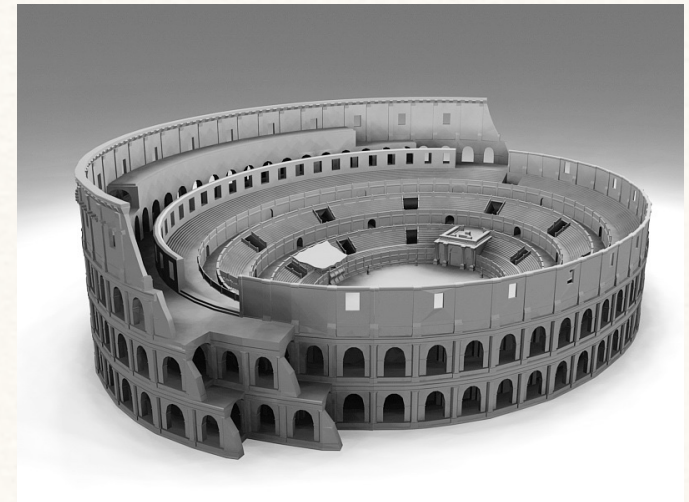
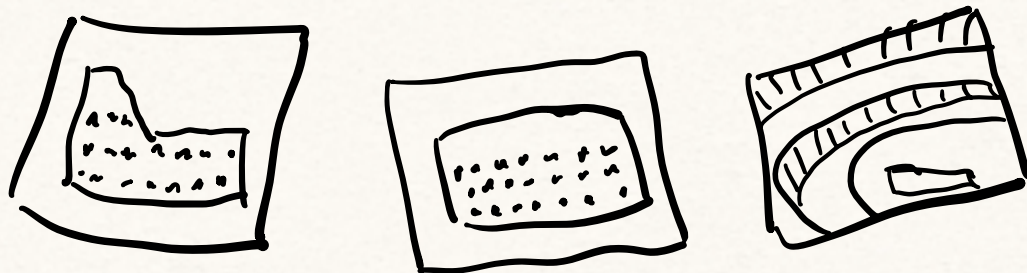


A Computer Vision Problem: Structure from Motion (SfM)



3D model

100.000 pictures



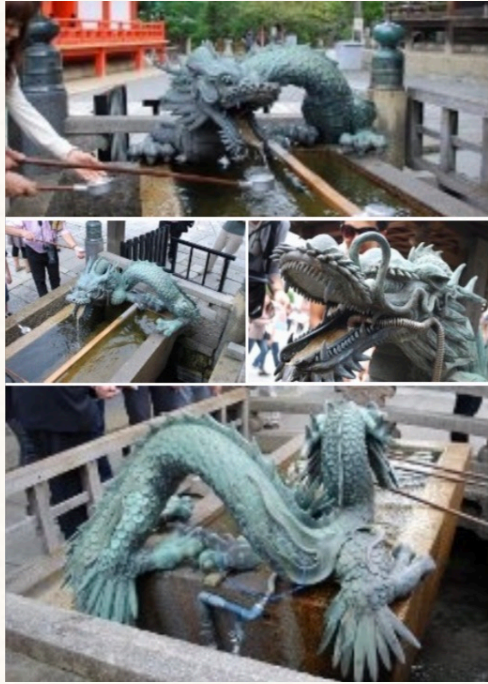
Difficult part:

We don't know the position / orientation of the cameras!

