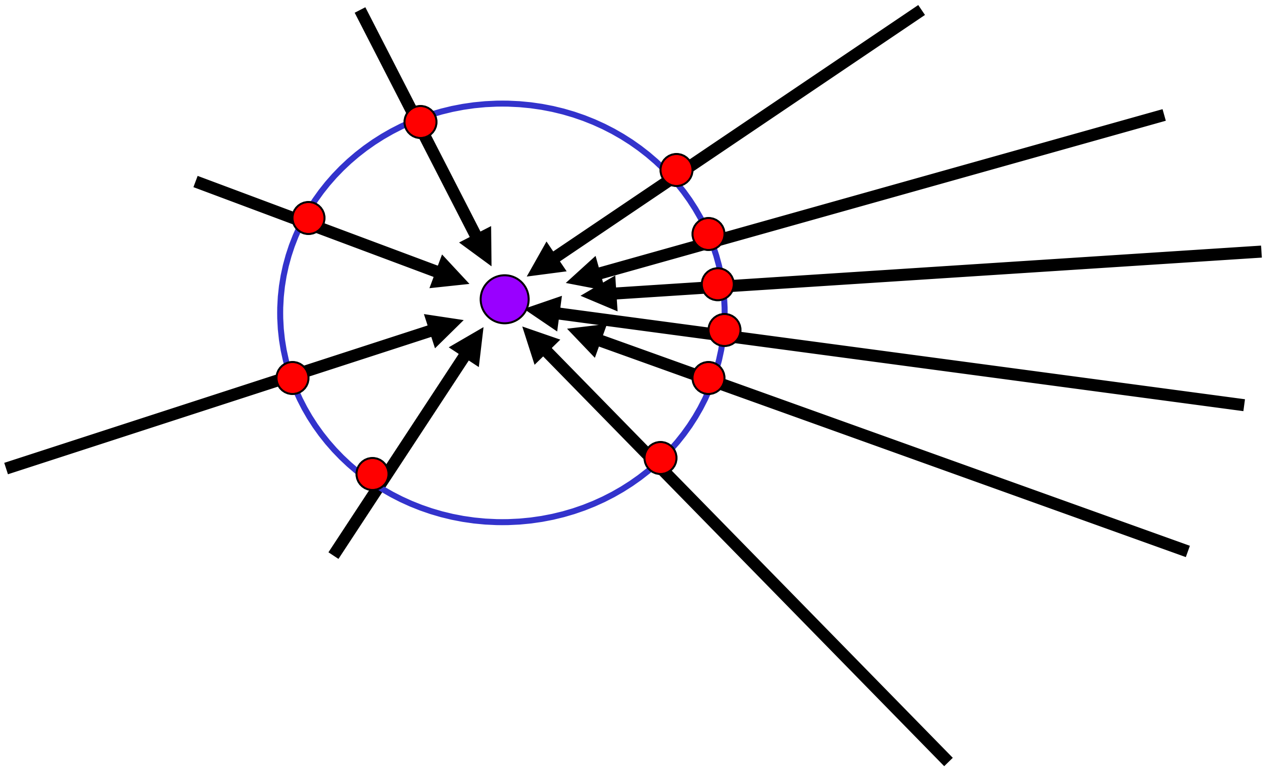
No distortion Pin cushion Barrel

- Radial distortion of the image
  - Caused by imperfect lenses
  - Deviations are most noticeable for rays that pass through the edge of the lens

Correcting radial distortion

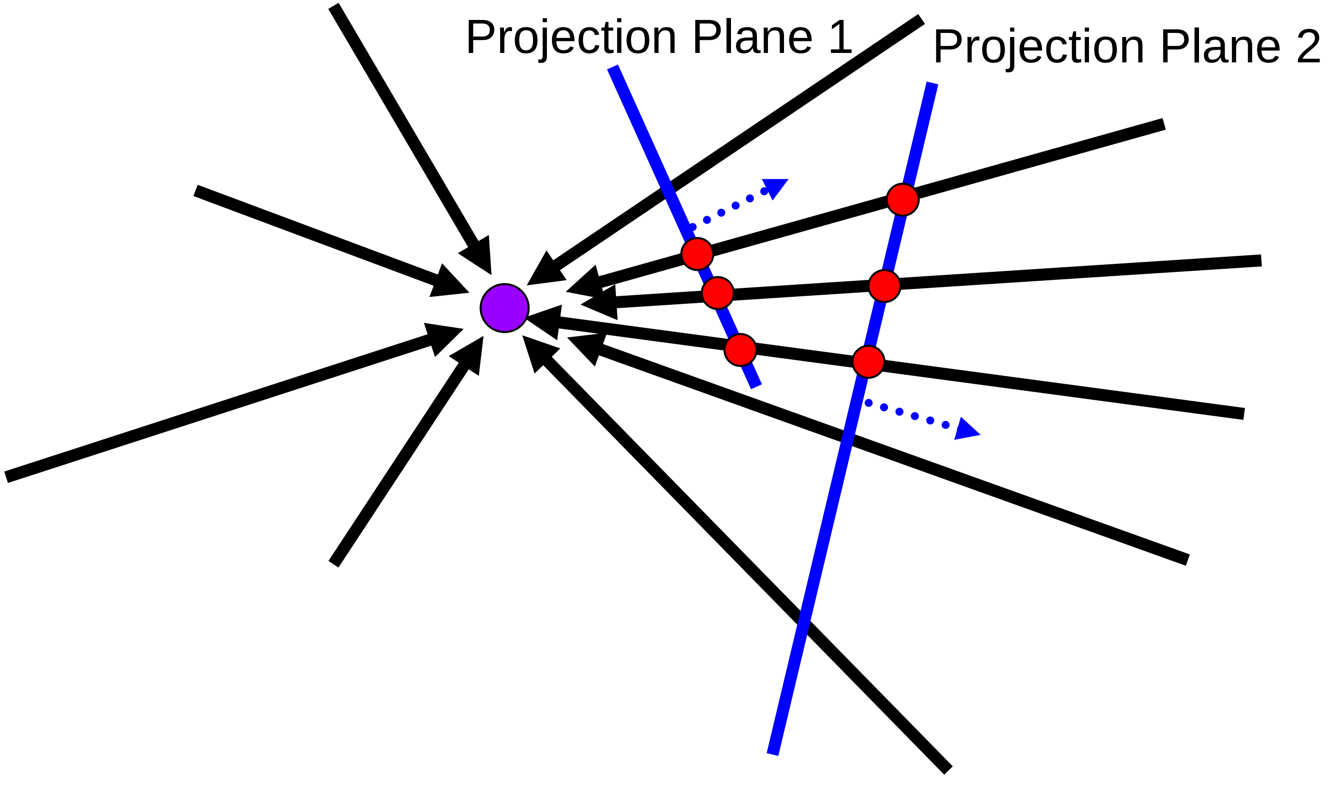
from Helmut Dersch

A pencil of rays contains all views



http://www.panoramas.dk/new-year-2005/FS1_times-square.html

Nothing special about a given projection



So, we can generate any synthetic camera view
as long as it has **the same center of projection**!

