Image Formation

Javier González Jiménez

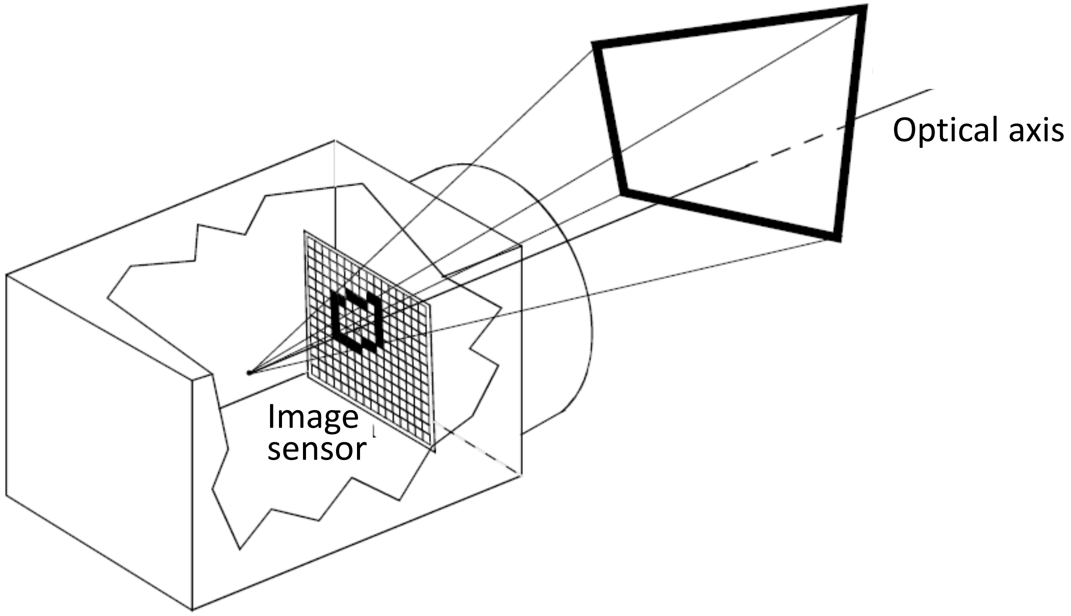
Reference material:

- *Computer Vision: Algorithms and Applications*. Richard Szeliski. Springer. 2010. http://szeliski.org/Book
- An Invitation to 3-D Vision: From Images to Geometric Models. Y Ma, S. Soatto, J. Kosecha, S. Shankar Sanstry. Springer 2003. https://www.eecis.udel.edu/~cer/arv/readings/old_mkss.pdf
- The Perspective Camera - An Interactive Tour (by Kyle Simek) http://ksimek.github.io/2012/08/13/introduction/

Content

1. Introduction

IMAGE FORMATION: Process of projecting the 3D scene objects on to the image plane (2D)



Two main questions:

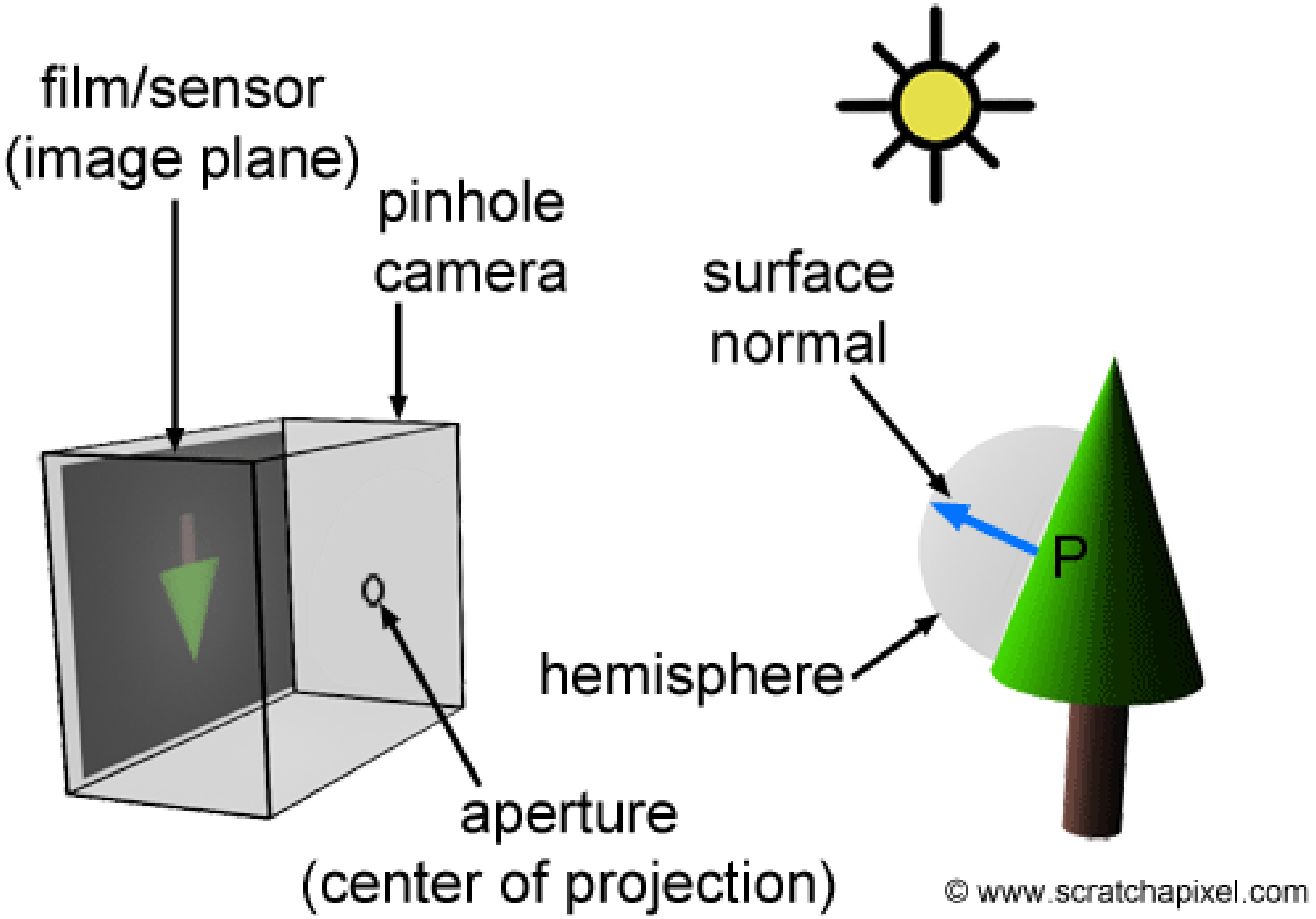
We will study this problem

- Where does each 3D point project on the image? → the geometric problem
- What will be the color on the image of each 3D point → the radiometric problem

WHY?

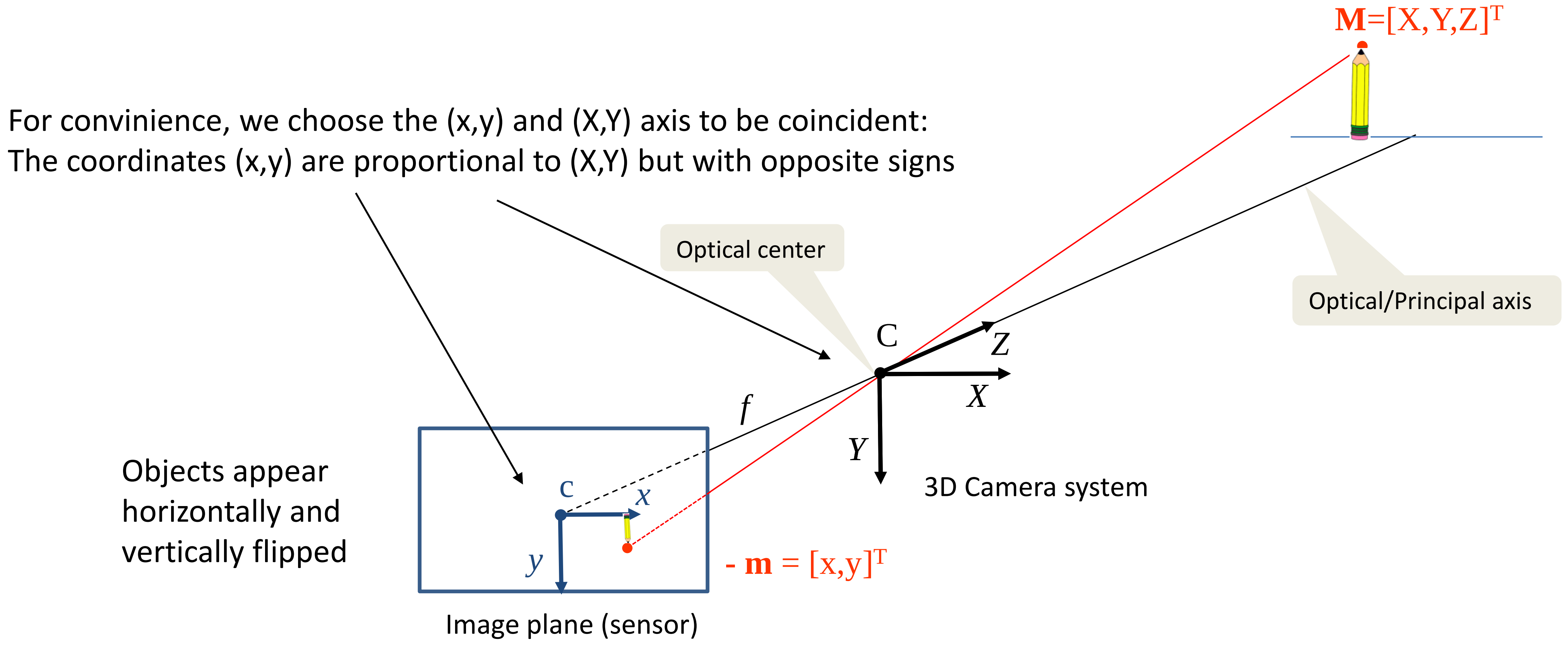
1. Introduction

Image formation: The Pinhole model of a camera



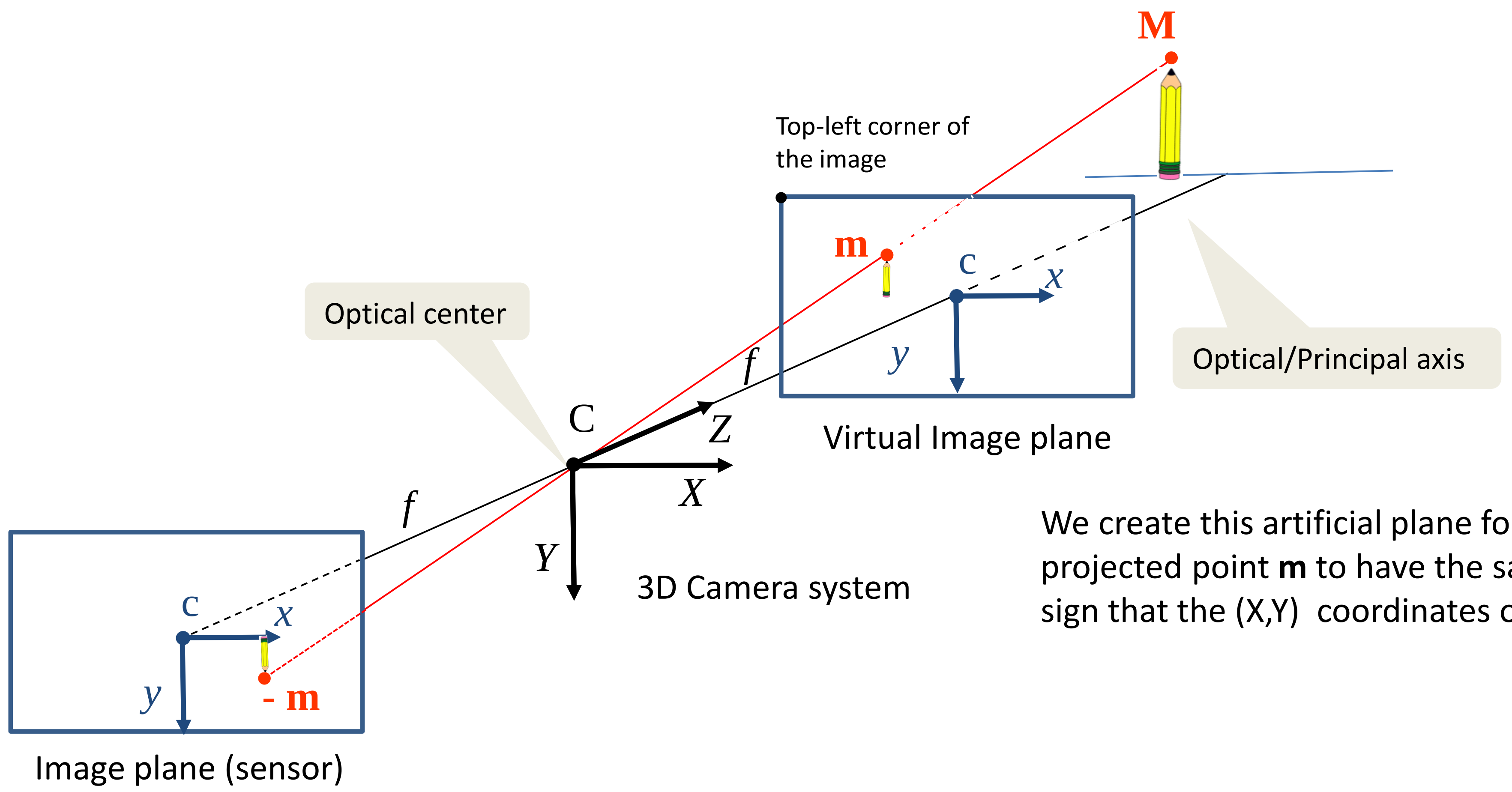
The Pinhole model

From all the light rays departing from a scene 3D point (M) only one passes through the pinhole and projects onto the image plane



The Pinhole model

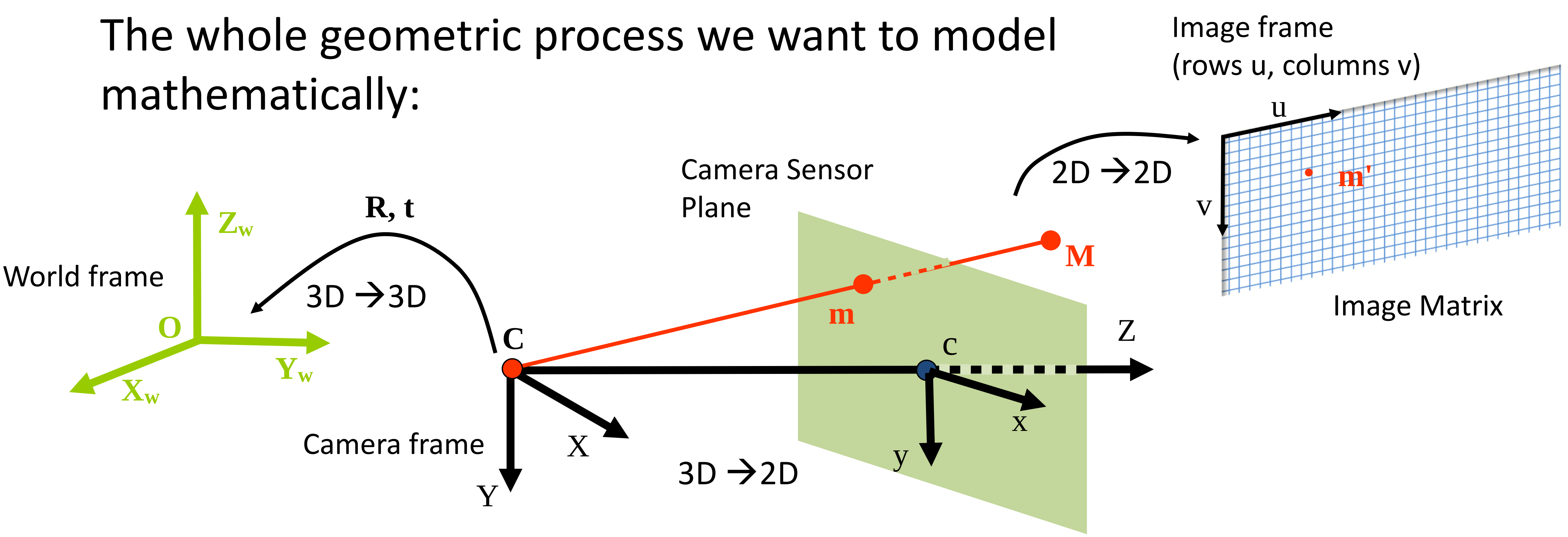
From all the light rays departing from a scene 3D point (M) only one passes through the pinhole and projects onto the image plane



We create this artificial plane for the projected point **m** to have the same sign that the (X,Y) coordinates of M

1. Introduction

The whole geometric process we want to model mathematically:



3 geometric transformations:

3D →2D: A point **M** expressed in the **camera frame** projects on the camera sensor (PINHOLE model)

3D →3D: A point **M** expressed in the **world frame** is transformed to the camera frame

2D →2D: A point **m** expressed in the **sensor plane** is transformed to the **image matrix** (in the computer)

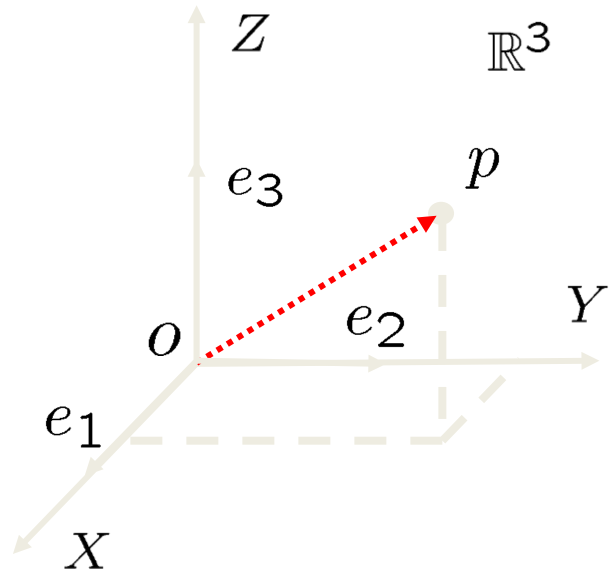
2. Mathematical tools: Euclidean transformations in 3D

3D → 3D

Euclidean basis in 3D:

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Basis: independent vectors that "span the space".



Every vector in the space is a unique combination of the basis vectors

Coordinates of a point in 3D:

$$\mathbf{X} = X\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + Y\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + Z\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = Xe_1 + Ye_2 + Ze_3 = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \in \mathbb{R}^3$$

Courtesy of *Invitation to 3D vision, MASK*

2. Mathematical tools: Euclidean transformations in 3D

A **bound vector** is defined by a pair of points:

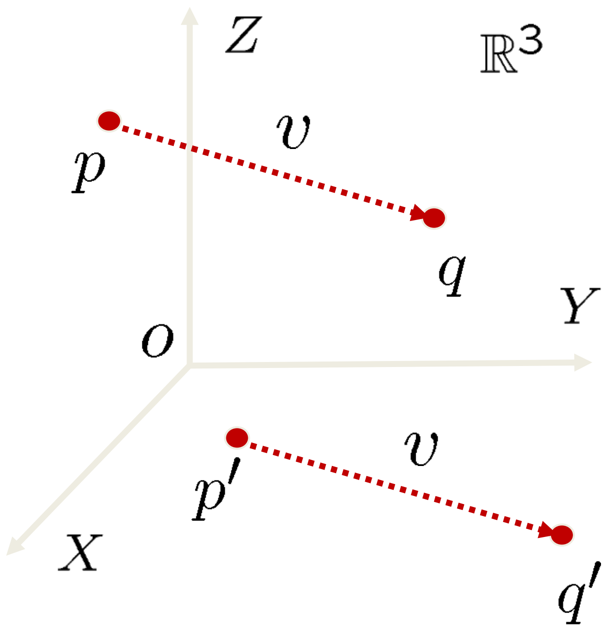
$$\boldsymbol{X}_p = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} \in \mathbb{R}^3, \quad \boldsymbol{X}_q = \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} \in \mathbb{R}^3,$$

A **free vector (or just vector)** is defined by one point (3 coordinates):

$$v = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} X_2 - X_1 \\ Y_2 - Y_1 \\ Z_2 - Z_1 \end{bmatrix} \in \mathbb{R}^3$$

Usually, we will refer to this one!

(p, q)



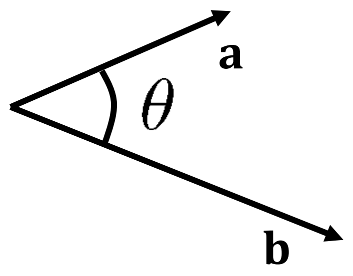
Different end-points, different bound vectors, **SAME free vector** !