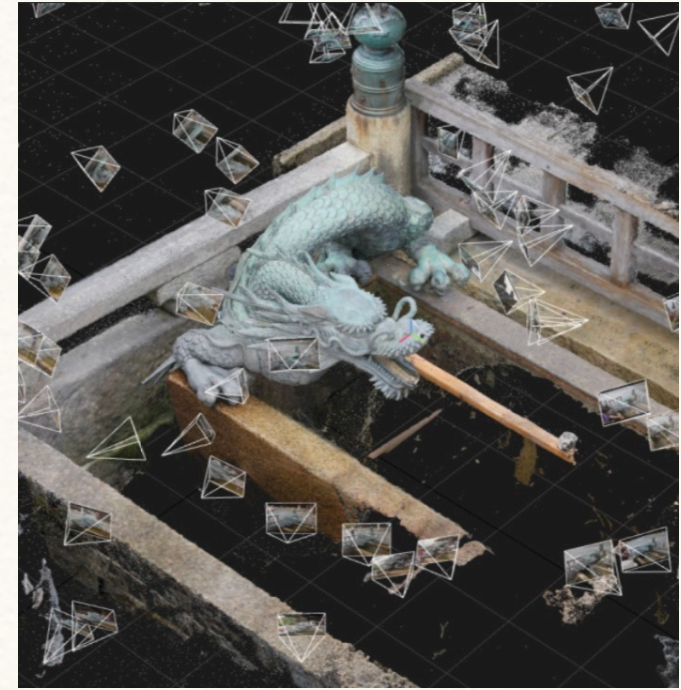
SfM pipeline

2

Given: pictures



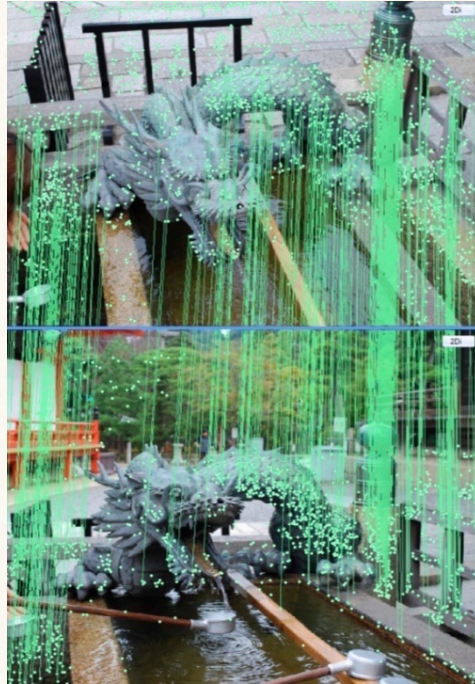
Output: 3D model
+ camera poses



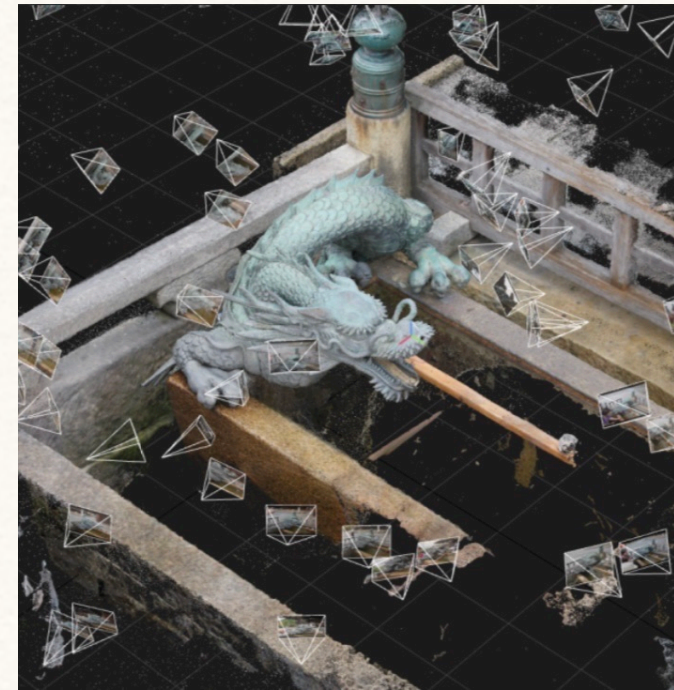
SfM pipeline

2

Given: pictures



Output: 3D model
+ camera poses



Step 1:

Identify corresponding
points (or lines, etc)