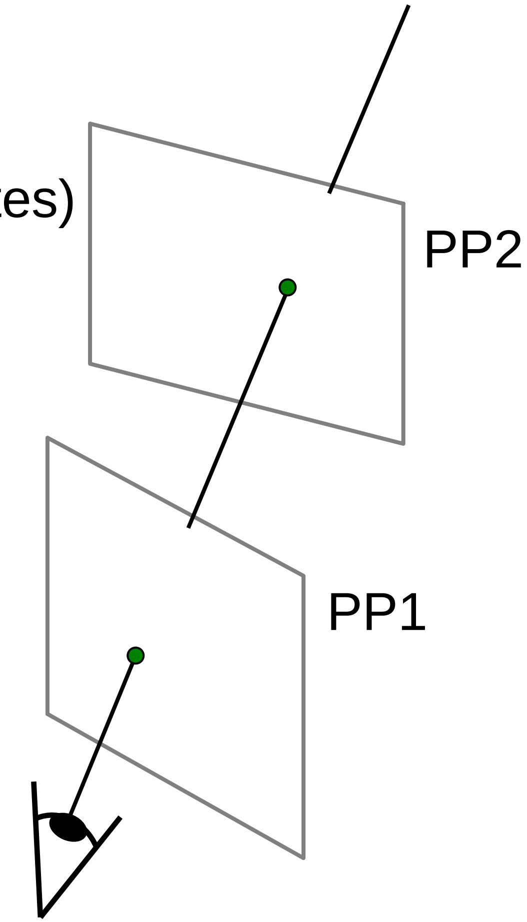
Homography

A projective mapping between any two PPs with the same center of projection

- rectangle should map to arbitrary quadrilateral
- Parallelism not preserved
- but must preserve straight lines
- Maps 2D to 2D (but in homogenous coordinates)

called Homography

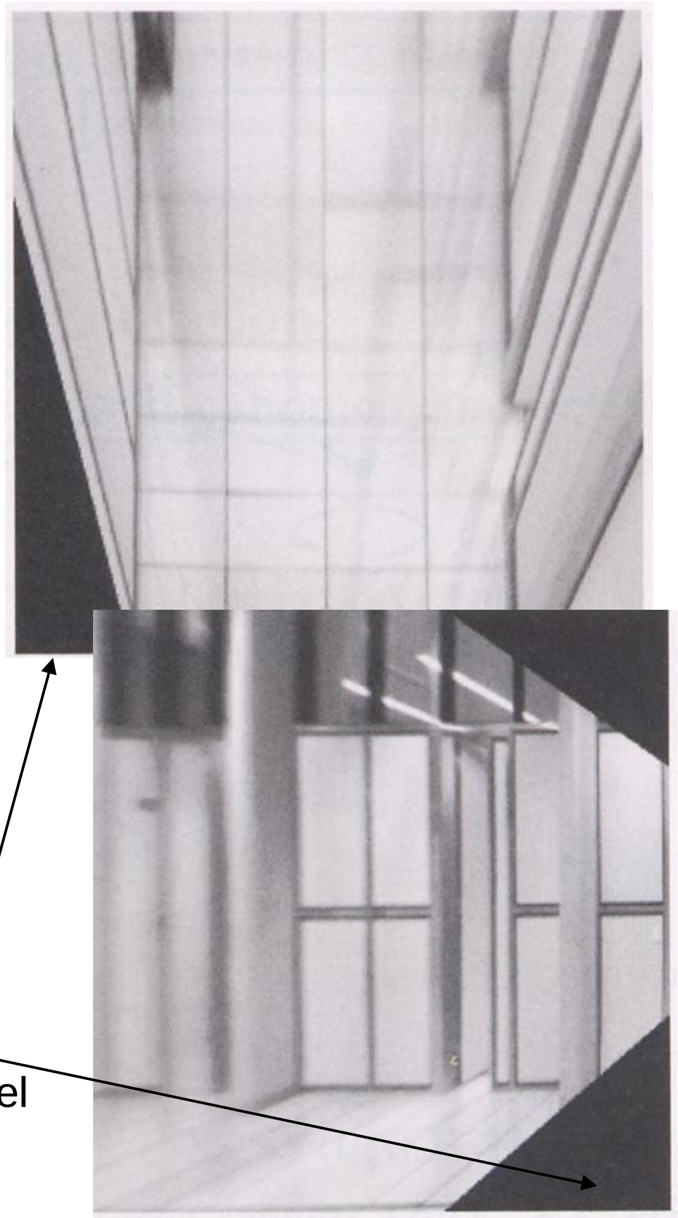
$$\underset{\mathbf{p'}}{\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}} = \underset{\mathbf{H}}{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}} \underset{\mathbf{p}}{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}$$



To apply a homography **H**

- Compute **p’** = **Hp** (regular matrix multiply)
- Convert **p’** from homogeneous to image coordinates

Generating new views with same COP



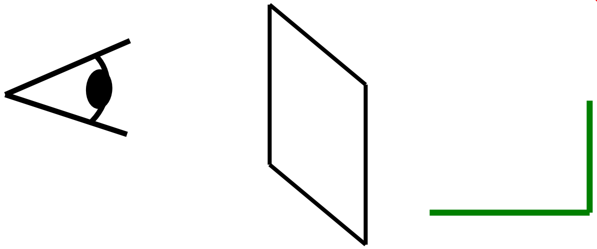
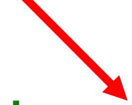
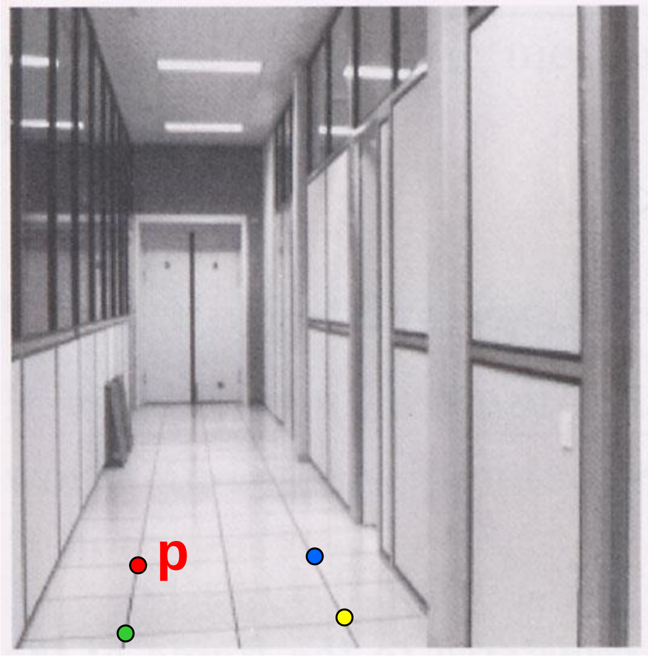