A 3D free vector consists of a direction (2 *angles*) + a magnitude (1 scalar) = 3 *dof*

dof :degrees-of-freedom

2. Mathematical tools: Euclidean transformations in 3D

Dot (*inner, scalar*) product of two vectors:



$$\cos\theta = \frac{\mathbf{a}^{\mathrm{T}}\mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|}$$

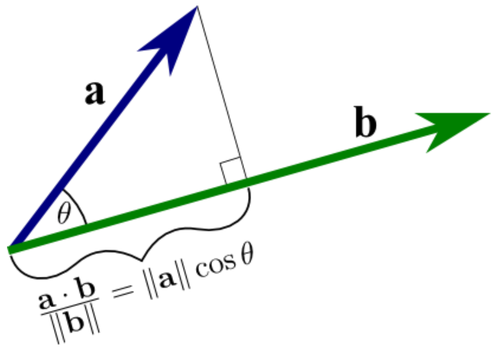
$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^{\mathrm{T}}\mathbf{b} = trace\left(\mathbf{a}\,\mathbf{b}^{\mathrm{T}}\right) = a_1 b_1 + a_2 b_2 + a_3 b_3$$

It's a number! $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^{\mathrm{T}}\mathbf{b} \in \mathbb{R}$

It's commutative! $\langle \mathbf{a}, \mathbf{b} \rangle = \langle \mathbf{b}, \mathbf{a} \rangle$

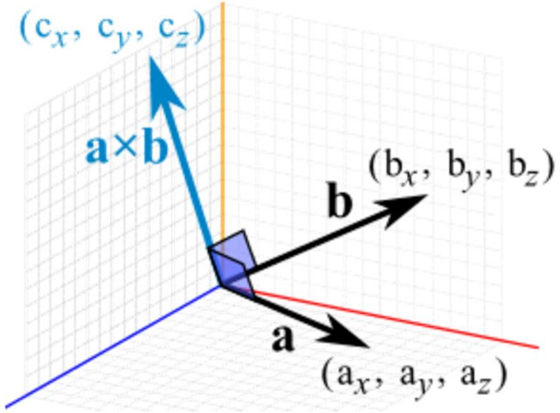
Geometric Meaning: projection of one vector onto the other



The *dot product* induces a **norm** for a vector (a distance between 2 points)

$$\|\mathbf{a}\| = \sqrt{\mathbf{a}^{\mathrm{T}}\mathbf{a}} = \sqrt{a_1^2 + a_2^2 + a_3^2} \qquad \text{Euclidean norm}$$

Cross product of two vectors: **a** x **b**



$$\mathbf{c} = \mathbf{a}\, x\, \mathbf{b} = \begin{vmatrix} i & j & k \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} = \begin{bmatrix} b_3 a_2 - b_2 a_3 \\ b_1 a_3 - b_3 a_1 \\ b_2 a_3 - b_3 a_2 \end{bmatrix}$$

It's **a vector** perpendicular to the two being multiplied!

More convenient if expressed as a **linear (matrix) transformation**

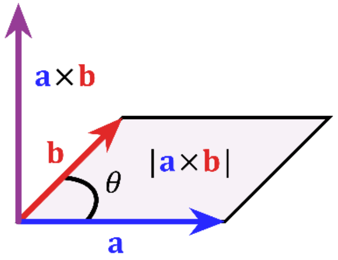
$$\mathbf{c} = \mathbf{a}\, x\, \mathbf{b} = \hat{\mathbf{a}}\mathbf{b} \qquad \underbrace{\hat{\mathbf{a}} = [\mathbf{a}]_x}_{\text{Two different notations}} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \in \mathbb{R}^{3x3}$$

Rank($\hat{\mathbf{a}}$) = 2

$[\mathbf{a}]_x$ is antisymmetric (or skew-symmetric): $A^T = -A$

Geometric Meaning: the module of c is the area of the parallelogram having **a** and **b** as sides

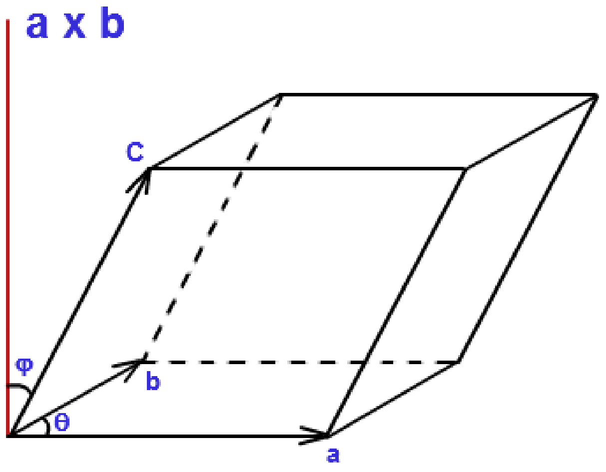
$$\|\mathbf{c}\| = \|\mathbf{a}\|\|\mathbf{b}\| sin\theta$$



2. Mathematical tools: Euclidean transformations in 3D

Product of three vectors:

$\mathbf{c}^T(\mathbf{a}$ x $\mathbf{b})$ is a scalar: (Signed) volume of the parallelepiped defined by the three vectors (**a**, **b**, **c**)



$$\mathbf{c}^T(\mathbf{a}\; x\; \mathbf{b}) = \det\left(\begin{bmatrix}\mathbf{c}\\ \mathbf{a}\\ \mathbf{b}\end{bmatrix}\right) = \begin{vmatrix} c_1 & c_2 & c_3 \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix}$$

Changing the order of the vectors gives the same volume but the sign may be different

A very useful principle in 3D Computer Vision:

When **a,b,c** are coplanar the volume is zero: $\mathbf{c}^T(\mathbf{a}$ x $\mathbf{b})$ = 0

Linear transformation of vectors

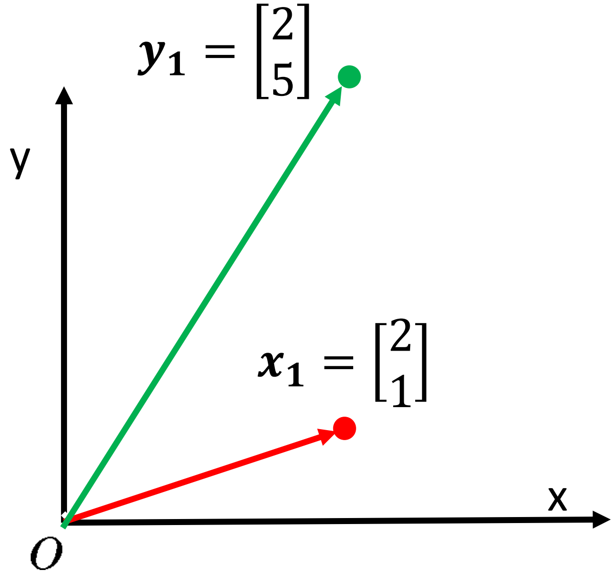
A matrix is a **collection of vectors**

$$\mathbf{A} = \begin{bmatrix} \boxed{\begin{matrix} a_{11} \\ a_{21} \end{matrix}} & \boxed{\begin{matrix} a_{12} \\ a_{22} \end{matrix}} & \boxed{\begin{matrix} a_{13} \\ a_{23} \end{matrix}} \end{bmatrix} = [\boldsymbol{a_1} \quad \boldsymbol{a_2} \quad \boldsymbol{a_3}] \in \mathbb{R}^{2x3}$$

$\boldsymbol{a_1}$ $\boldsymbol{a_2}$ $\boldsymbol{a_3}$ 3 vectors in $\mathbb{R}^2$

Matrix multiplication: linear function that transforms vectors

The transformation can be seen in two ways (Example): $\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 2 & 2 \end{bmatrix}$ $\boldsymbol{x_1} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$



- **Dot product of vectors:** [This is useful for practical multiplication]

$$\boldsymbol{y_1} = \mathbf{A}\boldsymbol{x_1} = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = [1 \quad 0] \begin{bmatrix} 2 \\ 1 \end{bmatrix} + [2 \quad 1] \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$$

- **Column combination:** [This is more interesting for understanding what the matrix multiplication does]

$$\boldsymbol{y_1} = \mathbf{A}\boldsymbol{x_1} = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = 2 \begin{bmatrix} 1 \\ 2 \end{bmatrix} + 1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$$

Combination of columns to produce $\boldsymbol{y_1}$: Now the vector $\boldsymbol{x_1}$ is expressed in a new basis defined by the columns of $\mathbf{A}$

What does LINEAR mean?

The following two properties hold:

$\boldsymbol{f}(\boldsymbol{x_1} + \boldsymbol{x_2}) = \boldsymbol{f}(\boldsymbol{x_1}) + \boldsymbol{f}(\boldsymbol{x_2})$	Aditivity
$\boldsymbol{f}(a\boldsymbol{x_1}) = a\boldsymbol{f}(\boldsymbol{x_1})$	Scaling

Equivalently, the *superposition principle* holds:

$$\boldsymbol{f}(\alpha\boldsymbol{x_1}+\beta\boldsymbol{x_2}) = \alpha\boldsymbol{f}(\boldsymbol{x_1}) + \beta\boldsymbol{f}(\boldsymbol{x_2})$$

Matrix multiplication ($\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$) satisfies this property:

$$\mathbf{A}(\alpha\mathbf{x_1}+\beta\mathbf{x_2}) = \alpha\mathbf{A}\mathbf{x_1} + \beta\mathbf{A}\mathbf{x_2} = \alpha\mathbf{y_1} + \beta\mathbf{y_2}$$

The transformation of linear combination of vectors ($\alpha\mathbf{x_1}+\beta\mathbf{x_2}$) is the same linear combination of the transformed vectors ($\alpha\mathbf{y_1} + \beta\mathbf{y_2}$).

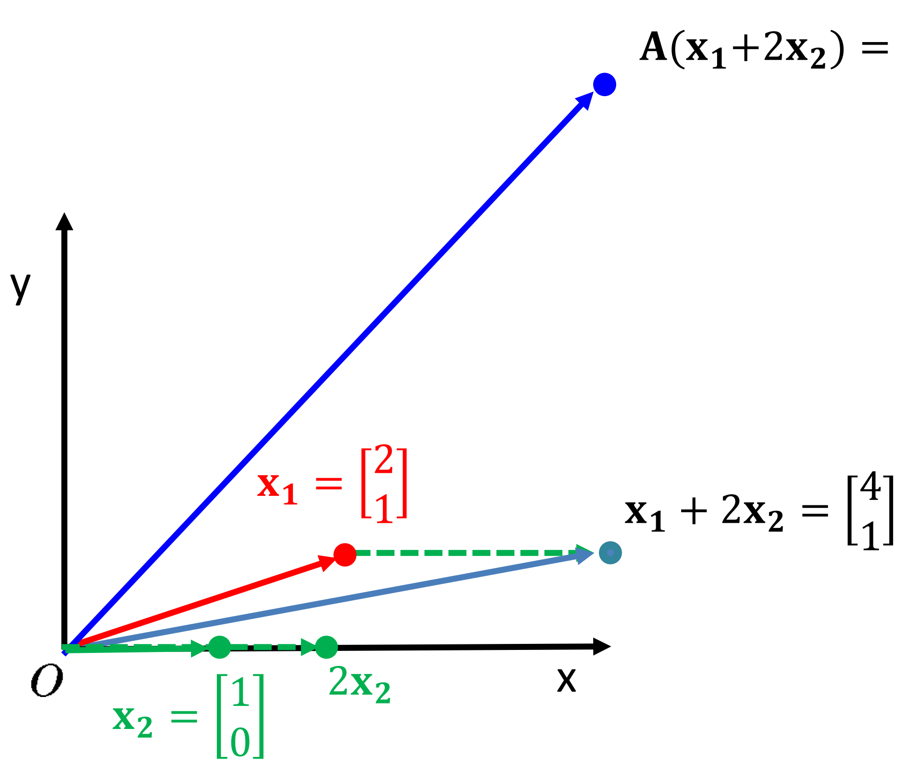
Notice: $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$ is NOT a linear transformation (additivity *does not* hold)

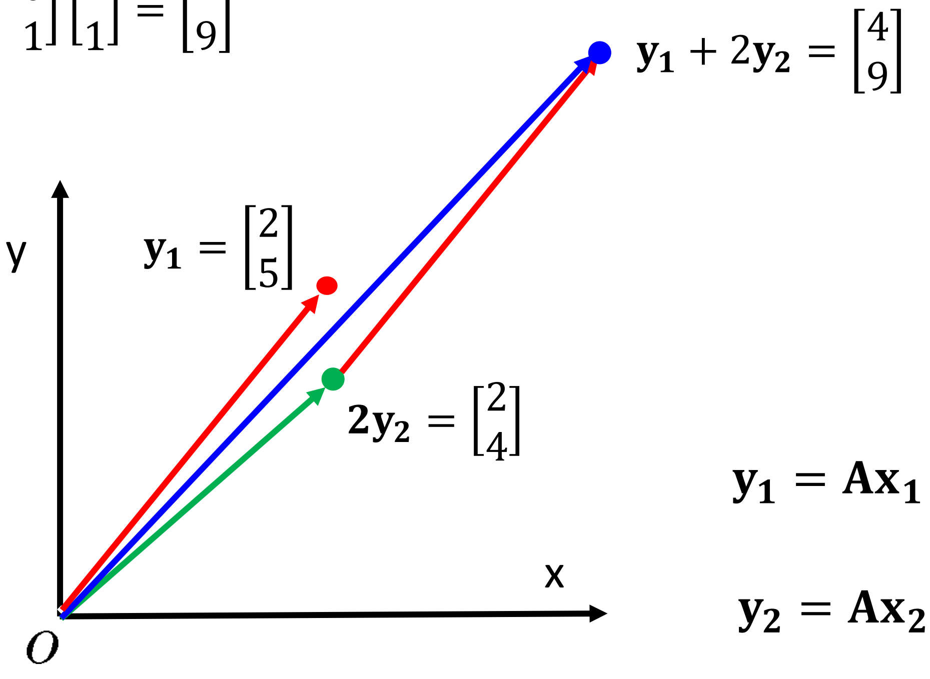
$$\boldsymbol{f}(\boldsymbol{x_1} + \boldsymbol{x_2}) = \boldsymbol{A}(\boldsymbol{x_1} + \boldsymbol{x_2}) + \boldsymbol{b} \neq \boldsymbol{f}(\boldsymbol{x_1}) + \boldsymbol{f}(\boldsymbol{x_2}) = \boldsymbol{A}\boldsymbol{x_1} + \boldsymbol{b} + \boldsymbol{A}\boldsymbol{x_2} + \boldsymbol{b}$$

What does LINEAR mean? The superposition principle holds

$$\mathbf{A}(\alpha\mathbf{x}_1+\beta\mathbf{x}_2) = \alpha\mathbf{A}\mathbf{x}_1 + \beta\mathbf{A}\mathbf{x}_2 = \alpha\mathbf{y}_1 + \beta\mathbf{y}_2$$

Example: $\alpha = 1$ $\beta = 2$ $\mathbf{x}_1 = \begin{bmatrix}2\\1\end{bmatrix}$ $\mathbf{x}_2 = \begin{bmatrix}1\\0\end{bmatrix}$ $\mathbf{A} = \begin{bmatrix}1 & 0\\2 & 1\end{bmatrix}$





$$\mathbf{y}_1 = \mathbf{A}\mathbf{x}_1 = \begin{bmatrix}1 & 0\\2 & 1\end{bmatrix}\begin{bmatrix}2\\1\end{bmatrix} = \begin{bmatrix}2\\5\end{bmatrix}$$

$$\mathbf{y}_2 = \mathbf{A}\mathbf{x}_2 = \begin{bmatrix}1 & 0\\2 & 1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1\\2\end{bmatrix}$$

2. Euclidean transformations in 3D

Rotation matrix: linear transformation of vectors that preserves their length

$$R = \begin{array}{c} \left[\begin{array}{ccc} \boxed{\begin{matrix} r_{11} \\ r_{21} \\ r_{31} \end{matrix}} & \boxed{\begin{matrix} r_{12} \\ r_{22} \\ r_{32} \end{matrix}} & \boxed{\begin{matrix} r_{13} \\ r_{23} \\ r_{33} \end{matrix}} \end{array}\right] \\ \begin{array}{ccc} r_x \quad & \quad r_y \quad & \quad r_z \end{array} \end{array} = [r_x \quad r_y \quad r_z] \in \Re^{3x3}$$

Coordinates of the original basis $<r_x, r_y, r_z>$ in the new one

A rotation matrix *R* verifies:

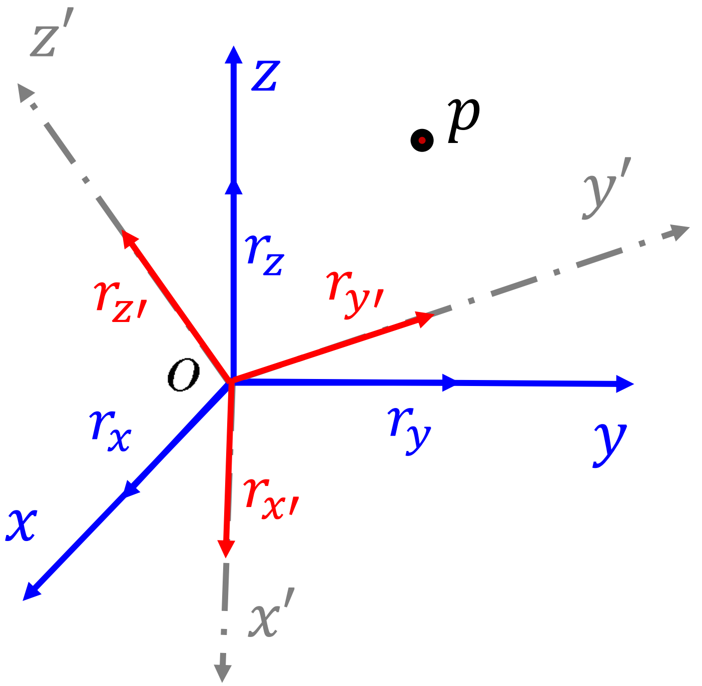
1. R is an **orthogonal** matrix:

$$R^T R = R R^T = I \quad \rightarrow \quad R^T = R^{-1}$$

2. $\det(R) = +1$

Rotation of a (vector) point $p = [x, y, z]^{\mathrm{T}}$

$$p' = Rp$$



$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

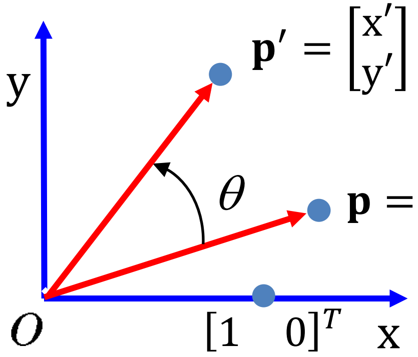
Rotation transformation

We have two possibilities (example in 2D for simplicity):

- Rotate a point in a still frame (called *active rotation*)

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

[Useful for computer graphics/robotics: moving objects with a fixed frame]



$$\mathbf{p}' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

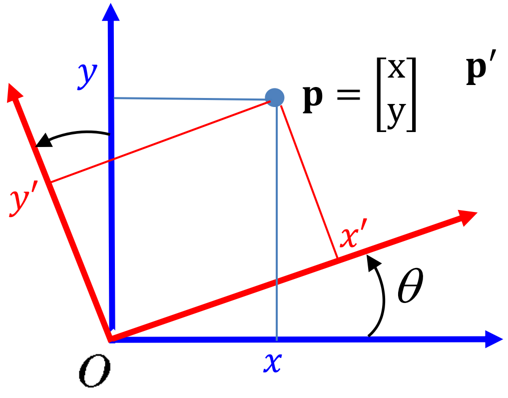
$$\mathbf{p} = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} \cos 45º & -\sin 45º \\ \sin 45º & \cos 45º \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

- Express a still point in a rotated new frame (called *passive rotation*)