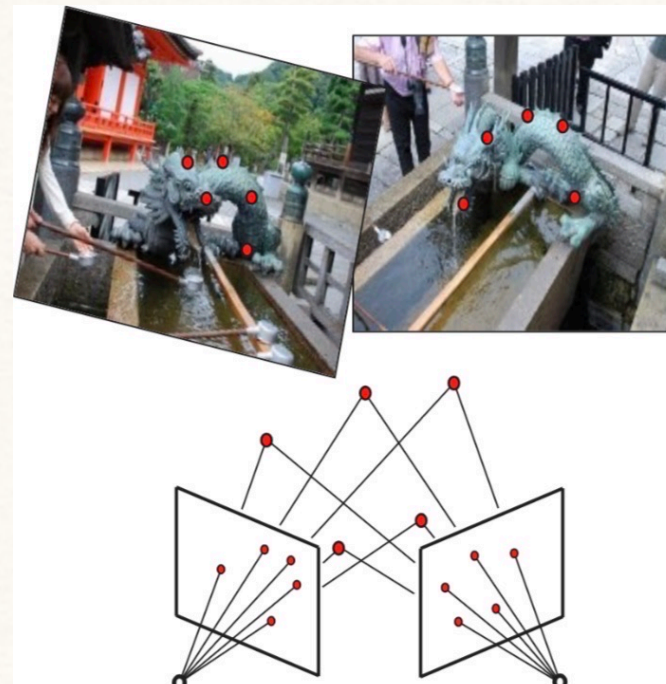
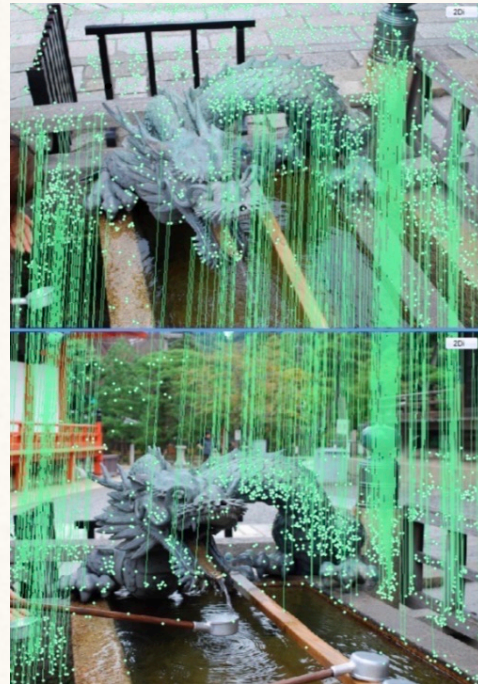
classical
algorithms

machine
learning

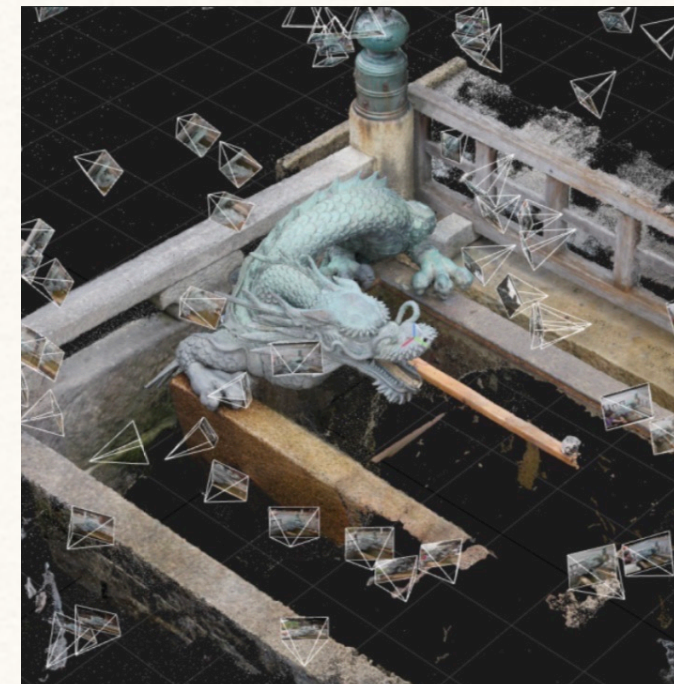
SfM pipeline

2

Given: pictures



Output: 3D model
+ camera poses



Step 1:

Identify corresponding
points (or lines, etc)