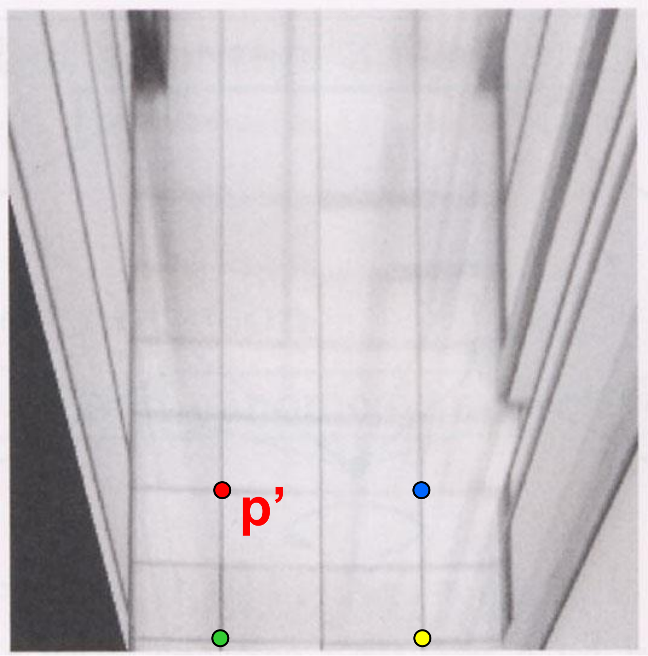
image plane in front



black area
where no pixel
maps to

Image rectification





To unwarp (rectify) an image

- Find the homography **H** given a set of **p** and **p’** pairs
- How many correspondences are needed?

Solving for homographies

p' = **Hp**

$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Can set scale factor *i*=1. So, there are 8 unkowns.

Set up a system of linear equations:

Ah = b

where vector of unknowns h = [a,b,c,d,e,f,g,h]^T

Need at least 8 eqs, but the more the better…

Solve for h. Can be done in Matlab using “\” command

- see “help lmdivide”

If overconstrained, can solve using least-squares:

$$\min \|Ah - b\|^2$$

Fun with homographies

Original image

St.Petersburg
photo by A. Tikhonov

Virtual camera rotations

Practicalities of applying transforms

Representing Images and Geometry 569

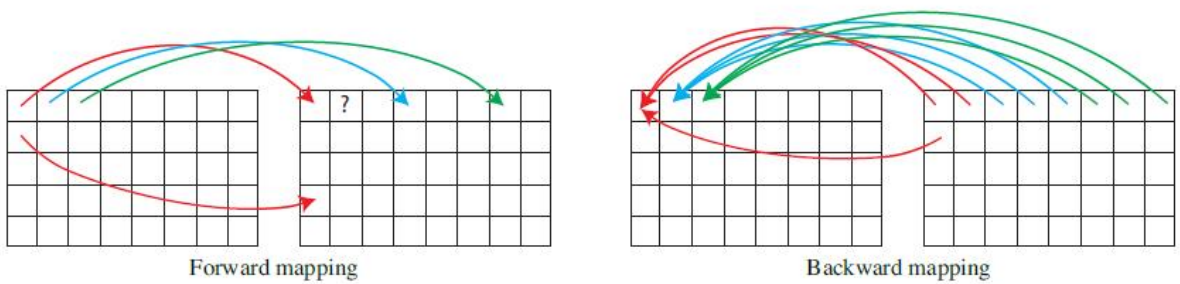


Figure 38.10: Comparison between forward and backward mapping using nearest neighbor interpolation. Forward mapping will produce missing values.

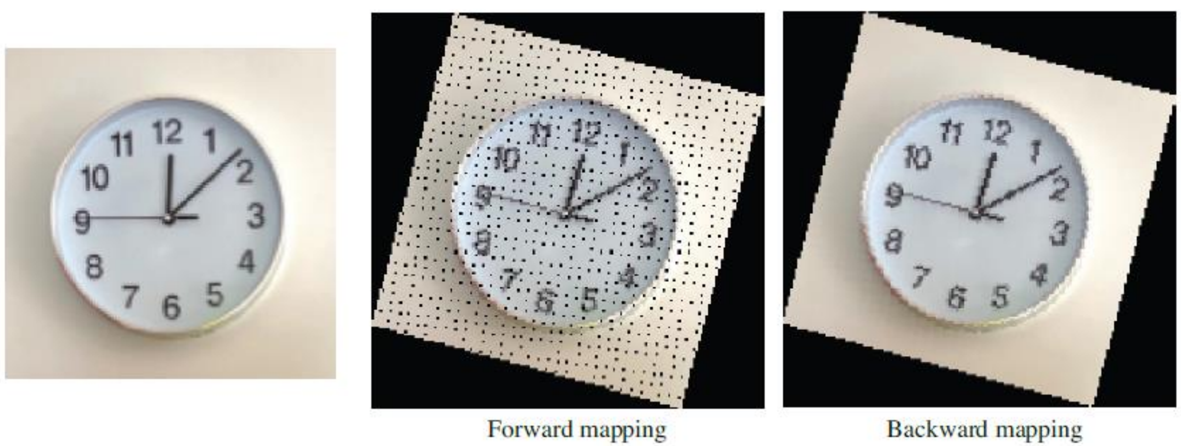
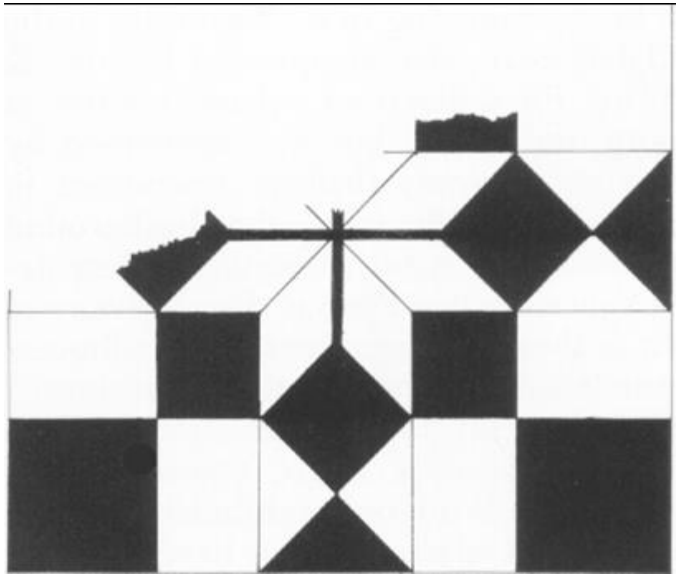


Figure 38.11: Comparison between forward and backward mapping. Forward mapping produces many artifacts. In both cases we use nearest neighbor interpolation.

What is the shape of the b/w floor pattern?