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

[Useful for Computer Vision (used later on): moving a camera in a static world]



$$\mathbf{p} = \begin{bmatrix} x \\ y \end{bmatrix}$$

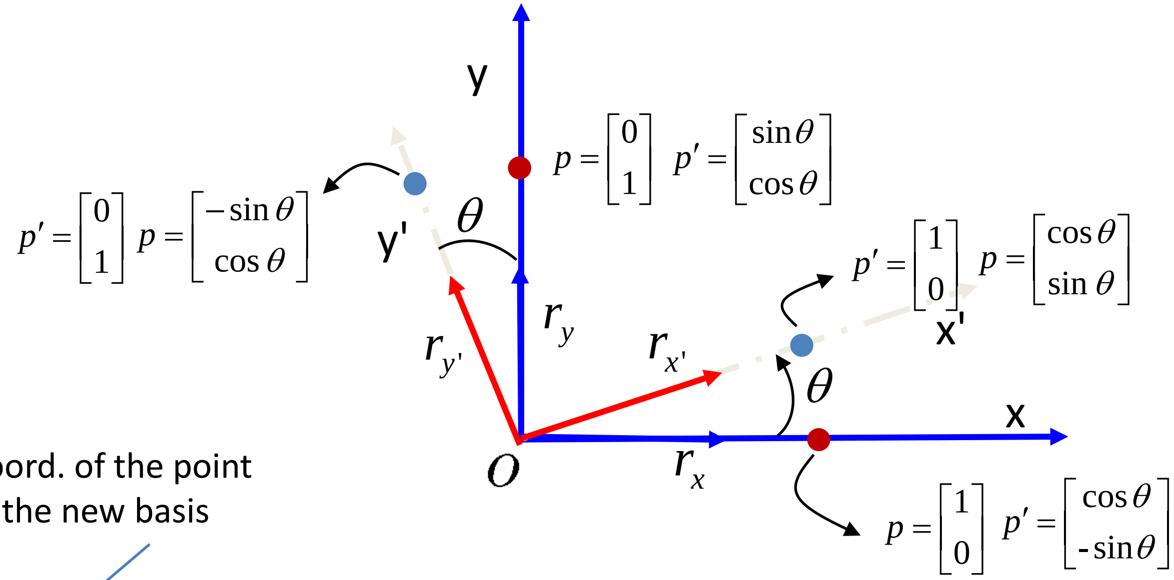
$$\mathbf{p}' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Note.- Any rotation is assumed to be counterclockwise

Passive rotation example in 2D (rotation of the frame): $p' = Rp$

$$R = [r_x \quad r_y] = \begin{bmatrix} \boxed{\begin{matrix}\cos\theta \\ -\sin\theta\end{matrix}} & \boxed{\begin{matrix}\sin\theta \\ \cos\theta\end{matrix}} \end{bmatrix}$$

Column vectors: coordinates of the original basis <r_x r_y> in the new one



Coord. of the point in the original basis

Coord. of the point in the new basis

Checking out:

$$p = r_x = \begin{bmatrix}1\\0\end{bmatrix} \longrightarrow p' = Rp = \begin{bmatrix}\cos\theta & \sin\theta \\ -\sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}\cos\theta\\ -\sin\theta\end{bmatrix}$$

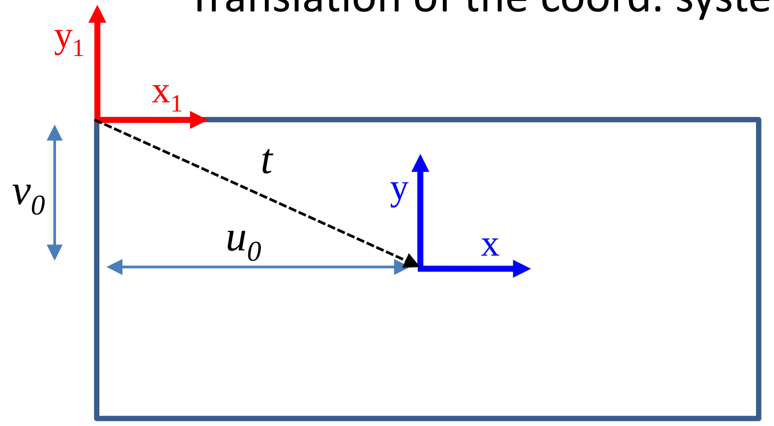
$$p = r_y = \begin{bmatrix}0\\1\end{bmatrix} \longrightarrow p' = Rp = \begin{bmatrix}\cos\theta & \sin\theta \\ -\sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}\sin\theta\\ \cos\theta\end{bmatrix}$$

Applying a rotation clockwise is equivalent to rotating the point p to a new position p' (active rotation):

$$p = R^T p' \qquad R^T = \begin{bmatrix} \boxed{\begin{matrix}\cos\theta \\ \sin\theta\end{matrix}} & \boxed{\begin{matrix}-\sin\theta \\ \cos\theta\end{matrix}} \end{bmatrix}$$

Image Rotation and Translation (2D example): A point (pixel) is expressed in different frames

Translation of the coord. system

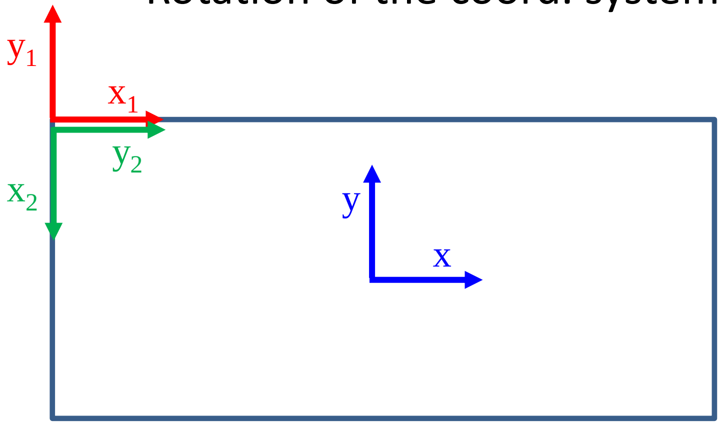


$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} u_0 \\ -v_0 \end{bmatrix}$$

Old origin in the new system

$$p_1 = p + t$$

Rotation of the coord. system



$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos(-90) & \sin(-90) \\ -\sin(-90) & \cos(-90) \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} -y_1 \\ x_1 \end{bmatrix}$$

Old basis (x_1,y_1) in the new system (x_2,y_2)

$$p_2 = Rp_1$$

All together:

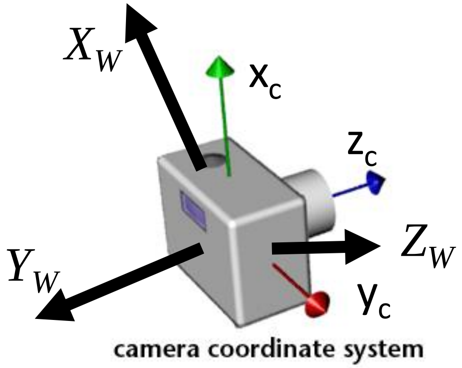
$$p_2 = Rp_1 = R\underbrace{(p + t)}_{p_1} = Rp + Rt = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u_0 \\ -v_0 \end{bmatrix} = \begin{bmatrix} -y + v_0 \\ x + u_0 \end{bmatrix}$$

Validation: $(x, y) \rightarrow (x_2, y_2)$

- $(0,0) \rightarrow (v_0, u_0)$
- $(1,0) \rightarrow (v_0, 1+u_0)$
- $(0,1) \rightarrow (v_0 - 1, u_0)$

2. Euclidean transformations in 3D

Rotation matrix (from the World to the Camera): $P^C = R_W^C P^W$



Can be seen as a sequence of three elemental rotations:

$$R_W^C = \begin{bmatrix} r_{11} & \boxed{r_{12}} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = R_z(yaw)R_y(pitch)R_x(roll)$$

One out of many possibilities

Coordinates of Y_W in the Camera system $[X_C\ Y_C\ Z_C]$

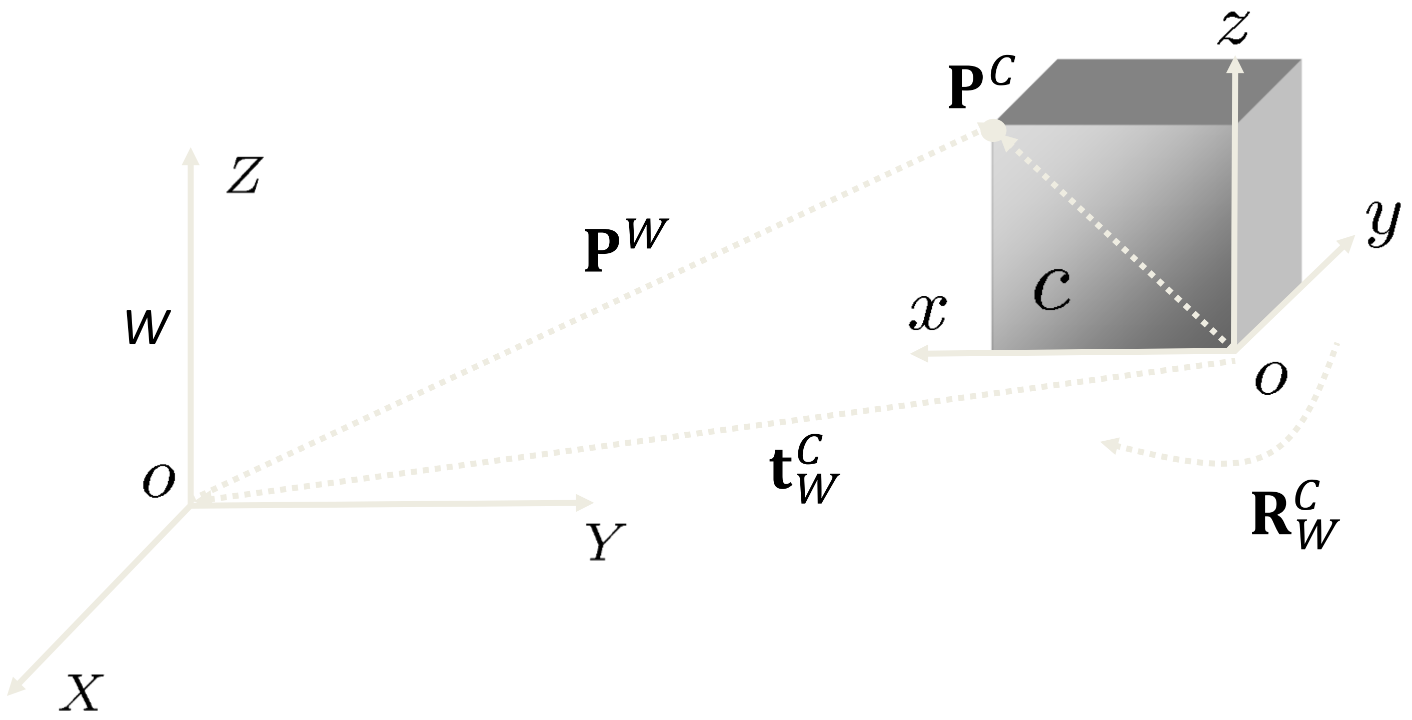
$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

No rotation $\quad R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

2. Euclidean transformations in 3D

If the coordinate system is also translated, the point coordinates are related by:

$$\mathbf{P}^C = \mathbf{R}_W^C \mathbf{P}^W + \mathbf{t}_W^C$$



$\mathbf{R}_W^C$: columns are the projection of the World axes in the Camera
$\mathbf{t}_W^C$: origin of the World in the Camera

2. Mathematical tools: Homogeneous transformations

- Homogeneous (also called projective) transformations are linear transformations (i.e. matrix multiplications) between homogeneous coordinates (vectors)

- Homogeneous coordinates are obtained from the Cartesian (inhomogeneous) vector by extending it with an arbitrary non-negative number (for convenience 1)

Inhomogeneous coordinates $\boldsymbol{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \in \Re^3$ $\Rightarrow$ Homogeneous coordinates $\tilde{\boldsymbol{X}} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \lambda X \\ \lambda Y \\ \lambda Z \\ \lambda \end{bmatrix} \in \Re^4$

This means EQUIVALENT

We can go back by dividing the 3-first coord. by the fourth:

$$\tilde{\boldsymbol{X}} = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} \in \Re^4 \Rightarrow \boldsymbol{X} = \begin{bmatrix} A/D \\ B/D \\ C/D \end{bmatrix} \in \Re^3$$

Notice:
If all the elements of $\tilde{\boldsymbol{X}}$ are scaled by any scalar λ, we obtain the same $\boldsymbol{X}$ because λ cancels out when dividing by the fourth element

2. Homogeneous transformations

About the scale factor λ :

The family of homogeneous vectors $\lambda \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \end{bmatrix}$ with $\lambda \neq 0$ represents the **same point in R³:** $\begin{bmatrix} x'_1/x'_4 \\ x'_2/x'_4 \\ x'_3/x'_4 \end{bmatrix}$ λ does not affect!

As a consequence: Any transformation in homogeneous coordinates holds for any scaled matrix:

$$\lambda \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} \xRightarrow{\text{Equivalent to}} \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ 1 \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix}$$

Then, $\begin{bmatrix} 1 & 2 & 3 & 4 \\ -2 & 1 & -2 & -1 \\ 3 & 0 & 1 & 2 \\ -2 & -1 & -4 & 1 \end{bmatrix}$ $\lambda \begin{bmatrix} 1 & 2 & 3 & 4 \\ -2 & 1 & -2 & -1 \\ 3 & 0 & 1 & 2 \\ -2 & -1 & -4 & 1 \end{bmatrix}$ represent the SAME transformation

(we say that is a **Homogeneous Matrix**)

- This indetermination is typically handled by fixing one entry of the matrix, e.g.: p44= 1
- These matrices must be non-singular (Rank = 4)

2. Homogeneous transformations

Rules of use for homogeneous coordinates (for example in 3D)

1. Go from inhomogeneous to homogeneous

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \rightarrow \lambda \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix}$$

Point in 3D → line in 4D

2. Transformation in the linear space by matrix multiplication

$$\lambda \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix}$$

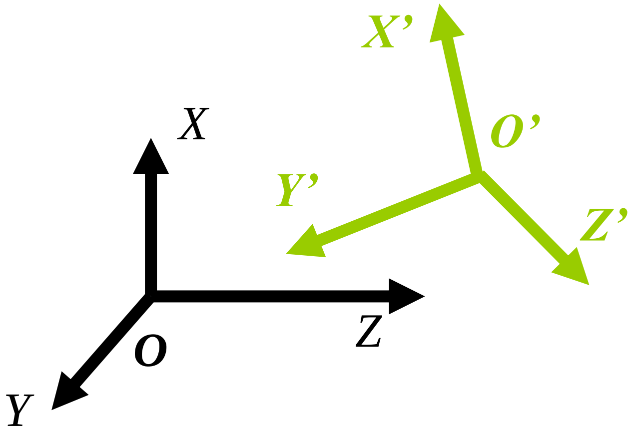
3. Go back from homogeneous to inhomogeneous

$$\lambda \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \end{bmatrix} \rightarrow \begin{bmatrix} x'_1/x'_4 \\ x'_2/x'_4 \\ x'_3/x'_4 \end{bmatrix}$$

What if x_4=0? → point at infinity (not really a point, but a direction in 3D, i.e. a line in 3D)

2. Homogeneous transformations

Why needed? [1] it deals with transformations between coordinates systems very conveniently



This is an **affine** transformationof **p'**

$$\boxed{\mathbf{p} = \underset{\text{rotation}}{\mathbf{R}}\mathbf{p}' + \underset{\text{translation}}{\mathbf{t}}} \quad \longrightarrow \quad \boxed{\tilde{\mathbf{p}} = \underset{\text{1 matrix 4x4}}{\mathbf{T}}\tilde{\mathbf{p}}'}$$

This is a **linear** transformationof **p'**

$$\underbrace{\begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix}}_{\tilde{\mathbf{p}}} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{p}' \\ 1 \end{bmatrix}}_{\tilde{\mathbf{p}}'} = \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_x \\ r_{31} & r_{32} & r_{33} & t_x \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

The inverse transformation:

$$\tilde{\mathbf{p}}' = \mathbf{T}^{-1}\tilde{\mathbf{p}} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}^{-1} \tilde{\mathbf{p}}$$

$$\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{R}^{-1} & -\mathbf{R}^{-1}\mathbf{t} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$$

In **cartesian coord.**, the concatenation becomes a mess:

$$\mathbf{p}^2 = \mathbf{R}_1^2\mathbf{p}^1 + \mathbf{t}_1^2$$

$$\mathbf{p}^3 = \mathbf{R}_2^3\mathbf{p}^2 + \mathbf{t}_2^3 = \mathbf{R}_2^3(\mathbf{R}_1^2\mathbf{p}^1 + \mathbf{t}_1^2) + \mathbf{t}_2^3 = \mathbf{R}_2^3\mathbf{R}_1^2\mathbf{p}^1 + \mathbf{R}_2^3\mathbf{t}_1^2 + \mathbf{t}_2^3$$

In **homogeneous** it is much easier:

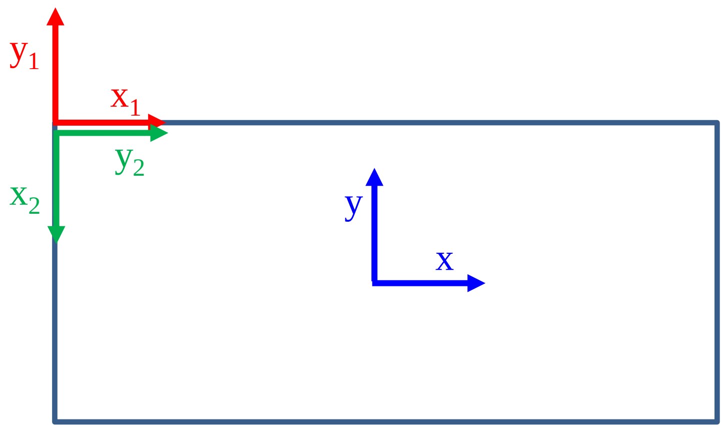
$$\tilde{\mathbf{p}}^2 = \mathbf{T}_1^2\,\tilde{\mathbf{p}}^1$$

$$\tilde{\mathbf{p}}^3 = \mathbf{T}_2^3\,\tilde{\mathbf{p}}^2 = \mathbf{T}_2^3\mathbf{T}_1^2\,\tilde{\mathbf{p}}^1$$

2. Homogeneous transformations

Why needed? [1] it deals with transformations between coordinates systems very conveniently

2D example (in previous slide)



In inhomogeneous: $\mathbf{p}_2 = \mathbf{R}\mathbf{p}_1 = \mathbf{R}(\mathbf{p}+\mathbf{t}) = \mathbf{R}\mathbf{p} + \mathbf{R}\mathbf{t} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} \mathrm{x} \\ \mathrm{y} \end{bmatrix} + \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} \mathrm{u}_0 \\ -\mathrm{v}_0 \end{bmatrix} = \begin{bmatrix} -\mathrm{y}+\mathrm{v}_0 \\ \mathrm{x}+\mathrm{u}_0 \end{bmatrix}$

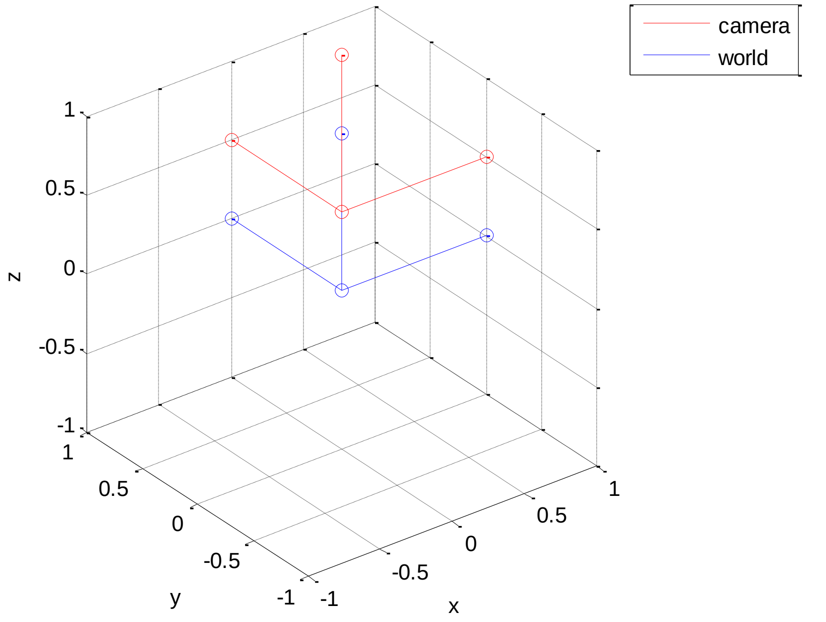
In homogeneous: $\tilde{\mathbf{p}}_2 = \mathbf{T}_{21}\tilde{\mathbf{p}}_1 = \mathbf{T}_2\mathbf{T}_1\tilde{\mathbf{p}} = \underbrace{\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Rotation } (\mathbf{T}_2)}\underbrace{\begin{bmatrix} 1 & 0 & u_0 \\ 0 & 1 & -v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Translation } (\mathbf{T}_1)}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -1 & v_0 \\ 1 & 0 & u_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T}_{21}}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} -y+v_0 \\ x+u_0 \\ 1 \end{bmatrix}$

2. Homogeneous transformations

Example: Moving the Camera C (Starting from C being coincident to W)