classical
algorithms

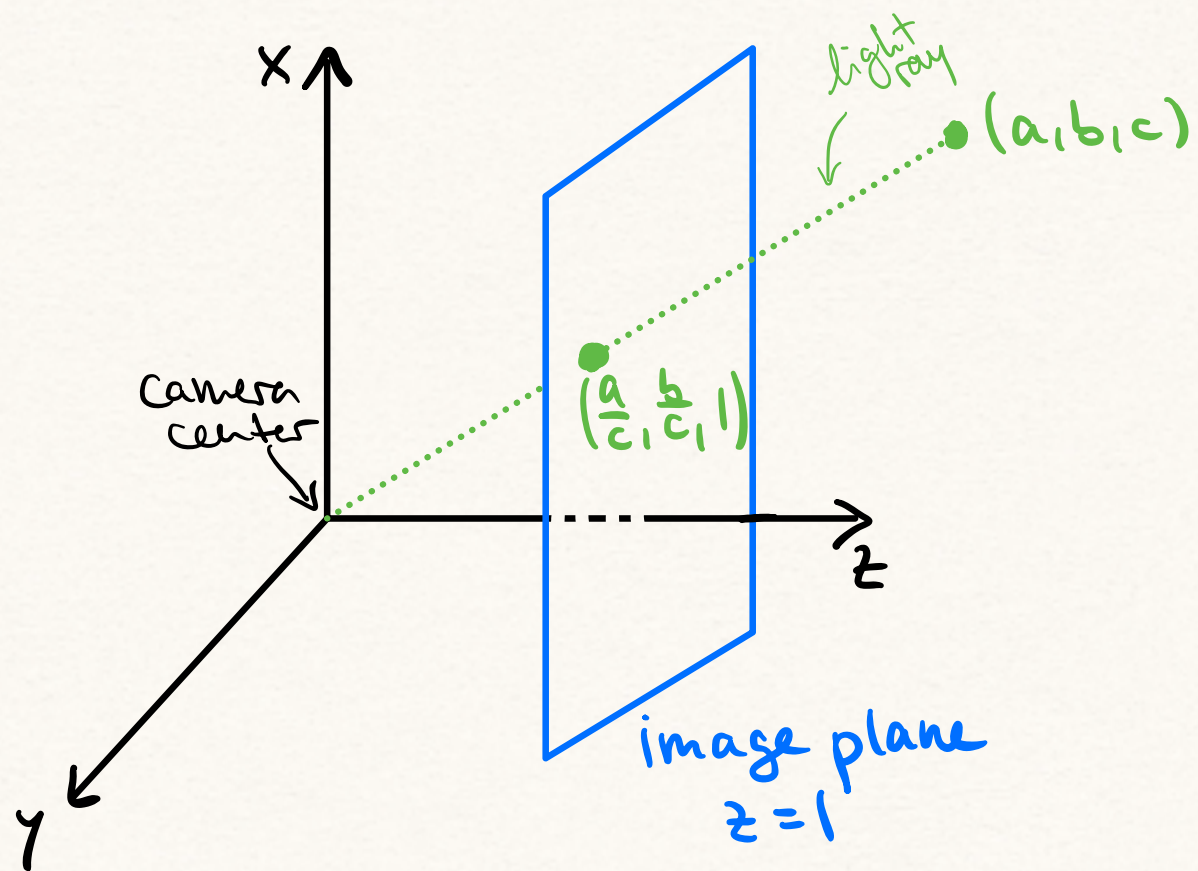
machine
learning

Step 2:

forget pictures,
solve for the 3D coordinates of the
points (lines, etc.) & the cameras