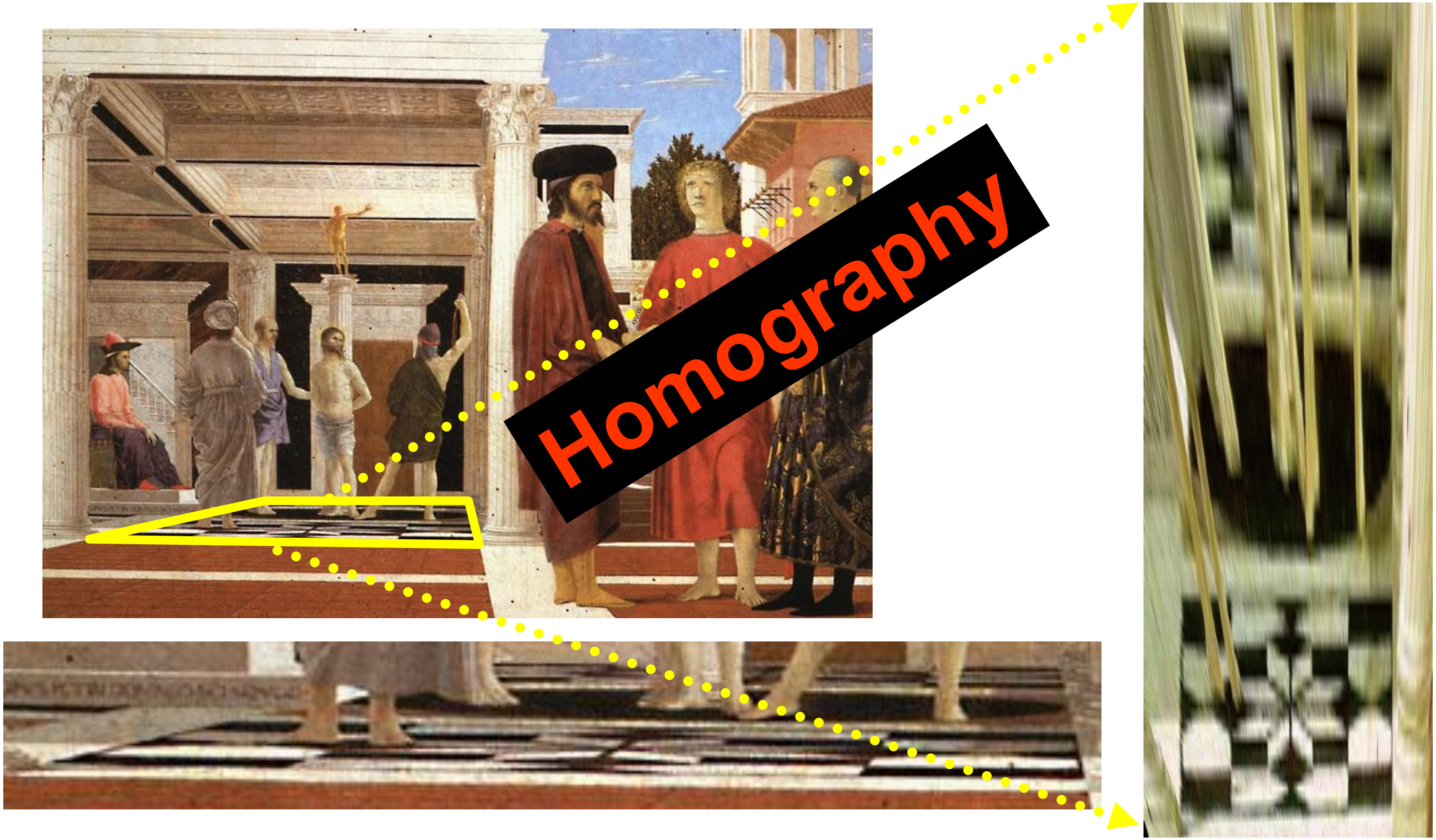


From Martin Kemp *The Science of Art*

The floor (enlarged)

Slide from Criminisi

What is the shape of the b/w floor pattern?



The floor (enlarged)

Automatically rectified floor

Slide from Criminisi

Analysing patterns and shapes

St. Lucy Altarpiece, D. Veneziano

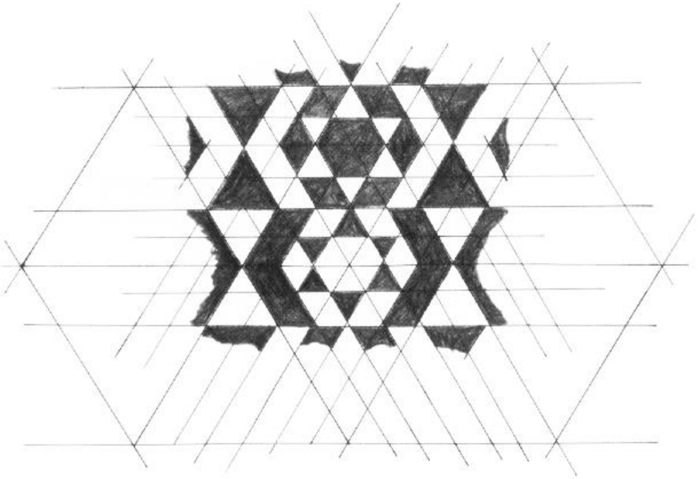
What is the (complicated) shape of the floor pattern?