- Move the camera 0.5 meters forward (z-axis)

```
% Transform the origin [0 0 0]' and the axis of the
% camera, whose endpoints are [1 0 0]',[0 1 0]',[0 0 1]'
t = [0;0;0.5];
R = eye(3,3);
Zero = zeros(3,1);
T = [R t; Zero',1];
showTransformation(T);
```



2. Homogeneous transformations

Example: Moving the Camera C (Starting from C being coincident to W)

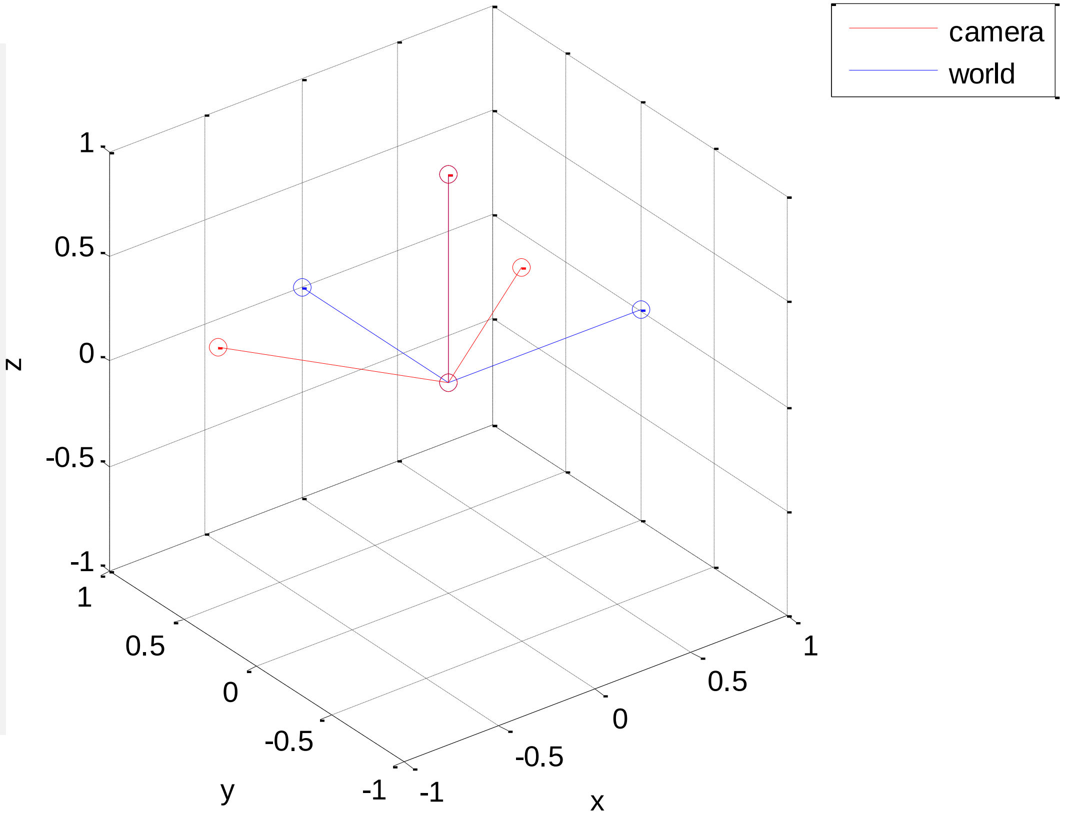
- Rotate the camera 35º to look to the left (yaw rotation).

```
t = [0;0;0];
yaw = degtorad(35); pitch = 0; roll = 0;

cy = cos(yaw); sy = sin(yaw);
cp = cos(pitch); sp = sin(pitch);
cr = cos(roll); sr = sin(roll);

R = [cy*cp, cy*sp*sr-sy*cr, cy*sp*cr+sy*sr; ...
     sy*cp, sy*sp*sr+cy*cr, sy*sp*cr-cy*sr; ...
      -sp, cp*sr, cp*cr];
T = [R t; Zero',1];

showTransformation(T);
```



2. Mathematical tools: Homogeneous transformations

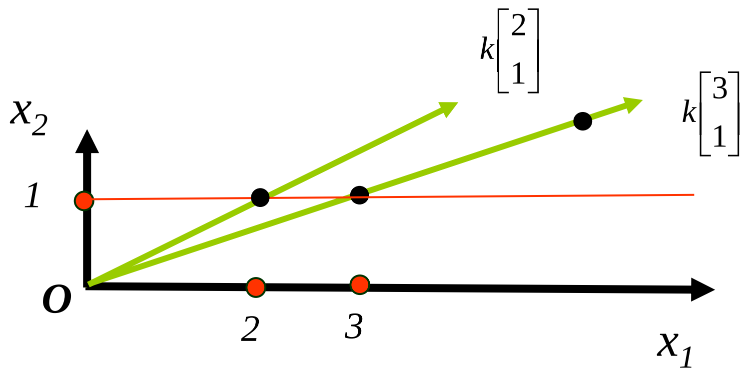
Why needed? : [2] A natural model for the camera: points in the image plane $\Re^2$ are projection rays in $\Re^3$

1D

Cartesian coordenates (inhomogeneous): $x=x_1=3$

Homogeneous coordinates: $x = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \equiv \begin{bmatrix} 6 \\ 2 \end{bmatrix} \equiv \begin{bmatrix} 3k \\ k \end{bmatrix} \quad k \neq 0$

All these pairs are equivant since they represent the same 1D point x=3 (x_1/x_2)



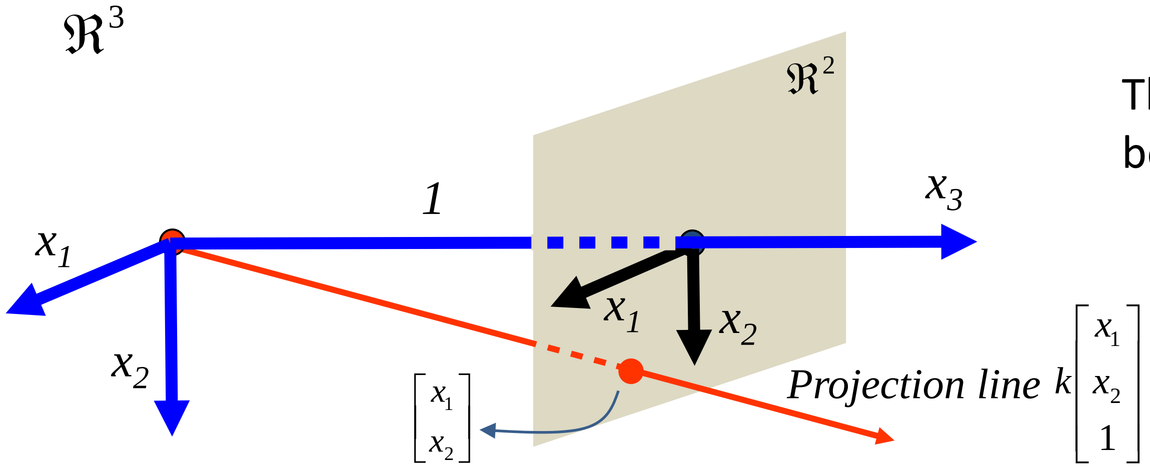
A 1D-point transforms to a line passing through the origin in 2D!

2. Homogeneous transformations

2D

Cartesian coord.: $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \Re^2$ $\Rightarrow$ Homogeneous coord.: $\mathbf{x} = k \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} \quad k \neq 0$

The **projective plane**, called P^2, is the set of 3-tuples of real numbers, such that $\begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} \equiv k \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} \quad k \neq 0$



These two 3-tuples are equivalent because they both represent the same point in $\Re^2$

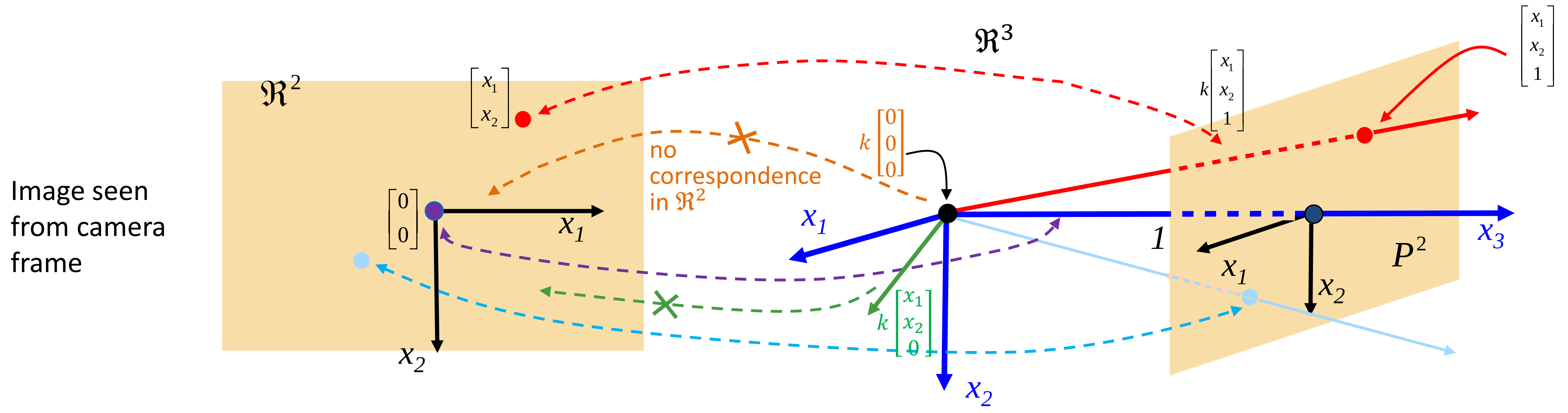
$$\begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix} \in P^2 \rightarrow \begin{bmatrix} \frac{x_1'}{x_3'} \\ \frac{x_2'}{x_3'} \end{bmatrix} \in \Re^2$$

- A point in P^2 (3-tuple) is represented in $\Re^3$ as a line passing through the origin
- k is the x_3 component (depth): indicates a specific point along the line

The homogeneous coordinates of a point in the plane ($\Re^2$) transform to a line passing through the origin in a reference frame parallel to the image plane (perpendicular to x_3)

2. Homogeneous transformations

Each point in $\Re^2$ has a mapping in P^2 (a ray passing through the origin in $\Re^3$)



But NOT all rays in $\Re^3$ (i.e. elements in P^2) have a mapping back to $\Re^2$

Homogeneous Points: $k\begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix} \in P^2$

- have no correspondence point in $\Re^2$ (image plane). They are points at infinity
- is a line on the plane $x_3 = 0$ at direction $[x_1, x_2]^T \rightarrow$ Indicates a direction in the image plane (i.e. it's a vector, not a point!)

Points at infinity have **last coordinate zero** in a homogeneous coordinate representation

2. Homogeneous transformations

Transformations with homogeneous coordinates

- **From 3D to 3D:** 3D homography

$$\lambda\begin{bmatrix}X'\\Y'\\Z'\\1\end{bmatrix}=\begin{bmatrix}p_{11}&p_{12}&p_{13}&p_{14}\\p_{21}&p_{22}&p_{23}&p_{24}\\p_{31}&p_{32}&p_{33}&p_{34}\\p_{41}&p_{42}&p_{43}&p_{44}\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

Particular case $\Longrightarrow$

Euclidean (or rigid) transformation in 3D

$$\lambda\begin{bmatrix}X'\\Y'\\Z'\\1\end{bmatrix}=\begin{bmatrix}r_{11}&r_{12}&r_{13}&t_{14}\\r_{21}&r_{22}&r_{23}&t_{24}\\r_{31}&r_{32}&r_{33}&t_{34}\\0&0&0&\lambda\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

Seen before

$$\begin{bmatrix}\mathbf{p}\\1\end{bmatrix}=\begin{bmatrix}\mathbf{R}&\mathbf{t}\\0&1\end{bmatrix}\begin{bmatrix}\mathbf{p}'\\1\end{bmatrix}=\underbrace{\begin{bmatrix}r_{11}&r_{12}&r_{13}&t_x\\r_{21}&r_{22}&r_{23}&t_x\\r_{31}&r_{32}&r_{33}&t_x\\0&0&0&1\end{bmatrix}}_{\mathbf{T}}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

- **From 2D to 2D**: 2D homography

$$\lambda\begin{bmatrix}x'\\y'\\1\end{bmatrix}=\begin{bmatrix}p_{11}&p_{12}&p_{13}\\p_{21}&p_{22}&p_{23}\\p_{31}&p_{32}&p_{33}\end{bmatrix}\begin{bmatrix}x\\y\\1\end{bmatrix}$$

To be studied next

- **From 3D to 2D** : Projection of 3D points to an image plane → **Perspective projection**

$$\lambda\begin{bmatrix}x\\y\\1\end{bmatrix}=\begin{bmatrix}p_{11}&p_{12}&p_{13}&p_{14}\\p_{21}&p_{22}&p_{23}&p_{24}\\p_{31}&p_{32}&p_{33}&p_{34}\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

To be studied next

3. 2D Homography

2D Homography is a very general linear transformation between planes

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \qquad \begin{aligned} x' &= u/w \\ y' &= v/w \end{aligned}$$

(x,y) coordinates

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix}$$

(x',y') coordinates

We can wrap images freely but:

1. Lines are kept straight
2. Incident lines remain

from R. Szeliski, S. Seitz, D. Hoiem, and I. Kemelmacher

Family of projective transformations in 2D (2D Homography): $P^2 \rightarrow P^2$

EUCLIDEAN (rigid) Transformation

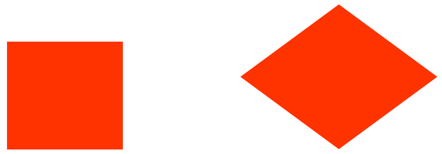
3 unknowns → 2 points needed

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\mathrm{sen}\,\theta & t_x \\ \mathrm{sen}\,\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

AFFINE Transformation

6 unknowns → 3 points needed

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & t_x \\ h_{10} & h_{11} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

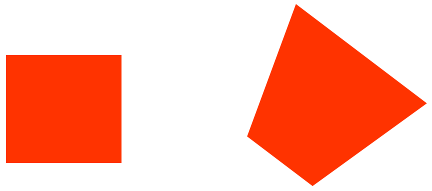


λ=1 (not needed in these two cases)

PROJECTIVE Transformación
(General 2D Homography)

8 unknowns → 4 points needed

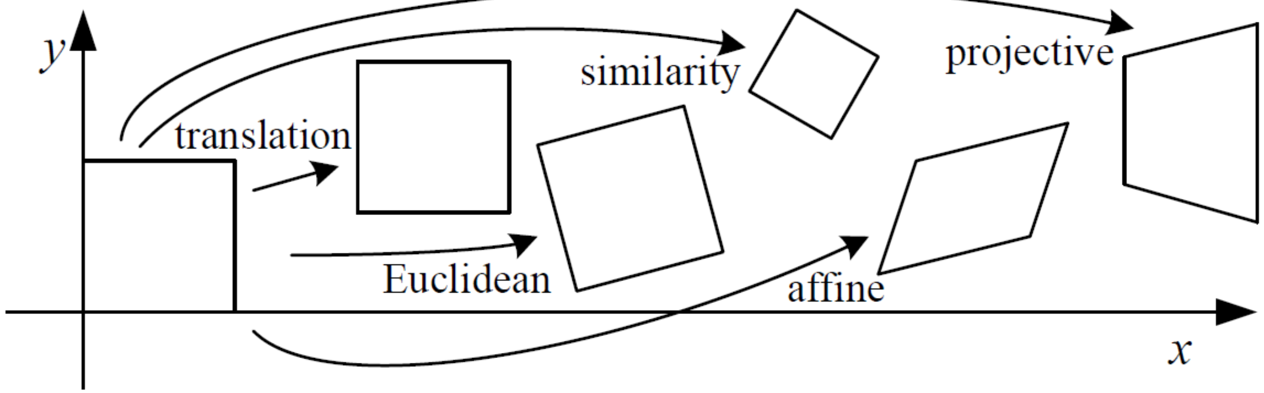
$$\lambda \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$



Remarks:

- Matrices must be non-singular → Rank 3
- Matrices are upto a scale factor:
  - Since H operates on homogeneous coordinates, we can divide H by any constant without changing the result
  - To solve it, we need to set one contraint e.g. h_{22}=1 or for $||\mathbf{h}|| = 1$ (better)

3. 2D Homography



Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} I \mid t \end{bmatrix}_{2\times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} R \mid t \end{bmatrix}_{2\times 3}$	3	lengths	
similarity	$\begin{bmatrix} sR \mid t \end{bmatrix}_{2\times 3}$	4	angles	
affine	$\begin{bmatrix} A \end{bmatrix}_{2\times 3}$	6	parallelism	
projective	$\begin{bmatrix} \tilde{H} \end{bmatrix}_{3\times 3}$	8	straight lines	

http://szeliski.org/Book/

3. 2D Homography

Affine vs. projective

Affine transform

When depth variation within the planar object is small and the camera is far away

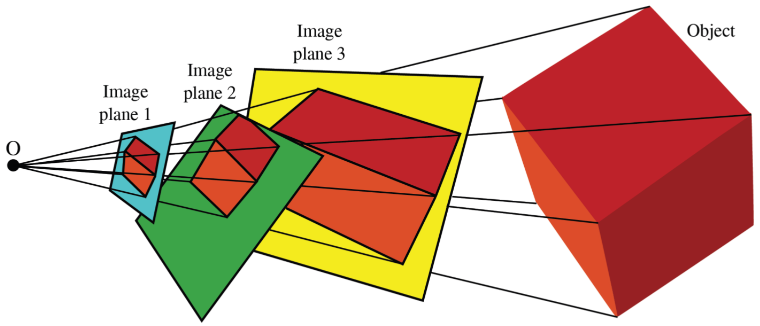
Projective transform

When variation in depth is comparable to distance to object

3. 2D Homography

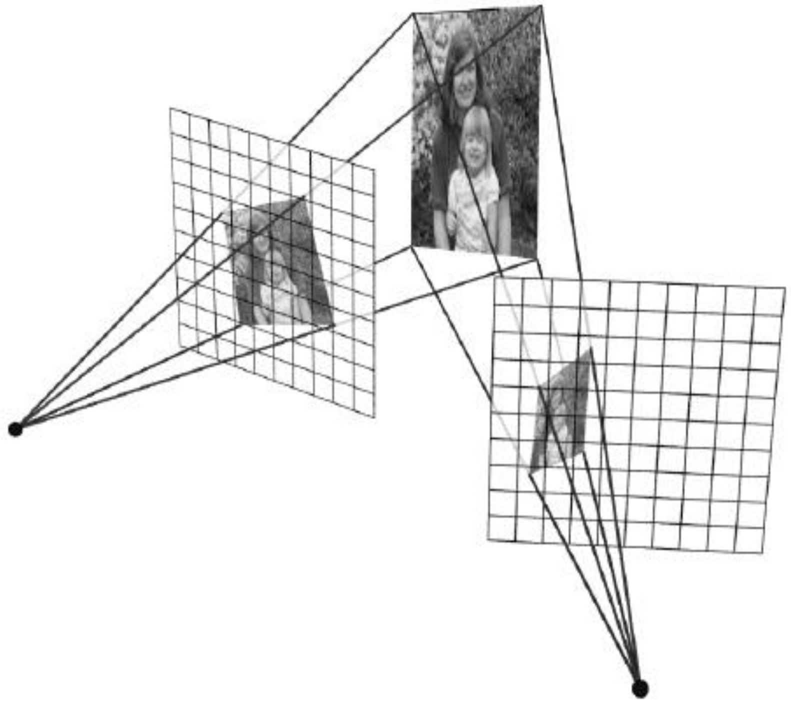
There is a homography H_{3x3} between points in these 3 cases:

Camera still (different image planes)