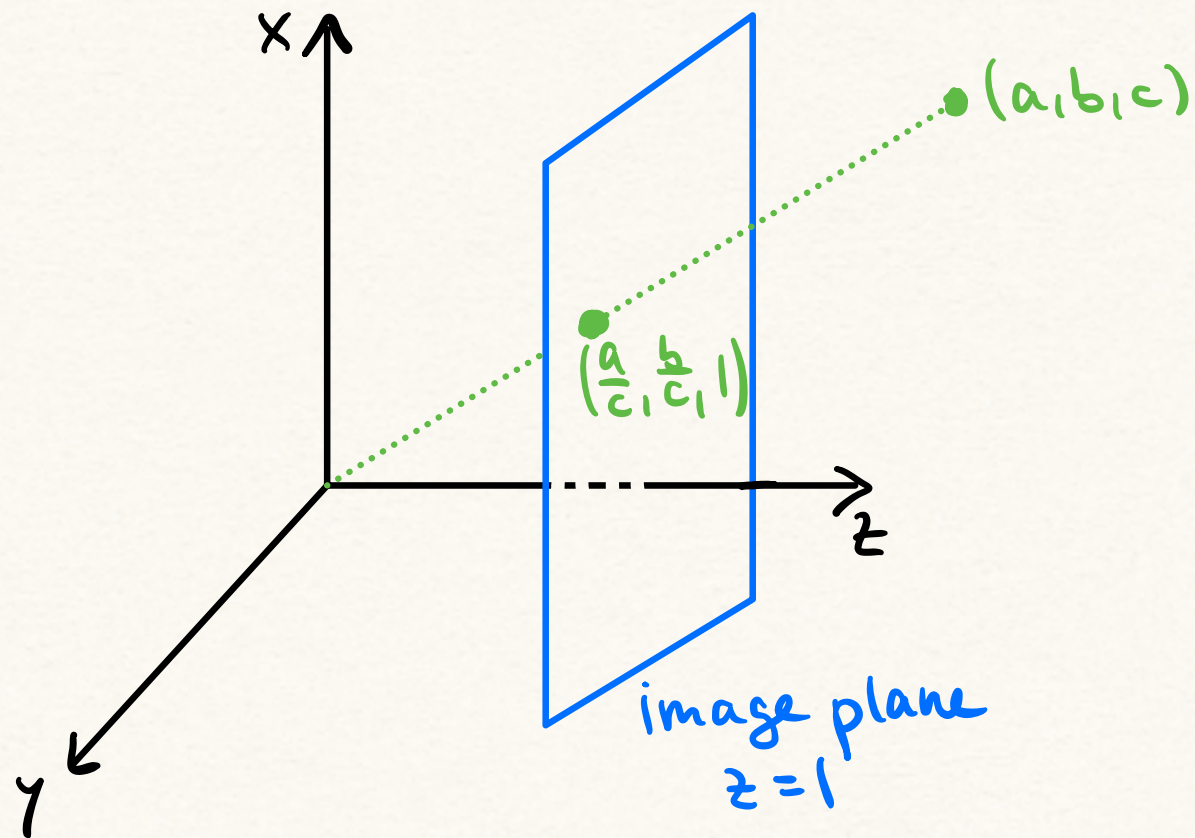
algebraic inverse problem!

What is a camera?



camera in
standard pose

What is a camera?