Automatically rectified floor

Slide from Criminisi

Analysing patterns and shapes

Automatic rectification

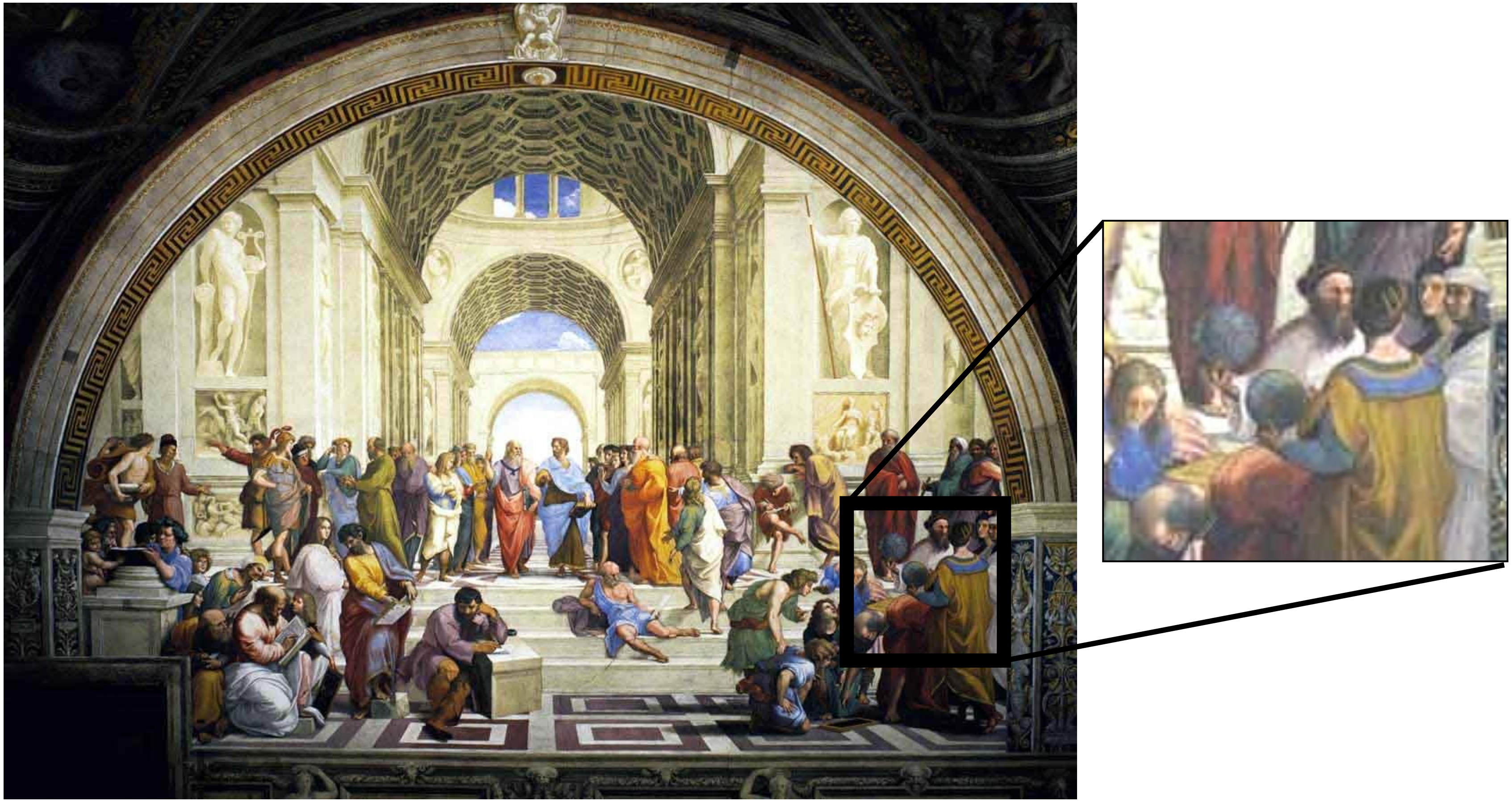


From Martin Kemp, *The Science of Art* *(manual reconstruction)*

Slide from Criminisi

Holbein, *The Ambassadors*

Personalized projections

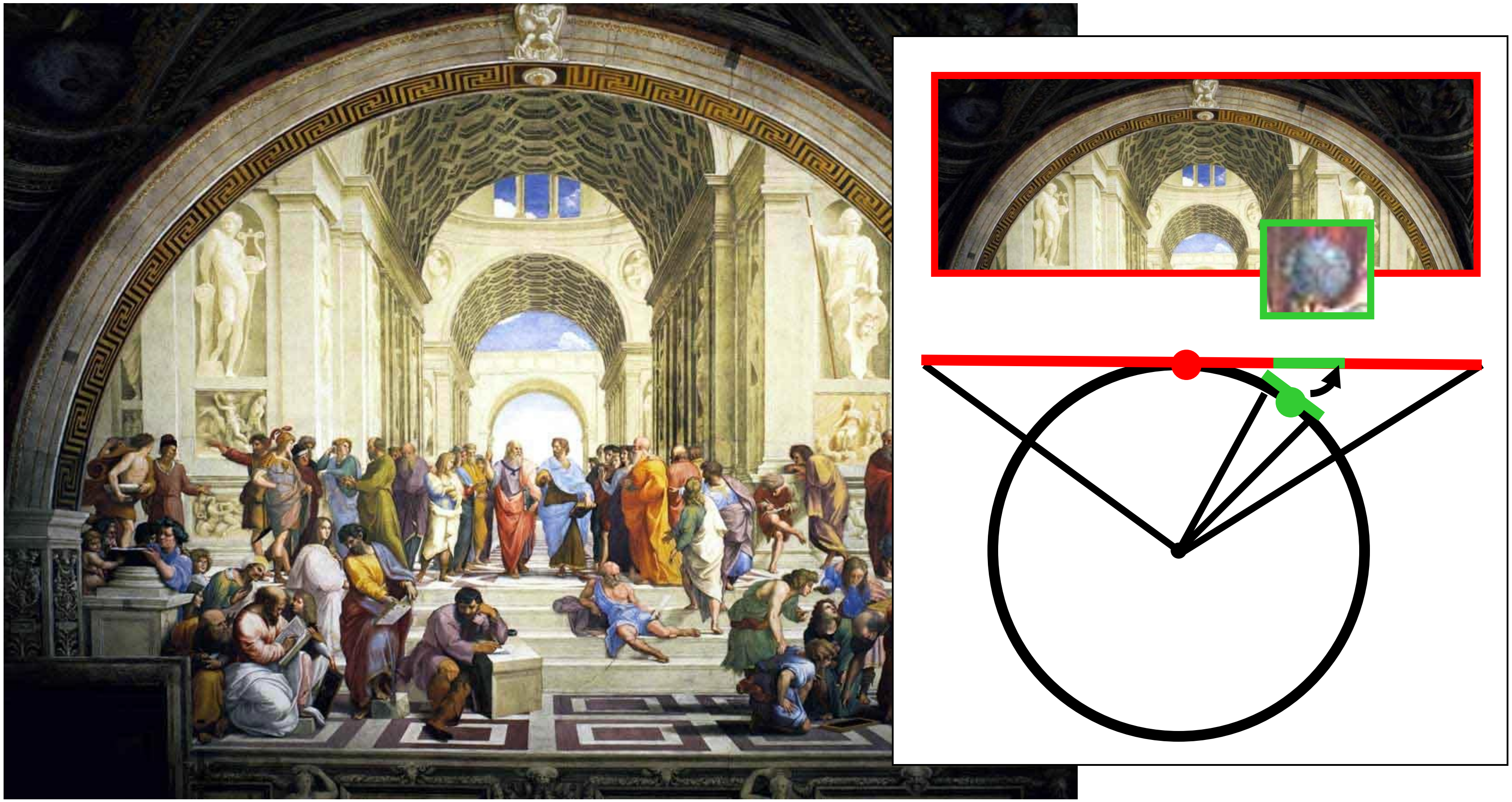


“School of Athens”, Raffaello Sanzio ~1510

Give a separate treatment to different parts of the scene

Slide by Lihi Zelnik-Manor

Personalized projections



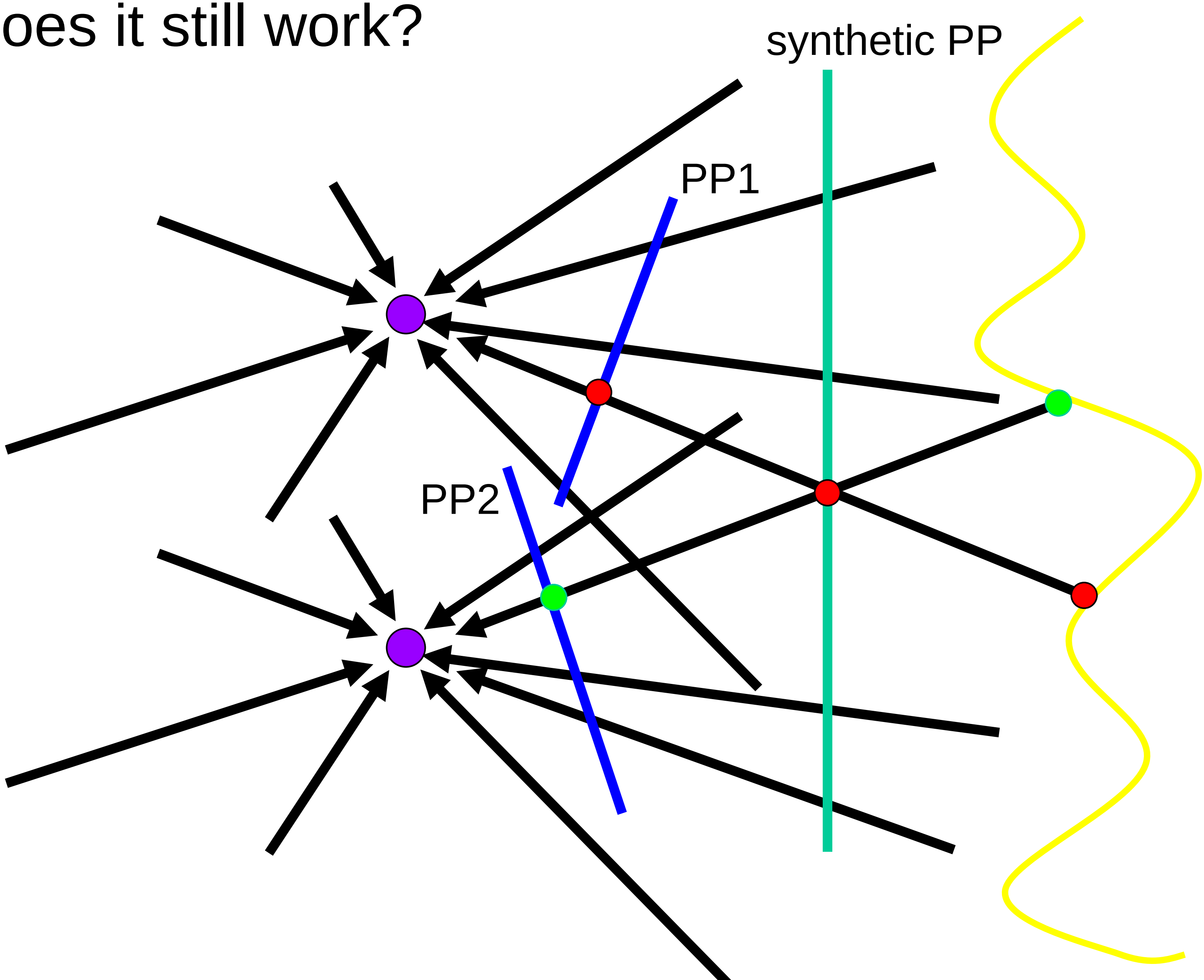