Rotating camera observing a nonplanar scene

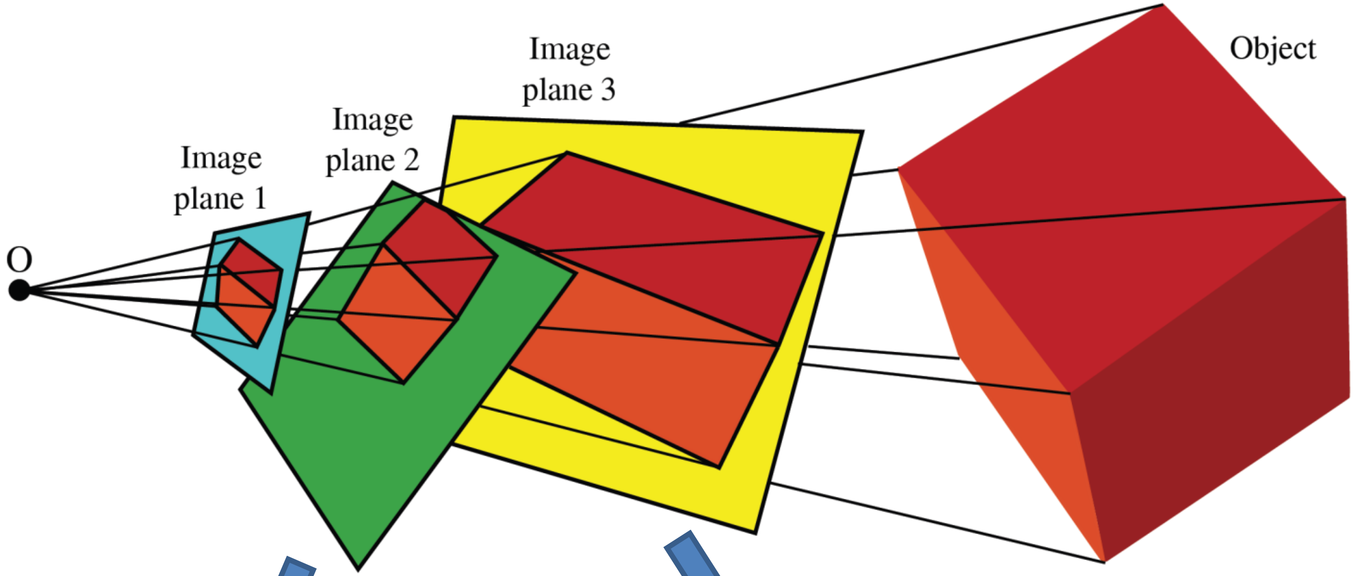
Moving camera observing a plane



More detail next

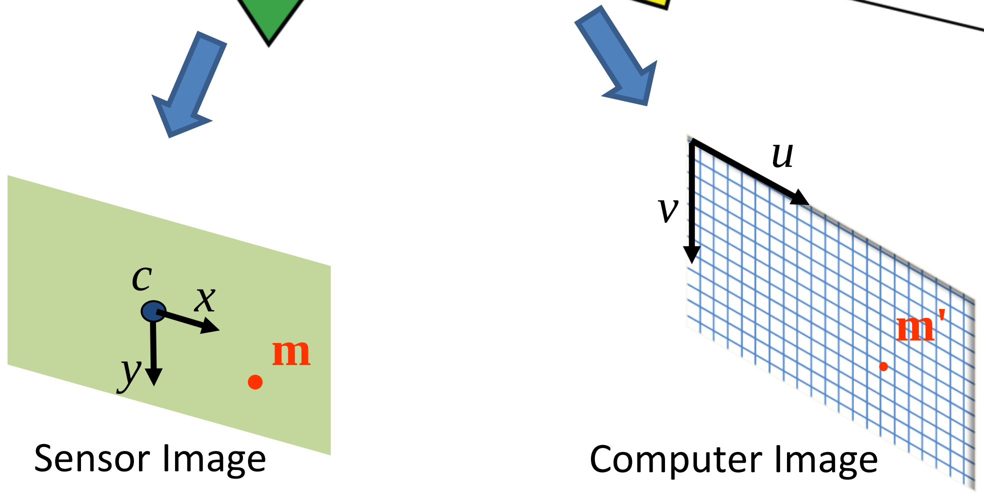
3. 2D Homography

Camera still (different image planes)



There is a homography H_{3x3} between points of all the 3 image planes

Example:



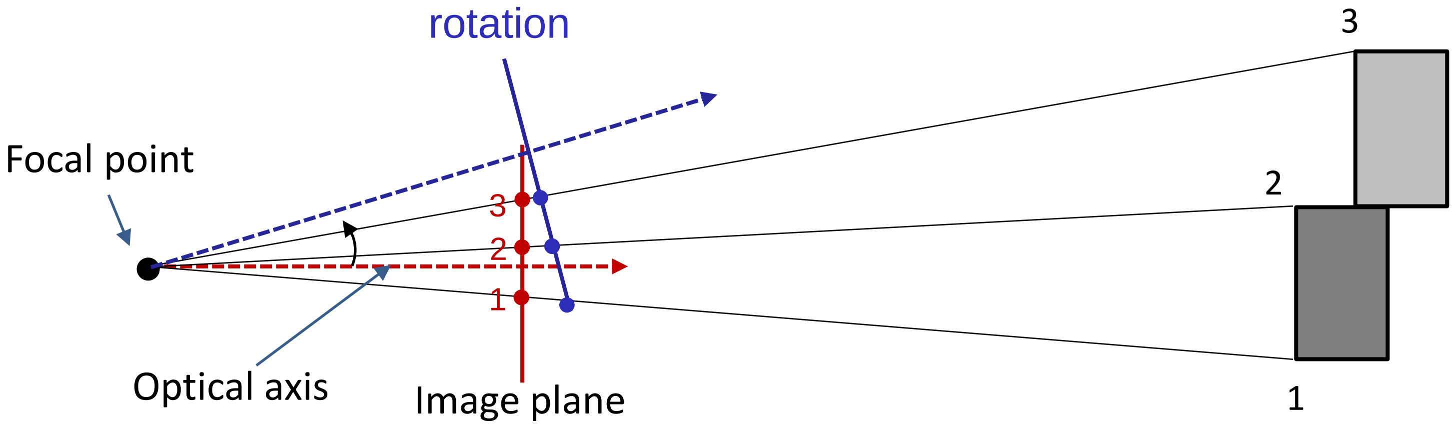
The **Zoom** is a simple homography

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

3. 2D Homography

Rotating camera observing a general (nonplanar) scene

2D Homography applies for images of the same (nonplanar) scene **without moving the focal point**, i.e. only rotation and/or zoom are allowed



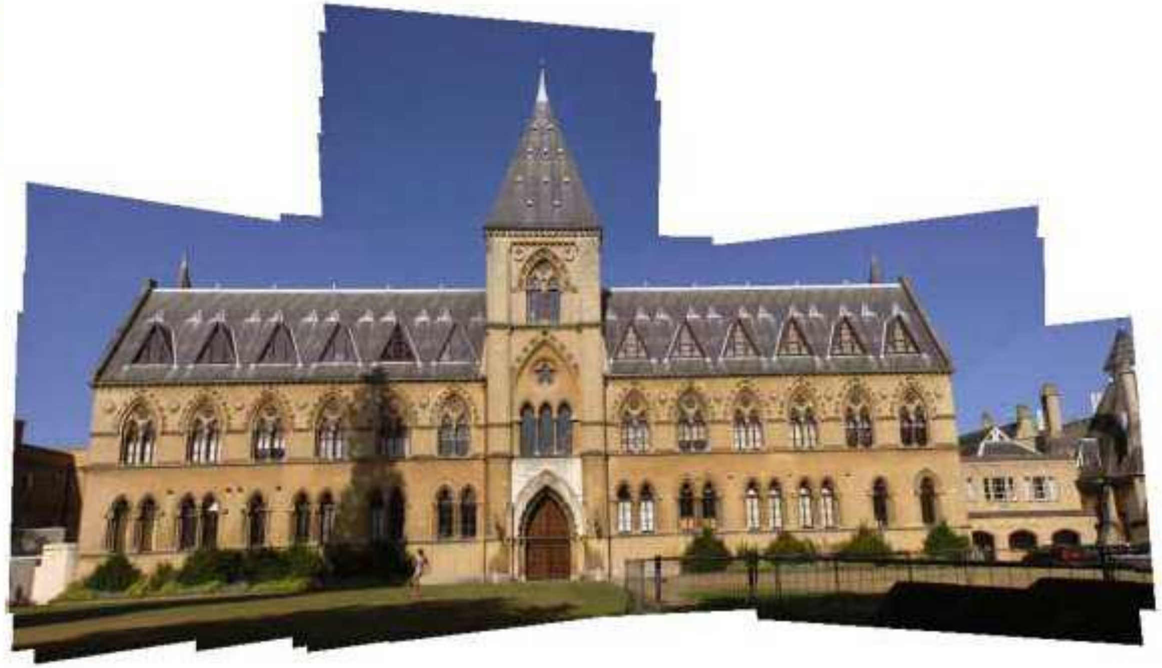
When the **camera rotates**, the projection rays of the points 1,2,3 do not move in space, only their projections on the image.

3. 2D Homography

Rotating camera observing a general (nonplanar) scene

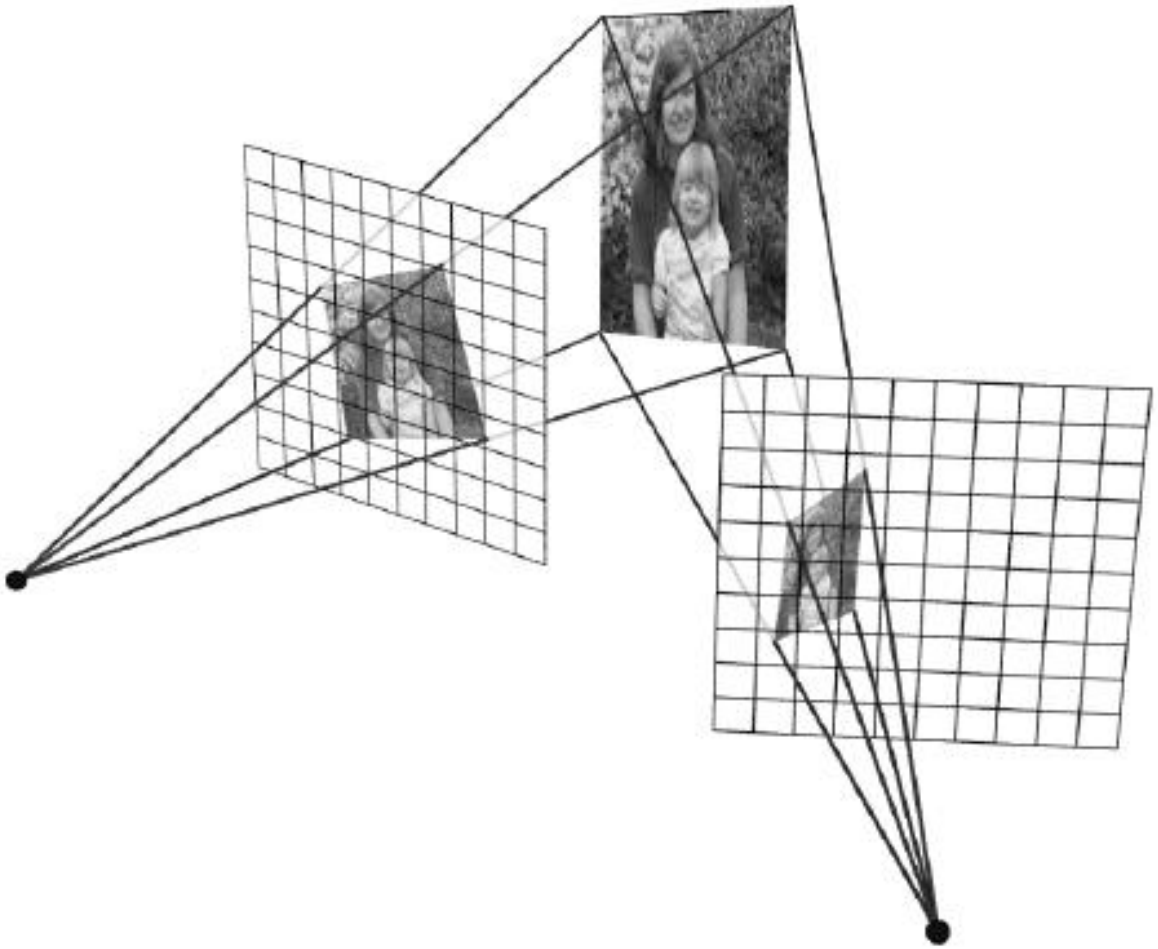
Application: Panoramic of a scene through image stitching (common *app* in mobile phones)

It doesn’t mind if the scene is not planar, as long as the focal point remains still (or negligible movement in comparison to the scene distance)



3. 2D Homography

Moving camera observing a plane



from R. Szeliski, S. Seitz, D. Hoiem, and I. Kemelmacher

3. 2D Homography

Moving camera observing a plane

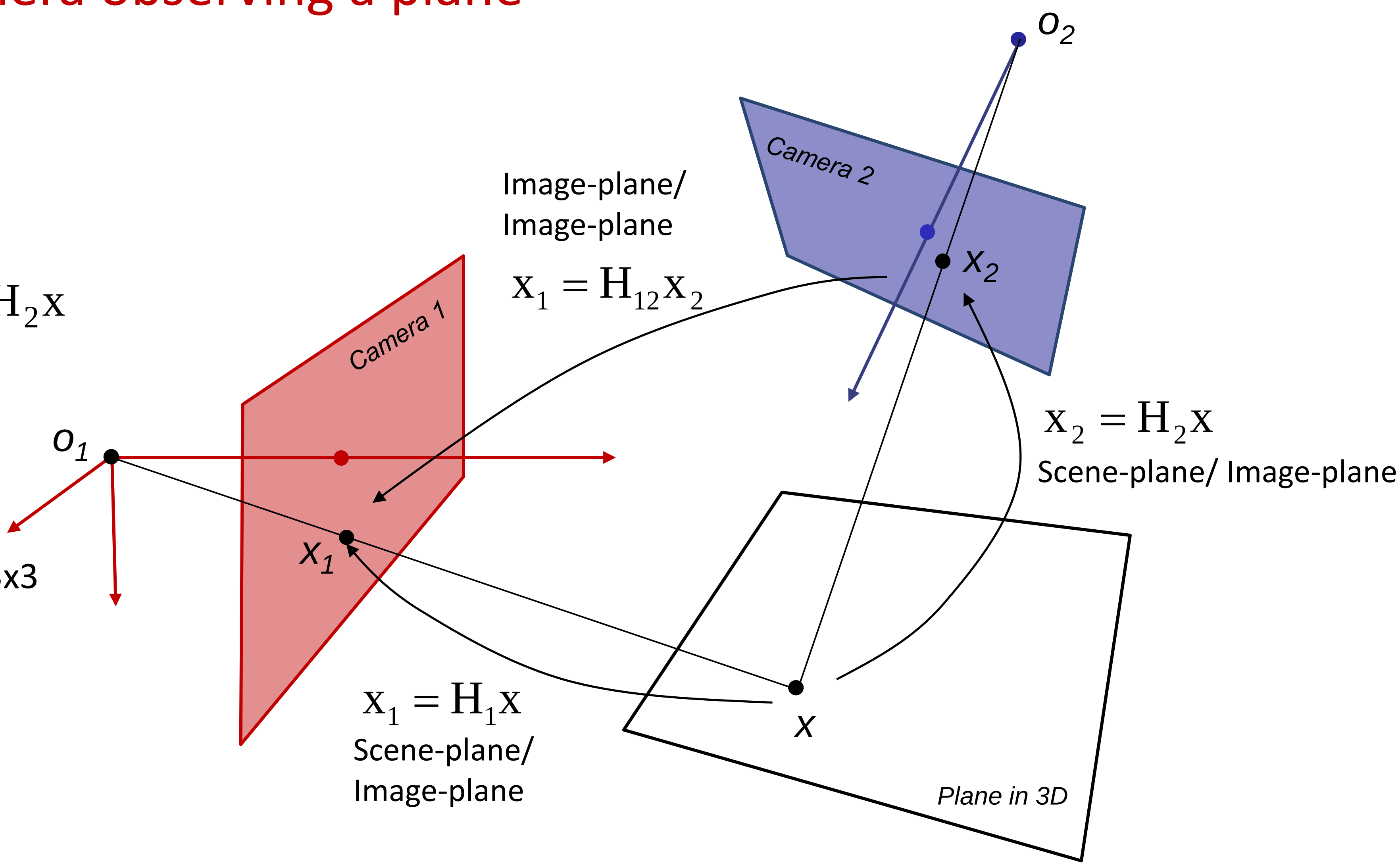
$$x_2 = H_2 x$$

$$H_1 x = x_1 = H_{12} x_2 = H_{12} H_2 x$$

$$\Downarrow$$

$$H_1 = H_{12} H_2$$

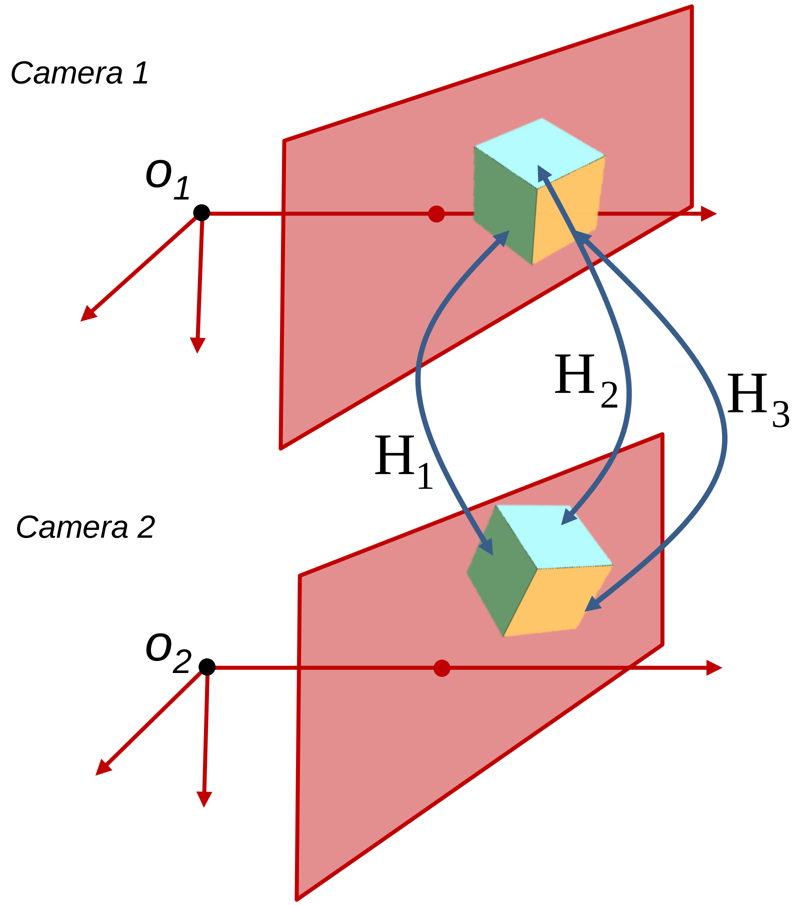
Still a Homography matrix 3x3

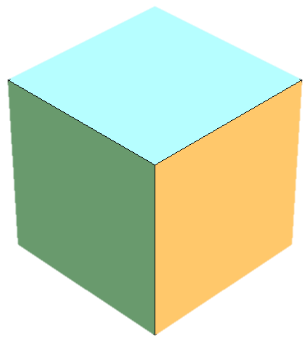


3. 2D Homography

Moving camera observing a **non-planar** structure

There is a homography between each pair of corresponding planes in the two images





3. 2D Homography

Solving the homography 2D: Direct Linear Transformation (**DLT**)

For each pair of points <x_i x_i'> in correspondence we set a linear equation system (**2 eq., 8 unknowns**):

$$\lambda \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

Minimum of 4 points needed

$$x_i' = \frac{h_{00}x_i + h_{01}y_i + h_{02}}{h_{20}x_i + h_{21}y_i + h_{22}}$$

$$y_i' = \frac{h_{10}x_i + h_{11}y_i + h_{12}}{h_{20}x_i + h_{21}y_i + h_{22}}$$

$$x_i'(h_{20}x_i + h_{21}y_i + h_{22}) = h_{00}x_i + h_{01}y_i + h_{02}$$

$$y_i'(h_{20}x_i + h_{21}y_i + h_{22}) = h_{10}x_i + h_{11}y_i + h_{12}$$

$$\begin{bmatrix} 0 & 0 & 0 & -x_i & -y_i & -1 & y_i'x_i & y_i'y_i & y_i' \\ -x_i & -y_i & -1 & 0 & 0 & 0 & x_i'x_i & x_i'y_i & x_i' \end{bmatrix} \begin{pmatrix} h_{00} \\ h_{01} \\ h_{02} \\ \vdots \\ h_{20} \\ h_{21} \\ h_{22} \end{pmatrix}_{1x9} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

<u>Note</u>: Any scale of this vector is also a solution.

Solving the homography 2D: Direct Linear Transformation (DLT) (cont.)