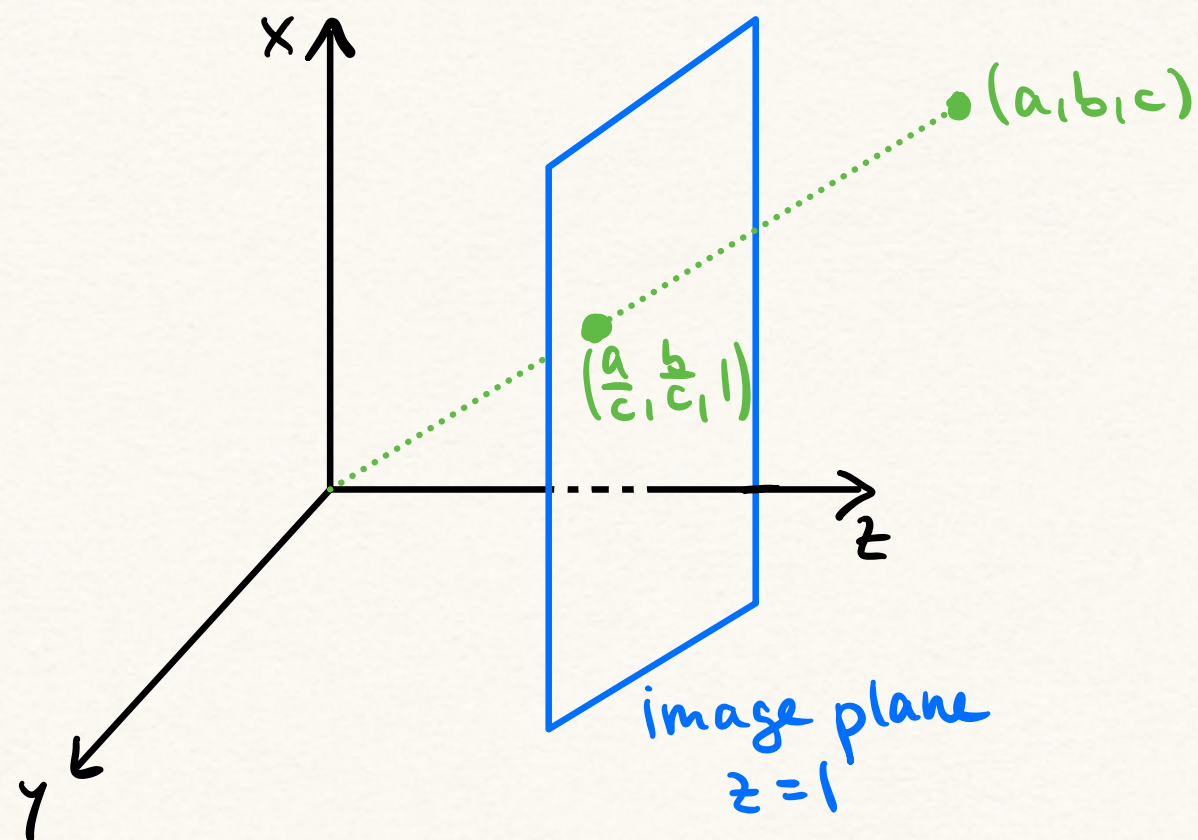


camera in
standard pose

Projectively: $[a:b:c:d]_{\mathbb{P}^3} \mapsto [a:b:c]_{\mathbb{P}^2}$

given by
the 3×4 matrix $[\mathbf{I}_3 \mid 0]$

What is a camera?



camera in
standard pose

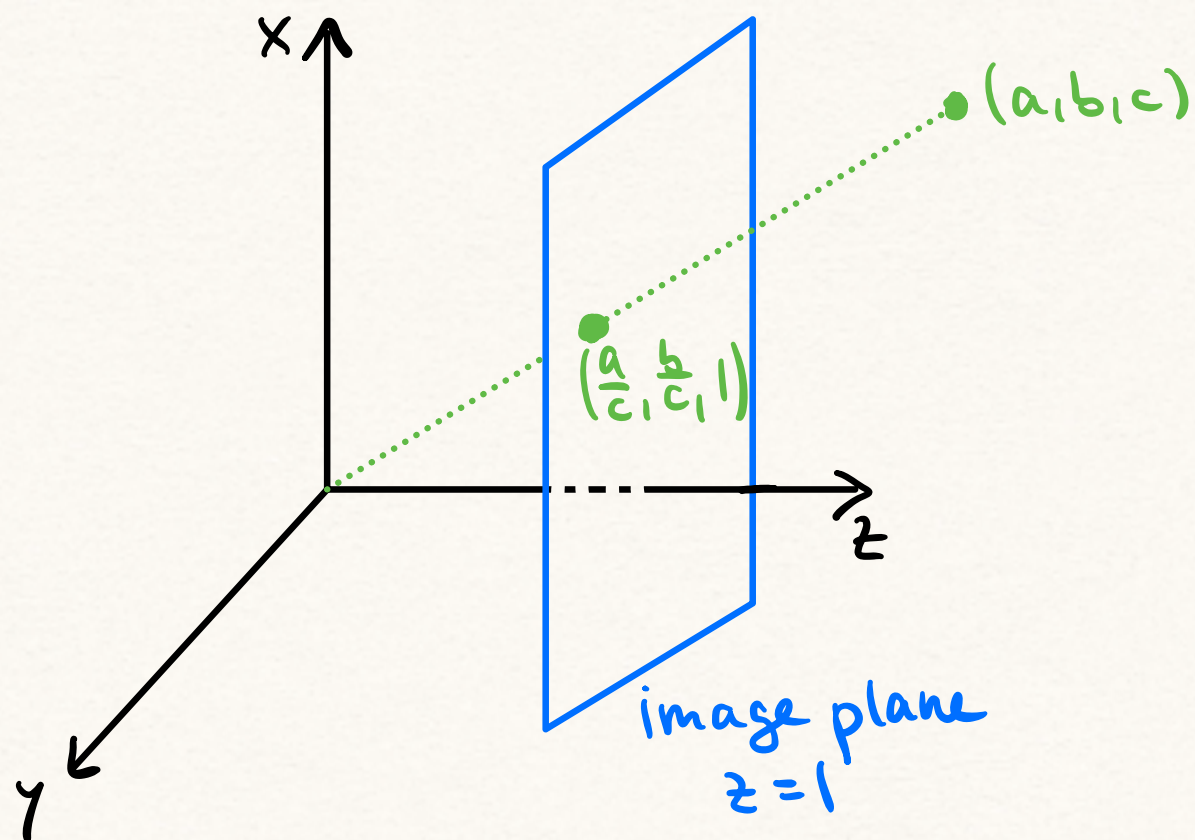
What is the base locus?