“School of Athens”, Raffaello Sanzio ~1510

Give a separate treatment to different parts of the scene

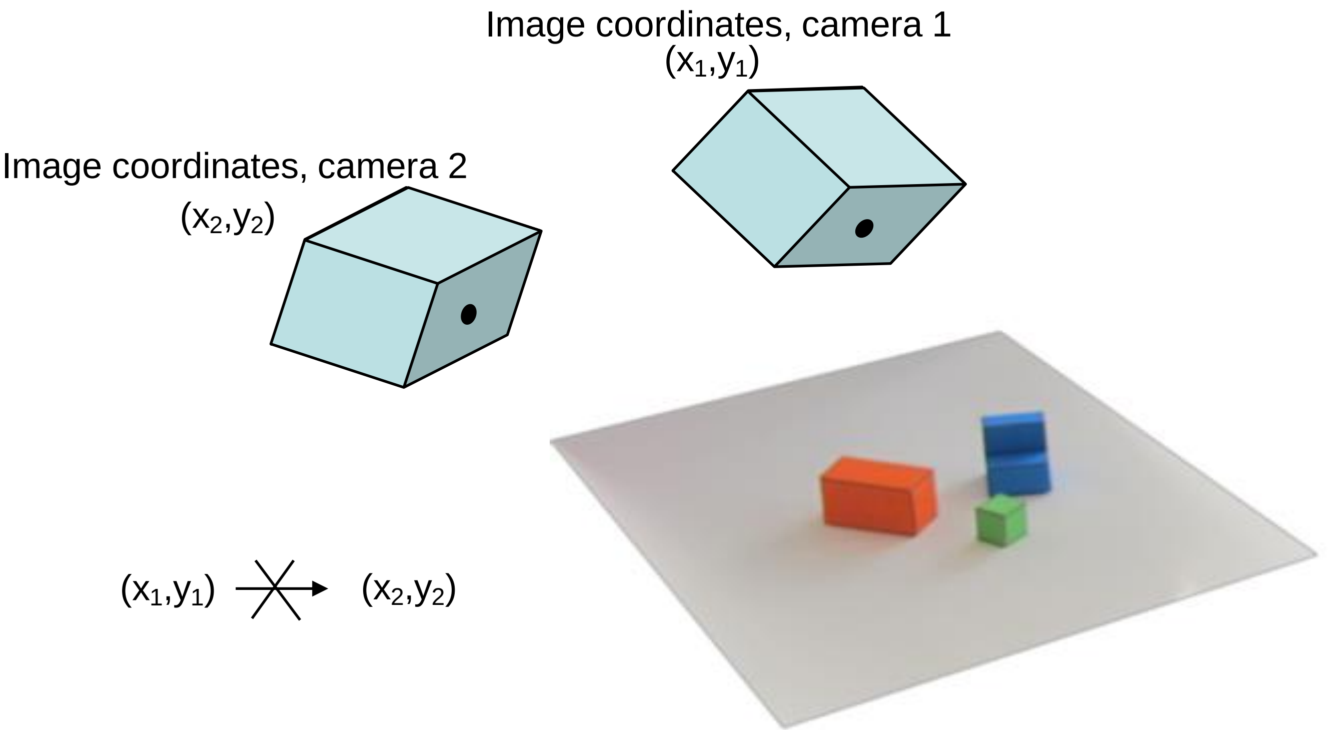
Slide by Lihi Zelnik-Manor

changing camera center

Does it still work?

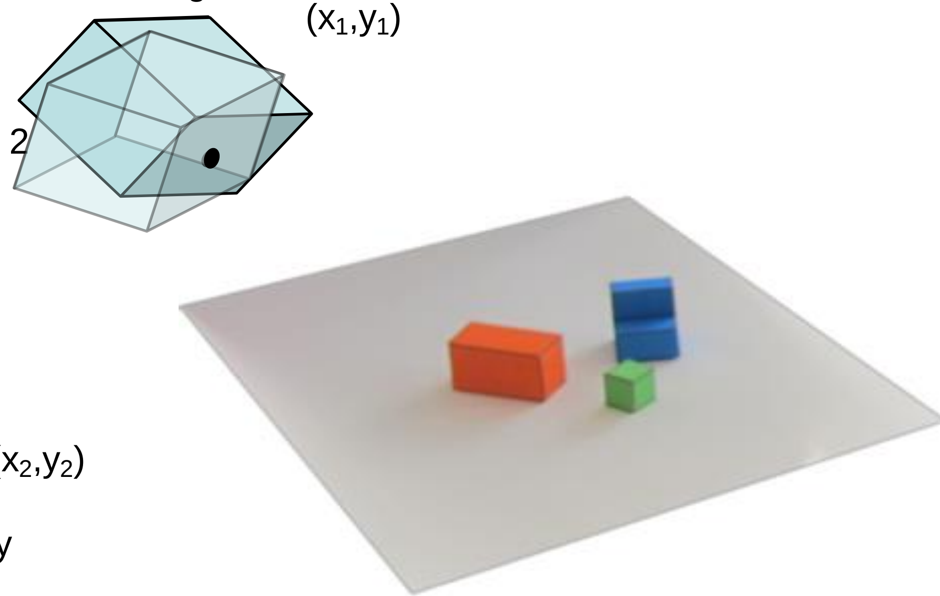


Mapping one camera into another



In general, we can not find a transformation from x_1 to x_2. It requires knowing
the 3D coordinates of each corresponding point.
(The general mapping has to depend on 3D shape, otherwise we would learn no information from the 2nd image of a stereo camera!)