For n pairs of correspondences:

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1x_1 & -y'_1y_1 & -y'_1 \\ & & & & & \vdots & & & \\ x_n & y_n & 1 & 0 & 0 & 0 & -x'_nx_n & -x'_ny_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -y'_nx_n & -y'_ny_n & -y'_n \end{bmatrix} \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

$\boldsymbol{A}_{2nx9}$ (known) $\quad$ $\mathbf{h}_{9x1}$ (unknown)

This is a Homogeneous eq. system $\mathbf{Ah} = \mathbf{0}$

- For n = 4 independent points, (**Rank(A) = 8**) , there is a solution k**h** with $\mathbf{h} \neq 0$ $\mathrm{k} \neq 0$

  To remove the dof (k) we can either fix an element of **h** (h_{22} = 1) or (better) solve for $||\mathbf{h}|| = 1$

- For n < 4 independent points (**Rank(A) < 8**), there are infinitely many solutions beyond k**h**

- For n > 4 points (**Rank(A) = 9**), there is no solution (apart from **h** =0) and we solve:

  Arg.min$_h$ $||\mathbf{Ah}||^2$ with $||\mathbf{h}|| = 1$

  Solution $\hat{\mathbf{h}}$: eigenvector of the smallest eigenvalue of $\boldsymbol{A}^T\boldsymbol{A}$

Using more points gives robustness to noise in the point coordinates

PROBLEM:

$$\text{Arg.min}_h \; ||\mathbf{Ah}||^2 \;\; \text{subject to} \; ||\mathbf{h}|| = 1$$

This is a problem that we will find very often in CV, when solving for

1. Homography H_{3x3}
2. Projection matrix P_{3x4}
3. Fundamental matrix F_{3x3}

Solution $\hat{\mathbf{h}}$:

- eigenvector of the smallest eigenvalue of $\mathbf{A}^T\mathbf{A}$

Or equivalently,

- singular vector associated to the smallest singular value of **A** → Preferred, since more efficient computing SVD(**A**) than eigen($\mathbf{A}^T\mathbf{A}$)

PROBLEM:

$\text{Arg.min}_h \, \|\mathbf{Ah}\|^2$ subject to $\|\mathbf{h}\| = 1$

Solution $\hat{\mathbf{h}}$: eigenvector of the smallest eigenvalue of $\mathbf{A}^T\mathbf{A}$

Just for completeness.
Not for the exam

WHY?

$$\|\mathbf{Ah}\|^2 = (\mathbf{Ah})^{\mathrm{T}}(\mathbf{Ah}) = \mathbf{h}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{Ah} = [h_1 \ \dots h_n]\, \boldsymbol{V\Sigma V}^{\mathrm{T}}\, [h_1 \ \dots h_n]^T$$

$\|c\|^2 = c^T c$

$$\mathrm{eigen}\left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right) = \boldsymbol{V\Sigma V}^{\mathrm{T}} = [\boldsymbol{v}_1 \ \dots \boldsymbol{v}_n]\boldsymbol{\Sigma}[\boldsymbol{v}_1^T \ \dots \boldsymbol{v}_n^T] = \sum_{i=1}^{n} \sigma_i \boldsymbol{v}_i \boldsymbol{v}_i^T$$

$$\|\mathbf{Ah}\|^2 = [h_1 \ \dots h_n]\boldsymbol{V\Sigma V}^{\mathrm{T}}\ [h_1 \ \dots h_n]^T = \sum_{i=1}^{n} \sigma_i \boldsymbol{h}^{\mathrm{T}}\boldsymbol{v}_i \boldsymbol{v}_i^{\mathrm{T}}\mathbf{h} = \sum_{i=1}^{n} \sigma_i \left\|\boldsymbol{v}_i^T\mathbf{h}\right\|^2$$

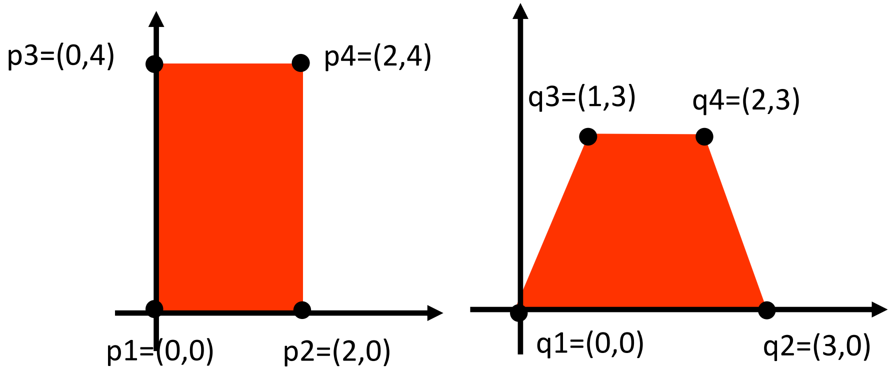
$\|\mathbf{h}\| = 1, \left\|\boldsymbol{v}_i^{\mathbf{T}}\right\| = 1, \forall i$ Vectors with length 1

If $\mathbf{h} = \boldsymbol{v}_i$ ($\boldsymbol{v}_i$ oriented in the direction of $\mathbf{h}$) $\rightarrow \left\|\boldsymbol{v}_i^T\mathbf{h}\right\| = \mathbf{1}, \left\|\boldsymbol{v}_j^T\mathbf{h}\right\| = \mathbf{0}\ \forall j \neq i$

$\boxed{\min \|\mathbf{Ah}\|^2 = \sigma_n \text{ for } \mathbf{h} = \boldsymbol{v}_n}$ the eigenvector of the smallest eigenvalue

3. 2D Homography

Example:



$$\lambda \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

Recall: At least 4 pairs of points needed
(for 8 unknowns)

$$H = \begin{matrix} 0.4242 & 0.2121 & 0.0000 \\ -0.0000 & 0.6363 & 0.0000 \\ 0.0000 & 0.1414 & 0.2828 \end{matrix} = \begin{matrix} 1.5000 & 0.7500 & \boxed{0.0000} \\ -0.0000 & 2.2500 & \boxed{0.0000} \\ 0.0000 & 0.5000 & 1.0000 \end{matrix}$$

$\|\mathbf{h}\| = 1$

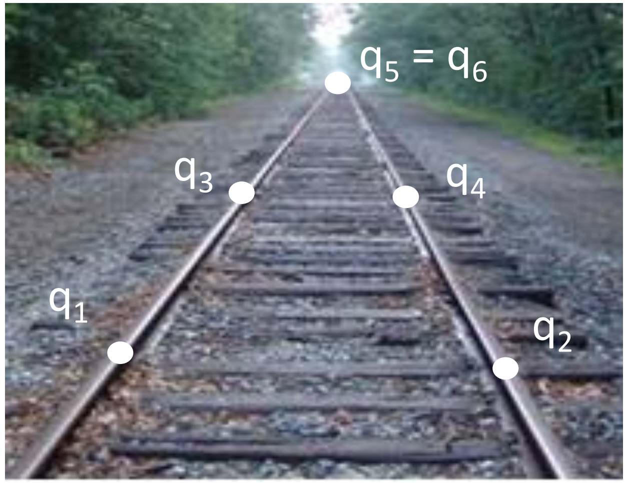
H./H(3,3)

No translation

```
% At least 4 pairs of points needed
P = [0 0 1; 2 0 1; 0 4 1; 2 4 1]';
Q = [0 0 1; 3 0 1; 1 3 1; 2 3 1]';
npoints = size(P,2); %4 in this example
H = homography2d(P,Q); %Transformation
from P --> Q (Q =H*P)
% Now we can transform any point p <-> q
% q = H*p;
% q_nh(1:2) = q(1:2)/q(3);
% Or p =inv(H)*q;
```

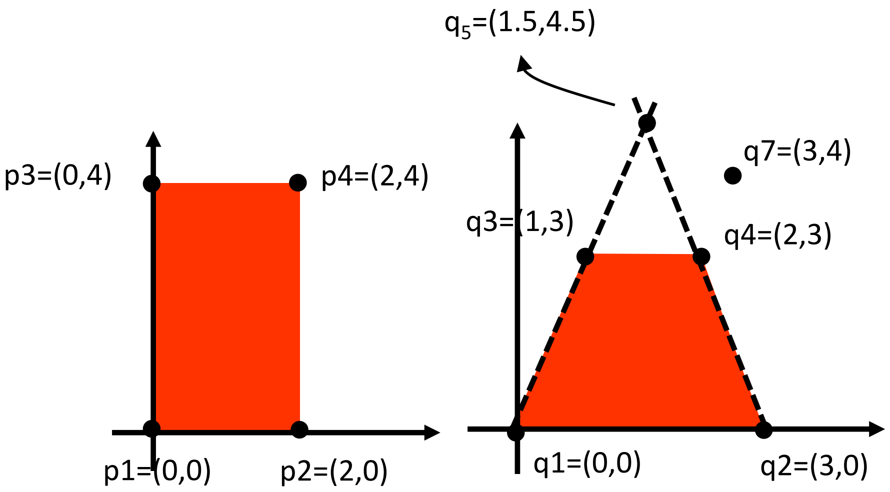
3. 2D Homography

Points at infinity



- q_5,q_6 are at infinity in 3D
- If we try to compute the homography H with {q_1,q_2,q_5,q_6} it becomes singular.

But, given H we can work with points at infinity, i.e. q_5:



- Where is q=(1.5,4.5) transformed to?

$$H\begin{bmatrix}1.5\\4.5\\1\end{bmatrix}=\begin{bmatrix}0\\7.07\\0\end{bmatrix} \Rightarrow \begin{bmatrix}0\\7.07\\0\end{bmatrix}\equiv\begin{bmatrix}0\\1\\0\end{bmatrix}$$

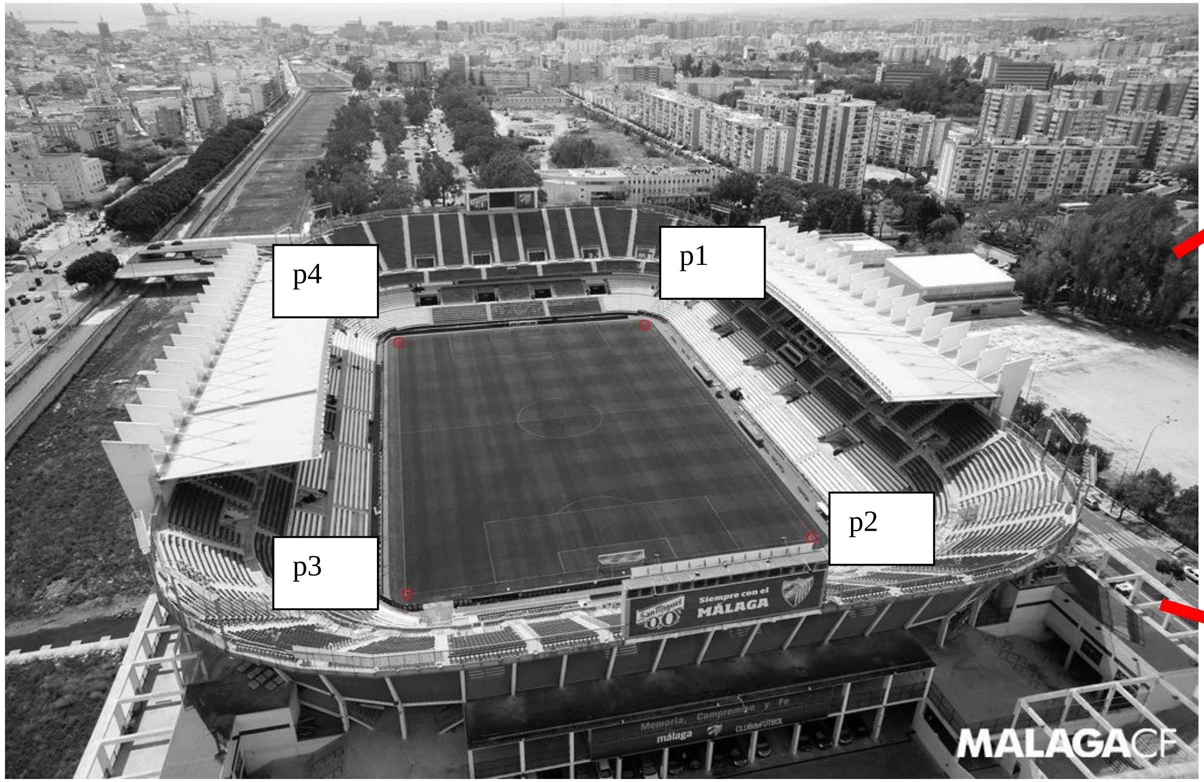
Point at infinity along the vertical axis ([0 1]^T)

- Where is q7=(3,4) transformed to?

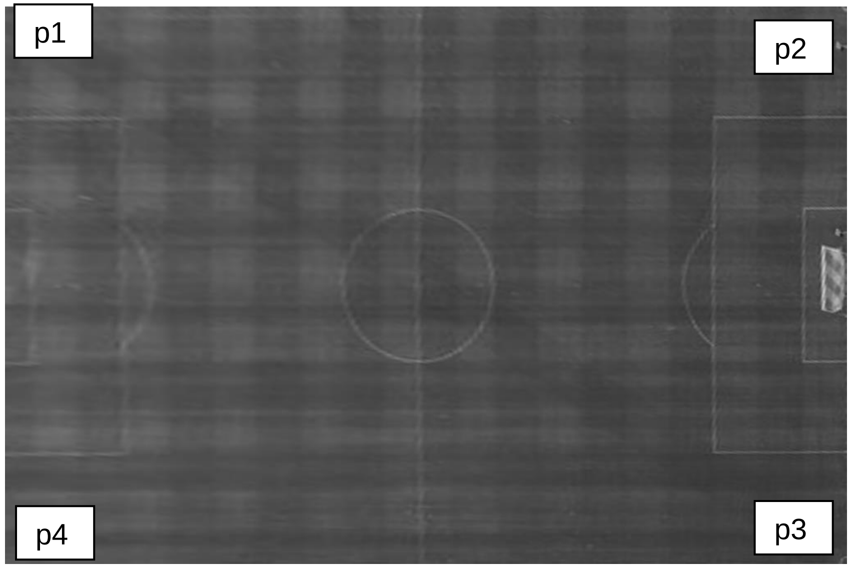
  Out of the rectangle (do it by yourself)

3. 2D Homography

Practical example: Undistort images



Undistort image

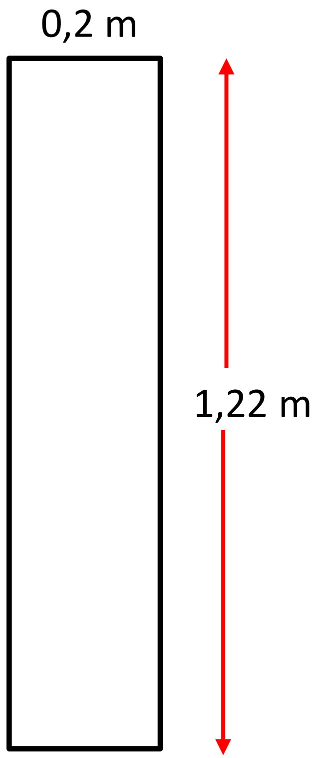


Distort image

3. 2D Homography

Practical example: Distance to fouling on the board in long jump

Image projection



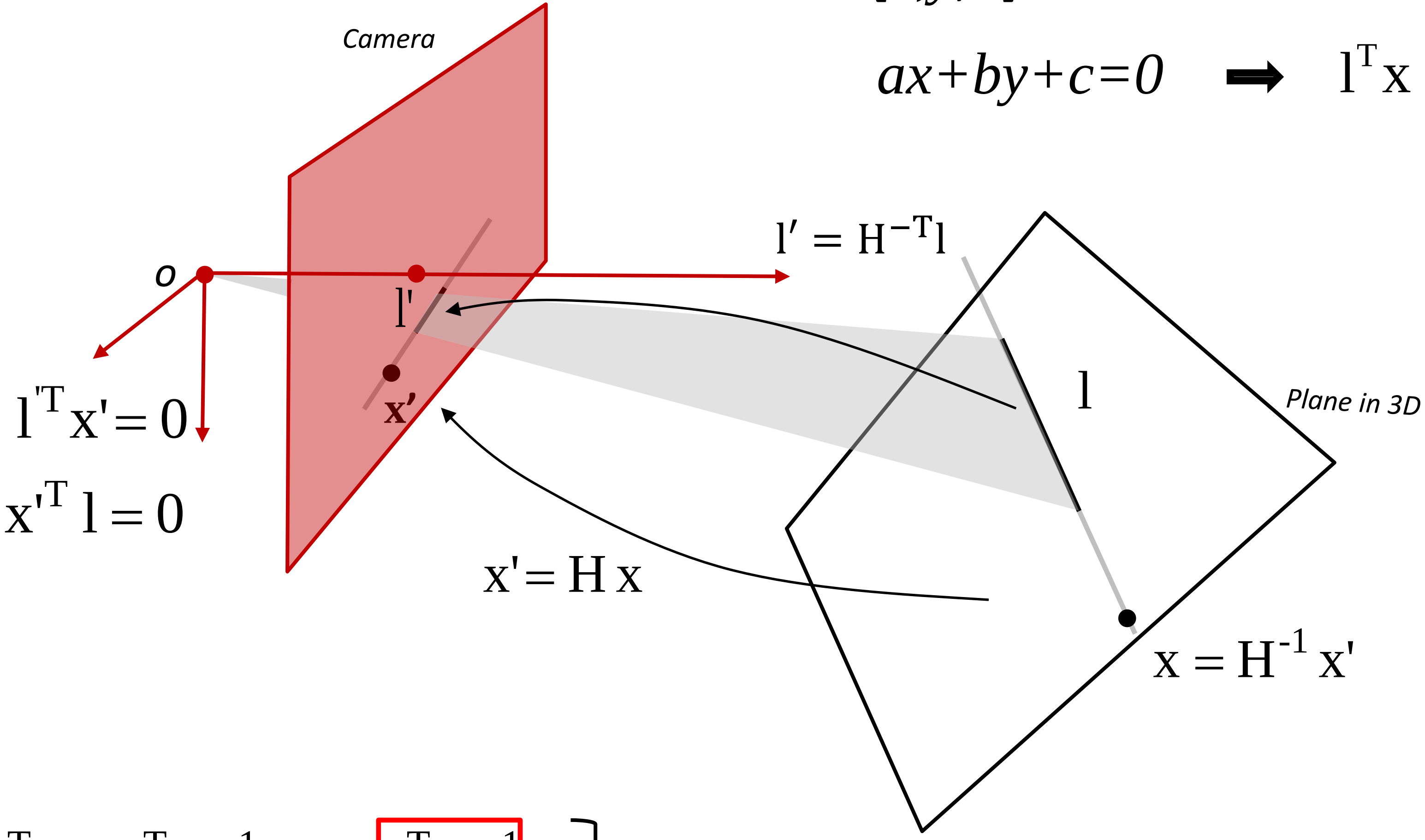
In real world

3. 2D Homography

Homography of lines

If $\mathrm{x} = [x,y,1]^T$ is on the line $\mathrm{l} = [a,b,c]^T$:

$$ax+by+c=0 \quad \Rightarrow \quad \mathrm{l}^T\mathrm{x} = \mathrm{x}^T\mathrm{l} = 0$$



$$\left.\begin{aligned} 0 &= \mathrm{l}^T\mathrm{x} = \mathrm{l}^T\mathrm{H}^{-1}\mathrm{H}\mathrm{x} = \boxed{\mathrm{l}^T\mathrm{H}^{-1}}\mathrm{x}' \\ 0 &= \boxed{\mathrm{l}'^T}\mathrm{x}' \end{aligned}\right\} \mathrm{l}^T\mathrm{H}^{-1} = \mathrm{l}'^T \quad \Rightarrow \quad \mathrm{l}^T = \mathrm{l}'^T\mathrm{H} \quad \Rightarrow$$

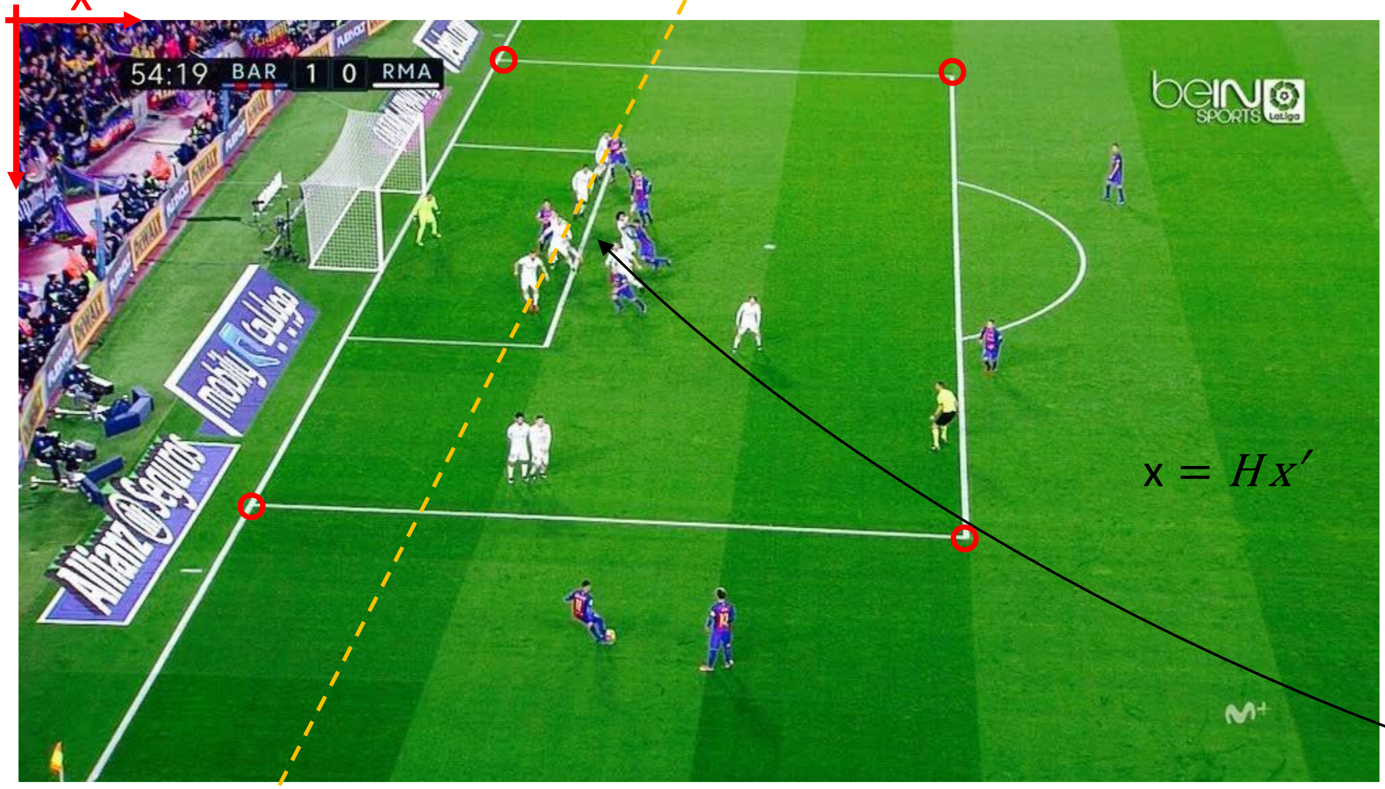
Summary

$$\mathrm{x}' = \mathrm{H}\mathrm{x}$$

$$\mathrm{l}' = \mathrm{H}^{-T}\mathrm{l}$$

Homography of lines. Example

f(x,y)



	-1.64	0.50	-881.93
H =	-0.05	-0.40	-79.14
	0.00	0.00	-2.48

$$x = H\,x'$$

(136, 784)

$$l = H^T l'$$

f’(x,y)

$$l = \begin{bmatrix} -0.59 \\ -0.23 \\ 272.01 \end{bmatrix}$$

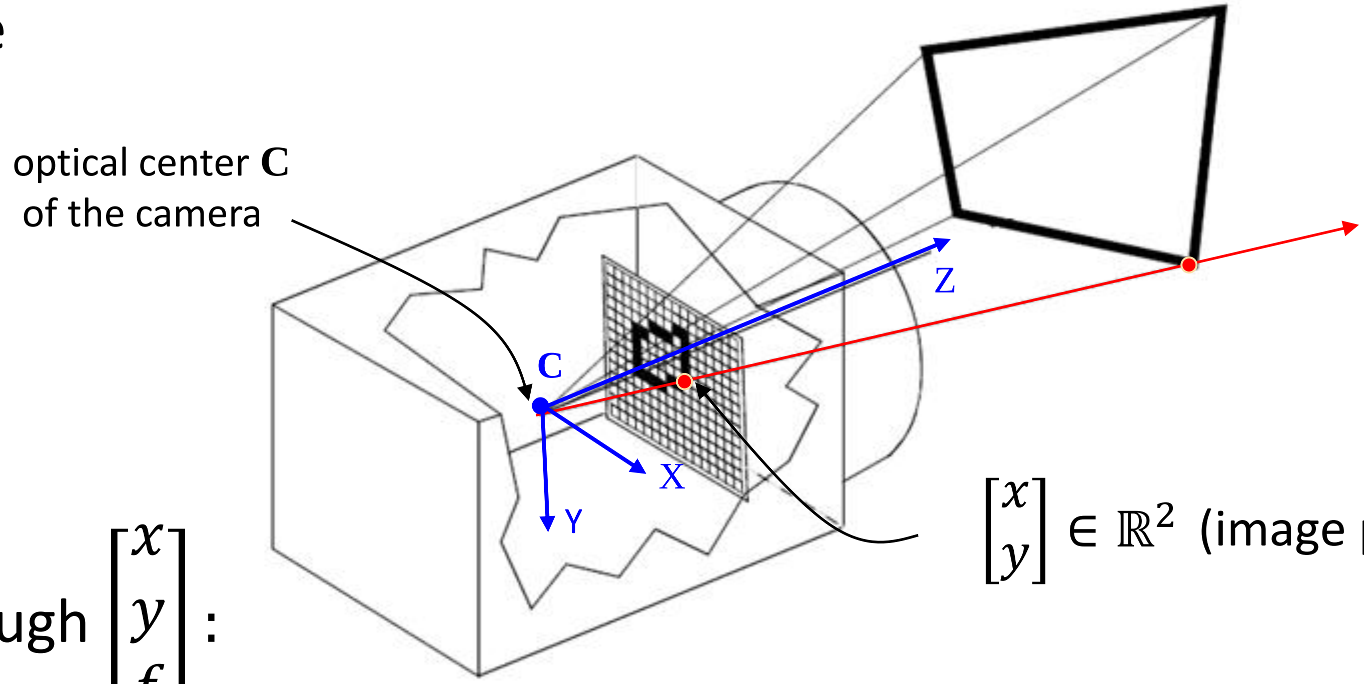
1. With 4 points, estimate H
2. Rectify image f(x,y) with H
3. Set vertical line l’ in f’(x,y)
4. $l = H^{-T} l'$

$$l' = \begin{bmatrix} 1 \\ 0 \\ -136 \end{bmatrix} \longleftarrow x' - 136 = 0$$

4. The Pinhole model

From 2D to 3D:

Given a point in the image plane, its projection line in **the camera frame** is very simple to compute



$\begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}^2$ (image plane)

Projection line through $\begin{bmatrix} x \\ y \\ f \end{bmatrix}$:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{C} + k(\mathbf{P} - \mathbf{C}) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + k\left(\begin{bmatrix} x \\ y \\ f \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}\right) = k\begin{bmatrix} x \\ y \\ f \end{bmatrix} \in P^2$$

P^2 is called the projective plane: Set of all lines in $\Re^3$ passing through the origin (0, 0, 0).

4. The Pinhole model

From 2D to 3D:

$$\mathrm{m} = \begin{bmatrix} x \\ y \end{bmatrix} \in \Re^2 \Rightarrow \mathrm{M} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = k \begin{bmatrix} x \\ y \\ f \end{bmatrix}$$

Point in the image plane

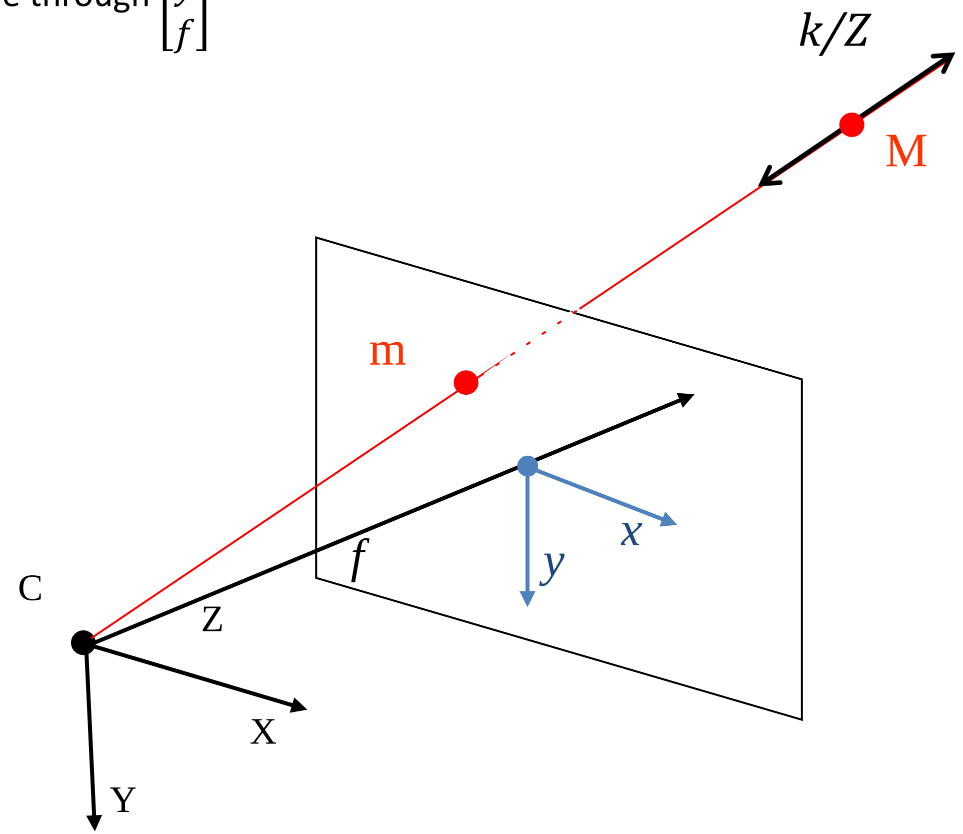
Projection line through $\begin{bmatrix} x \\ y \\ f \end{bmatrix}$

For k=1 we have a 3D point in the sensor image: $\begin{bmatrix} x \\ y \\ f \end{bmatrix}$

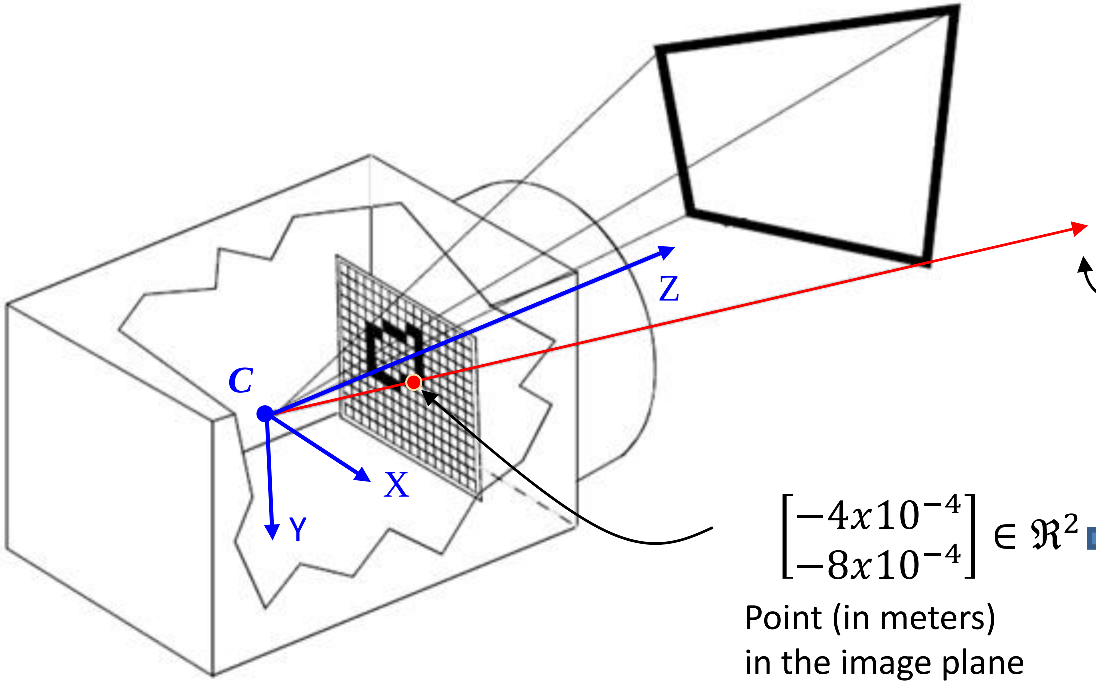
With Z (instead of k) as parameter:

$$\mathrm{M} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = Z \begin{bmatrix} \frac{x}{f} \\ \frac{y}{f} \\ 1 \end{bmatrix} \qquad Z = kf$$

For Z=f (k=1) we have: $\begin{bmatrix} x \\ y \\ f \end{bmatrix}$



From 2D to 3D (Example):



Projection line

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = k \begin{bmatrix} -4x10^{-4} \\ -8x10^{-4} \\ f \end{bmatrix} = Z \begin{bmatrix} -\frac{4}{3}10^{-1} \\ -\frac{8}{3}10^{-1} \\ 1 \end{bmatrix}$$

$Z = kf$ Focal length

f = 3x10^{-3} m

$$\begin{bmatrix} -4x10^{-4} \\ -8x10^{-4} \end{bmatrix} \in \Re^2 \Rightarrow \begin{bmatrix} -4x10^{-4} \\ -8x10^{-4} \\ f \end{bmatrix} \in \Re^3$$

Point (in meters)
in the image plane

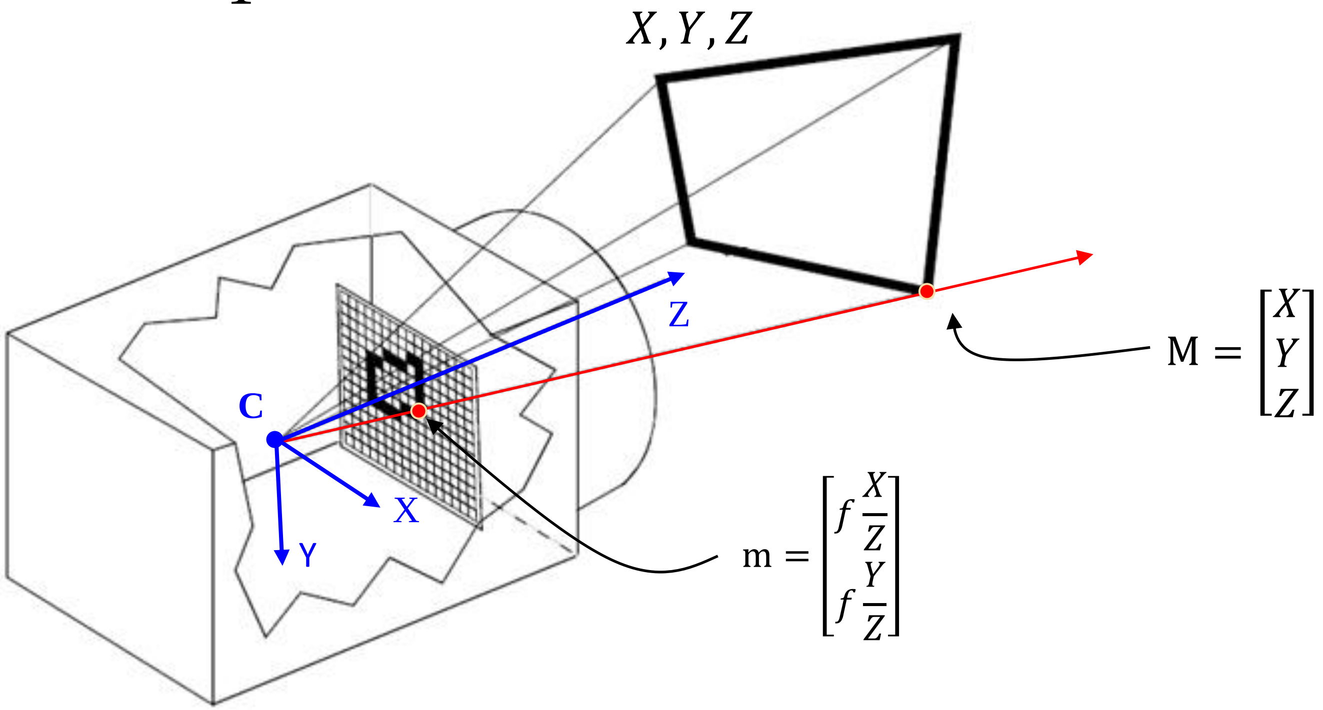
Coordinates of the point in
the *camera system* (blue)

As a function of Z
(Projection line)

Now, from 3D to 2D:

Any 3D point lays in the projection line of certain 2D image plane (for some *Z)*:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = Z \begin{bmatrix} \frac{x}{f} \\ \frac{y}{f} \\ 1 \end{bmatrix} \text{ for some } Z \quad \Longrightarrow \quad \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f\,\frac{X}{Z} \\ f\,\frac{Y}{Z} \end{bmatrix} \in \Re^2$$



2. The Pinhole model

From 3D to 2D:

Given a 3D point in **the camera frame** its projection on the image plane is:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \in \mathbb{R}^3 \Rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f\frac{X}{Z} \\ f\frac{Y}{Z} \end{bmatrix} \in \mathbb{R}^2$$

$x = f_1(X, Y, Z), x = f_2(X, Y, Z)$ are non-linear functions on X, Y, Z

But becomes **LINEAR** when using homogeneous coordinates:

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

Recall homogeneous coordinates:

From cartesian to homogeneous

$$\begin{bmatrix} x \\ y \end{bmatrix} \to \lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

From homogeneous to cartesian

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \to \begin{bmatrix} x = a/c \\ y = b/c \end{bmatrix}$$

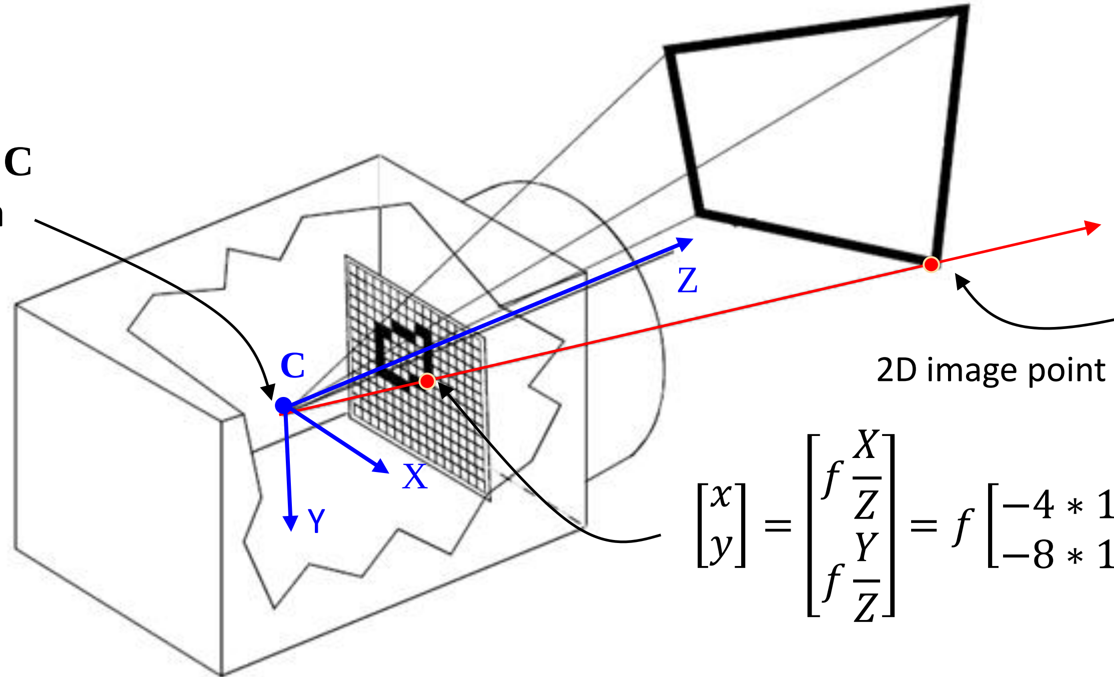
From 3D to 2D:

Given a 3D point in **the camera frame** its projection on the image plane is:

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

<u>Example:</u>

optical center **C**
of the camera



$\begin{bmatrix} -0.12 \\ -0.24 \\ 3x10^3 \end{bmatrix} \in \Re^3$ 3D point **in the camera system**

2D image point

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f\dfrac{X}{Z} \\ f\dfrac{Y}{Z} \end{bmatrix} = f\begin{bmatrix} -4 * 10^{-4} \\ -8 * 10^{-4} \end{bmatrix}$$

$$\begin{bmatrix} Zx \\ Zy \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} -0.12 \\ -0.24 \\ 3x10^3 \\ 1 \end{bmatrix}$$

4. The Pinhole model

Summary:

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

3D - 2D

given $\begin{bmatrix}X\\Y\\Z\end{bmatrix} \longrightarrow \begin{bmatrix}x\\y\end{bmatrix}=\begin{bmatrix}f\,\frac{X}{Z}\\ f\,\frac{Y}{Z}\end{bmatrix}\in\mathbb{R}^2$

2D - 3D

given $\begin{bmatrix}x\\y\end{bmatrix} \longrightarrow \begin{bmatrix}X\\Y\\Z\end{bmatrix}=Z\begin{bmatrix}\frac{x}{f}\\ \frac{y}{f}\\ 1\end{bmatrix}\in P^2$