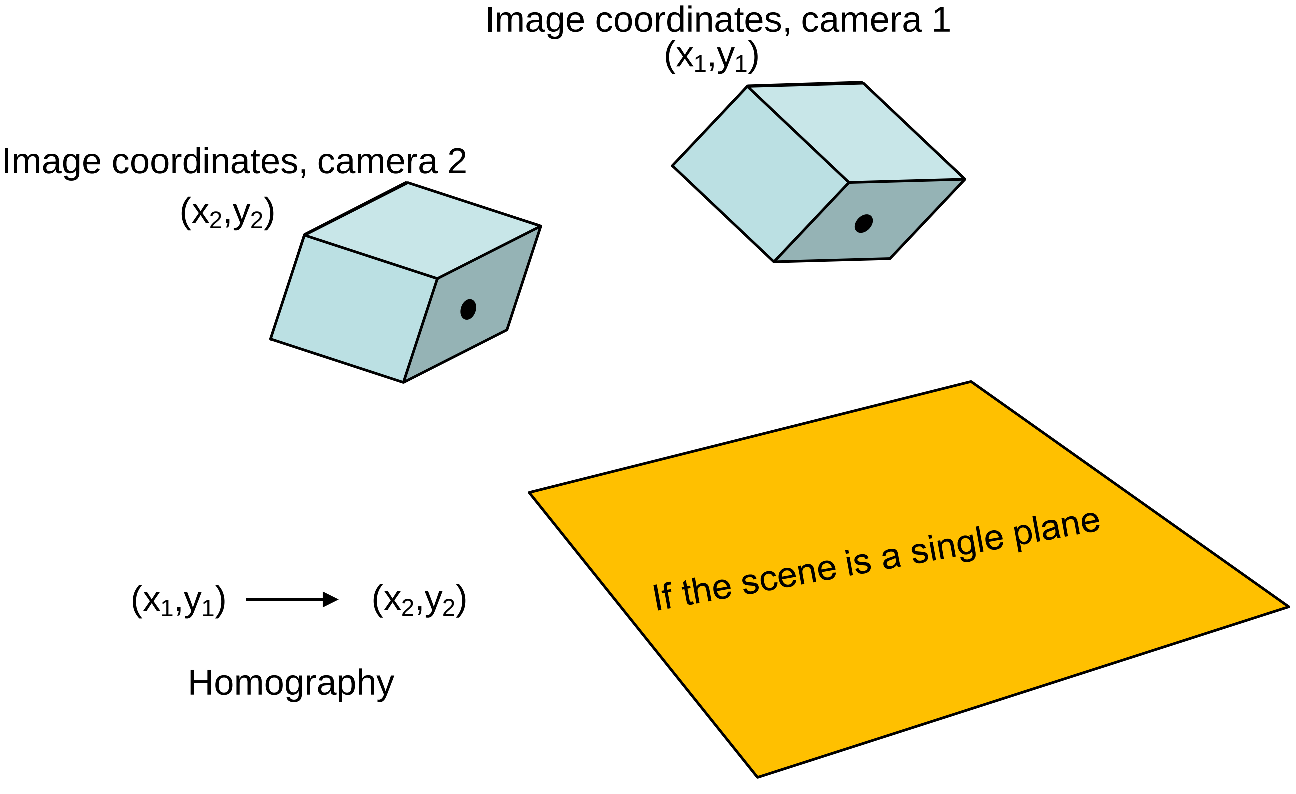
Mapping one camera into another



$(x_1,y_1) \longrightarrow (x_2,y_2)$

Homography

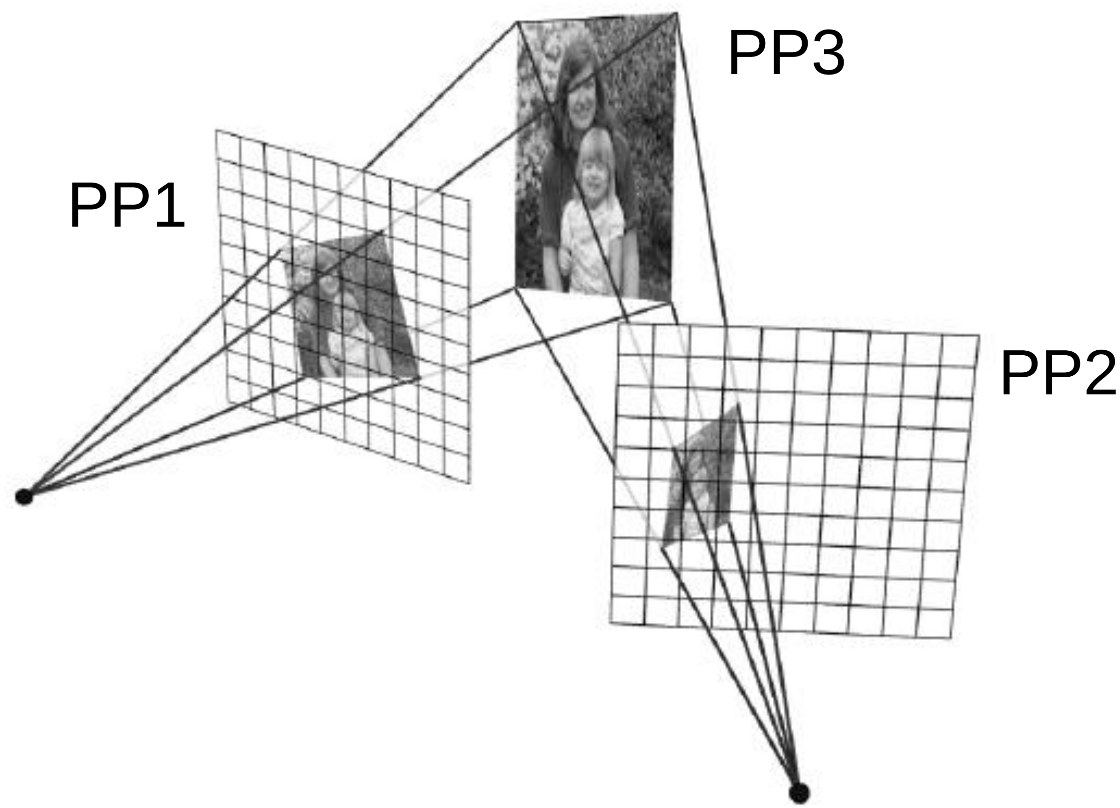
Mapping one camera into another



$(x_1,y_1) \longrightarrow (x_2,y_2)$

Homography

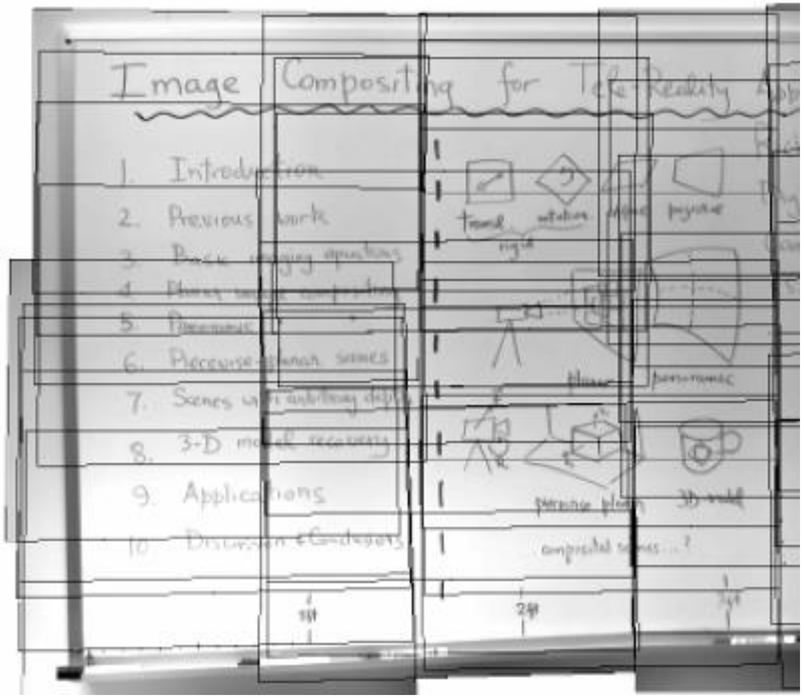
Planar scene (or far away)

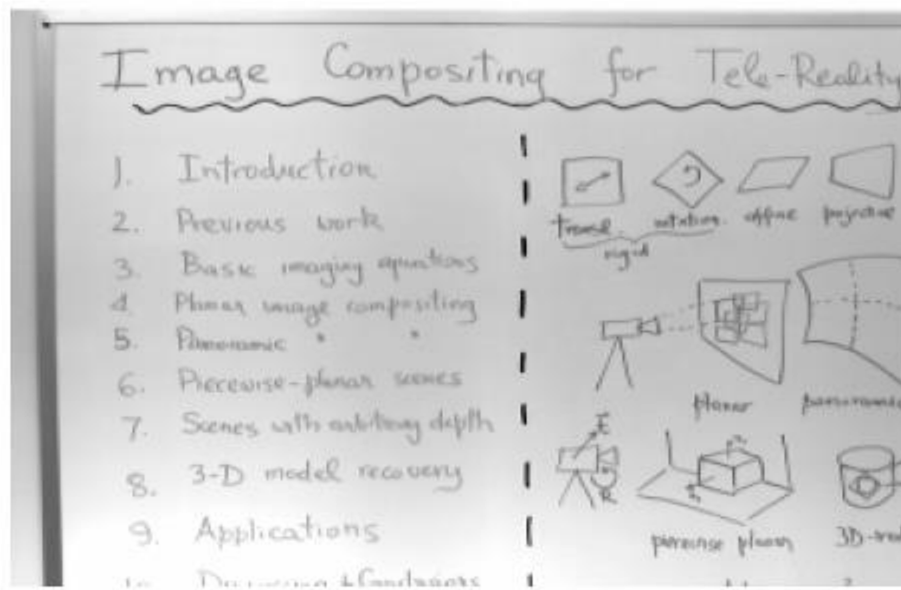


PP3 is a projection plane of both centers of projection, so we are OK!