Usually decomposed as:

$$\underbrace{Z\begin{bmatrix}x\\y\\1\end{bmatrix}}_{\tilde{\mathbf{m}}}=\underbrace{\begin{bmatrix}f&0&0\\0&f&0\\0&0&1\end{bmatrix}}_{\mathbf{K_f}}\underbrace{\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\end{bmatrix}}_{\mathbf{P}_0=[\mathbf{I}|0]}\underbrace{\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}}_{\tilde{\mathbf{M}}}$$

2D scale

Normalized projection (f=1)

Perspective projection :

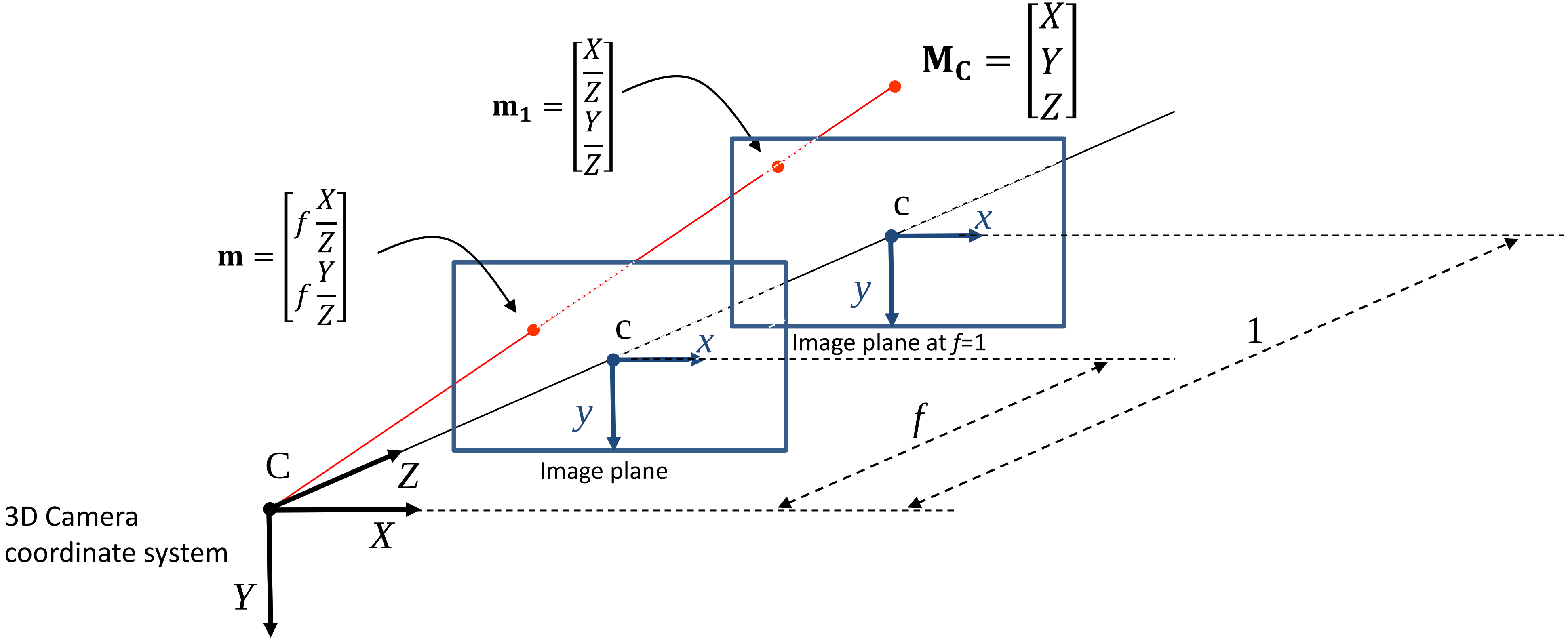
$$Z\tilde{\mathbf{m}}=\mathbf{K_f}\mathbf{P_0}\tilde{\mathbf{M}}_C$$

The decomposition of the Pinhole model

$$\underbrace{Z\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\tilde{\mathbf{m}}} = \underbrace{\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K_f}} \overbrace{\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{P}_0 = [\mathbf{I}\,|0]} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}^{\tilde{\mathbf{m}}_1}$$

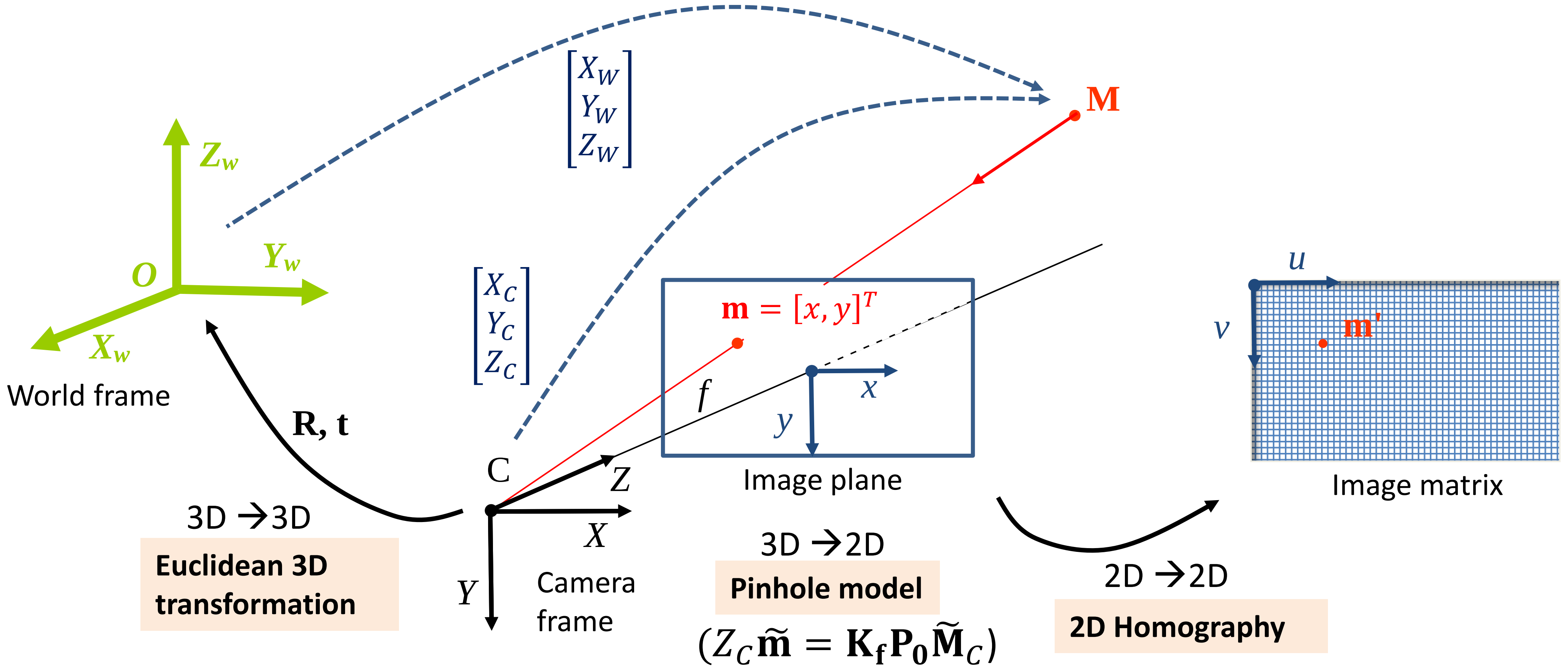
Perspective projection :

$$Z\tilde{\mathbf{m}} = \mathbf{K_f}\underbrace{\mathbf{P_0}\tilde{\mathbf{M}}_C}_{Z\tilde{\mathbf{m}}_1}$$



Pinhole model still useless:

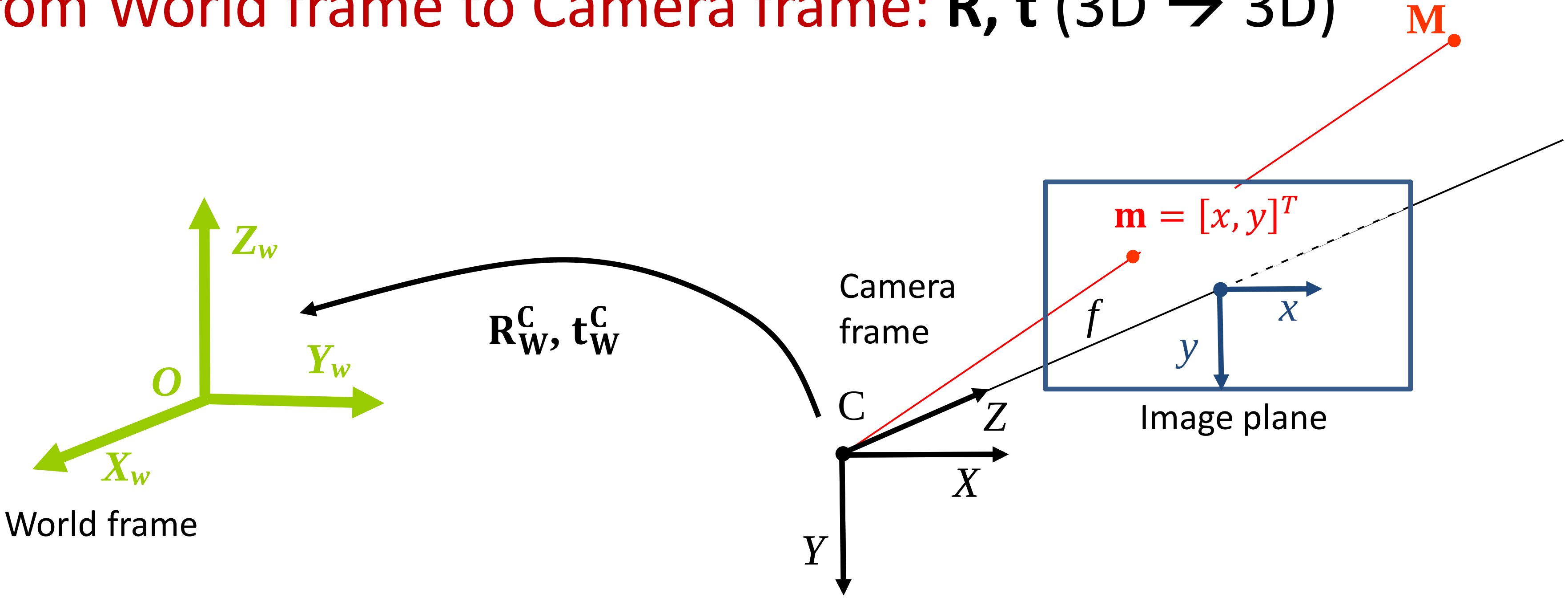
1. **M** is known in a World frame, not in the Camera frame
2. **m** is known in image pixels, not in the sensor plane (coordinates in meters)



The Camera Model adds to the pinhole Model 1) an Euclidean 3D transformation and 2) a 2D Homography

5. The camera model

From World frame to Camera frame: **R, t** (3D → 3D)



$$\mathbf{M_c} = \mathbf{R}_{\mathbf{W}}^{\mathbf{C}}\mathbf{M_w} + \mathbf{t}_{\mathbf{W}}^{\mathbf{C}} \xrightarrow{\text{In homogeneous coordinates}} \widetilde{\mathbf{M}}_{\mathbf{c}} = \mathbf{D}_{\mathbf{W}}^{\mathbf{C}}\widetilde{\mathbf{M}}_{\mathbf{w}} \quad \text{with} \quad \mathbf{D} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_3^{\mathbf{T}} & 1 \end{bmatrix} \quad \mathbf{0}_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The new **Perspective Projection** equation:

$$\lambda\widetilde{\mathbf{m}} = \mathbf{K_f}\mathbf{P_0}\widetilde{\mathbf{M}}_{\mathbf{c}} = \mathbf{K_f}\mathbf{P_0}\mathbf{D}\widetilde{\mathbf{M}}_{\mathbf{w}} \qquad \text{with } \lambda = Z_C$$

The camera model

From sensor image to computer image: 2D → 2D

We don't have access to the point coordinates on the sensor (units in *meters*) but to the image matrix coordinates (units: *raw, column*) in the computer

Computer image matrix (units in pixels)

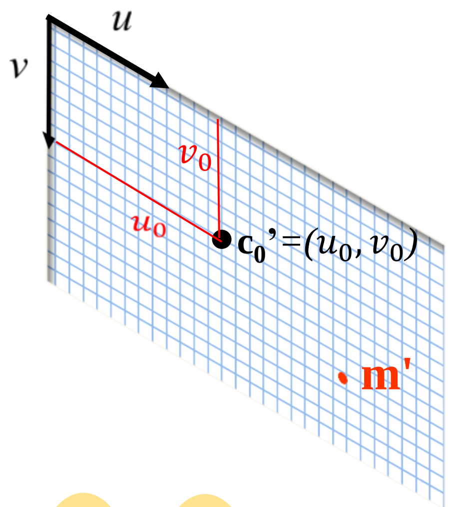
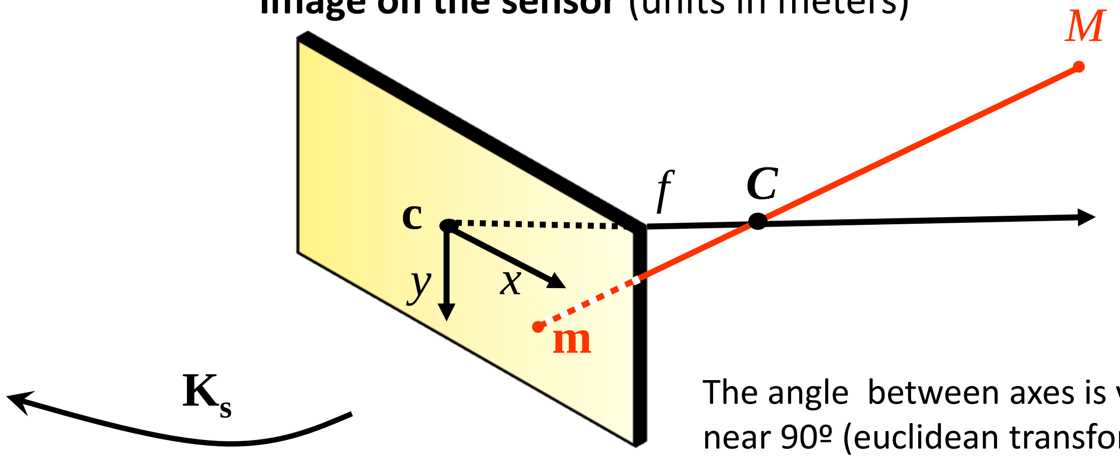


Image on the sensor (units in meters)



Both images are related by an **AFFINE transformation**

The angle between axes is very near 90º (euclidean transformation)

AFFINE transformation:

$$\mathbf{m}' = \mathbf{A}\mathbf{m} + \mathbf{b} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\text{meter}} + \underbrace{\begin{bmatrix} u_0 \\ v_0 \end{bmatrix}}_{\text{pixels}}$$

In homogeneous

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_x & s & u_0 \\ 0 & k_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \approx \begin{bmatrix} s_x & 0 & u_0 \\ 0 & s_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\tilde{\mathbf{m}}' = \mathbf{K}_s \tilde{\mathbf{m}}$$

Perspective matrix: Camera model

Pinhole model:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix}$$

$$\lambda \tilde{\mathbf{m}} = \mathbf{K_f P_0} \widetilde{\mathbf{M}}_C$$

In the Image/World (Camera model):

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_x & 0 & u_0 \\ 0 & k_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_3^{\mathbf{T}} & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \underbrace{f k_x}_{s_x} & 0 & u_0 \\ 0 & \underbrace{f k_y}_{s_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{R}\ \mathbf{t}] \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix}$$

$$\lambda \tilde{\mathbf{m}}' = \mathbf{K}_{3x3} [\mathbf{R}\ \mathbf{t}]_{3x4} \widetilde{\mathbf{M}}_W$$

Perspective matrix: General form

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} s_x & 0 & u_0 \\ 0 & s_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} [\mathbf{R}\ \mathbf{t}] \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix}$$

Scale + translation

$$\lambda \widetilde{\mathbf{m}}' = \mathbf{K}[\mathbf{R}\ \mathbf{t}]\widetilde{\mathbf{M}}_W$$

$$\lambda \widetilde{\mathbf{m}}' = \begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \mathbf{K}[\mathbf{R}\ \mathbf{t}]\widetilde{\mathbf{M}}_W = \mathbf{P}\widetilde{\mathbf{M}}_W = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix}$$

P=K[R t] is a homogeneous matriz → 11 d.o.f..

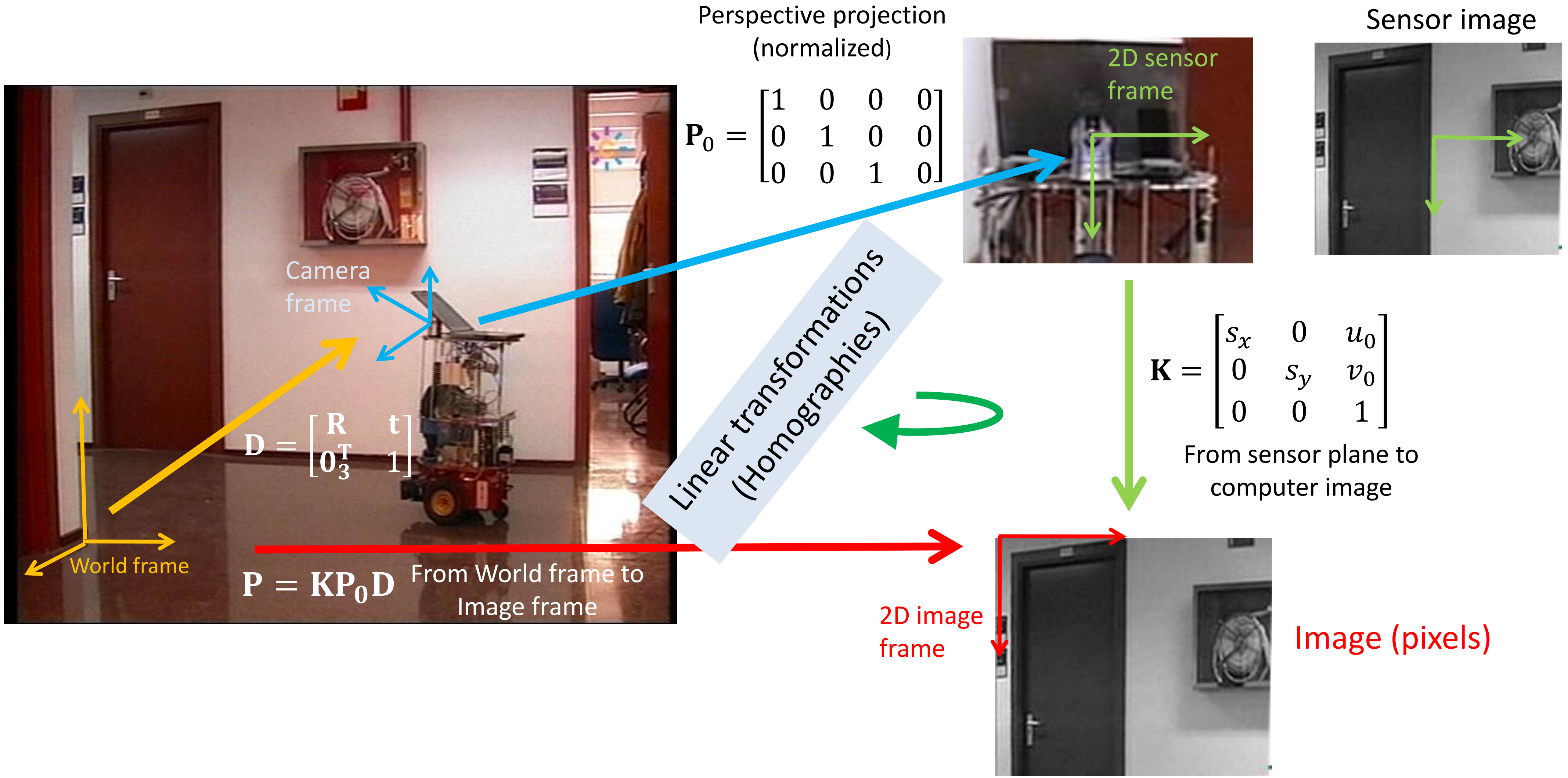
$$u = \frac{p_{11}X_W + p_{12}Y_W + p_{13}Z_W + p_{14}}{p_{31}X_W + p_{32}Y_W + p_{33}Z_W + p_{34}}$$

$$v = \frac{p_{21}X_W + p_{22}Y_W + p_{23}Z_W + p_{24}}{p_{31}X_W + p_{32}Y_W + p_{33}Z_W + p_{34}}$$

Two non-linear equations

5. The camera model

The camera model in action



5. The camera model

The whole camera model in action (+ lens distortion)