This is how big aerial photographs are made

Planar mosaic





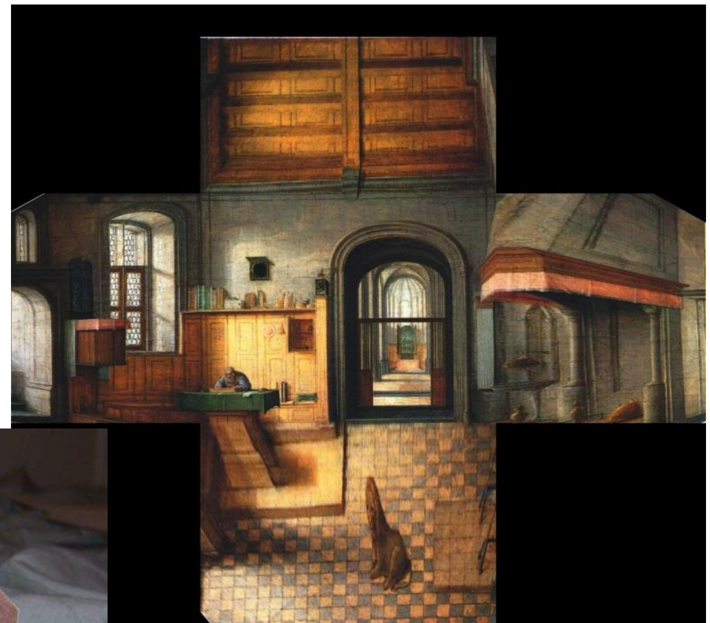
3D reconstruction from a single view

We want real 3D scene walk-throughs:
- Camera rotation
- Camera translation

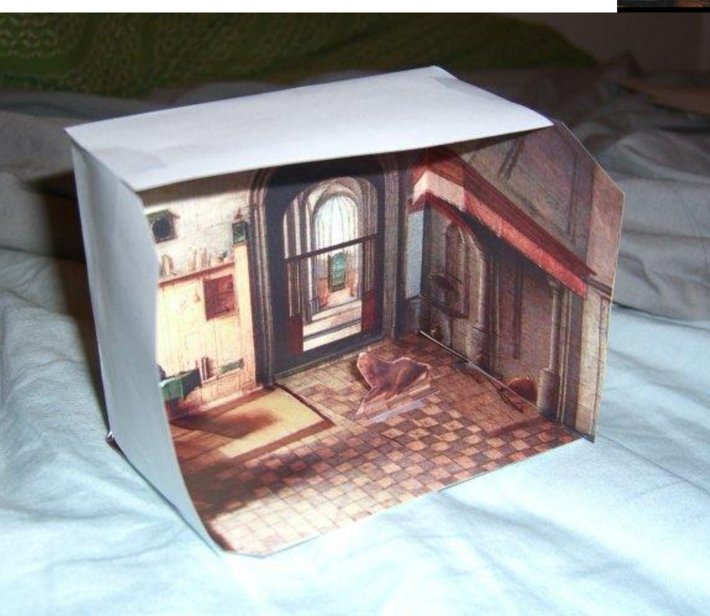
Can we do it from a single image?

“Tour into the Picture”

Step 1: define planes



p 2: rectify each plane



Step 3: compute 3D box coords