Computer vision
engineers actually
model their
problems in projective
space !!

Projectively: $[a:b:c:d] \mapsto [a:b:c]$
 $\mathbb{P}^3 \quad \dashrightarrow \mathbb{P}^2$

given by
the 3×4 matrix $[I_3 \mid 0]$

What is a camera?



camera in
standard pose

you cannot take
a picture of the
camera's center

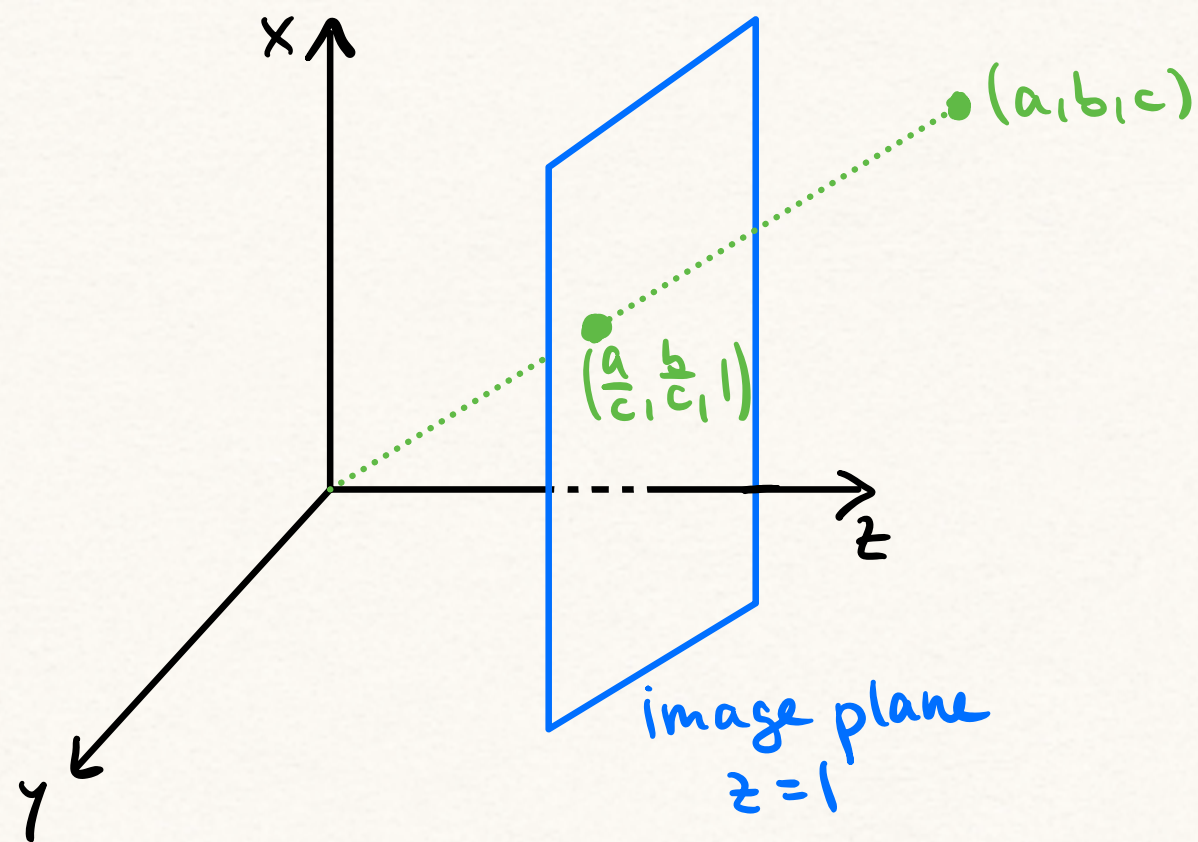
What is the base locus?
 $= \{[0:0:0:1]\}$

Projectively: $[a:b:c:d]_{\mathbb{P}^3} \mapsto [a:b:c]_{\mathbb{P}^2}$

given by
the 3×4 matrix $[I_3 \mid 0]$

Computer vision
engineers actually
model their
problems in projective
space !!

What is a camera?



camera in
standard pose

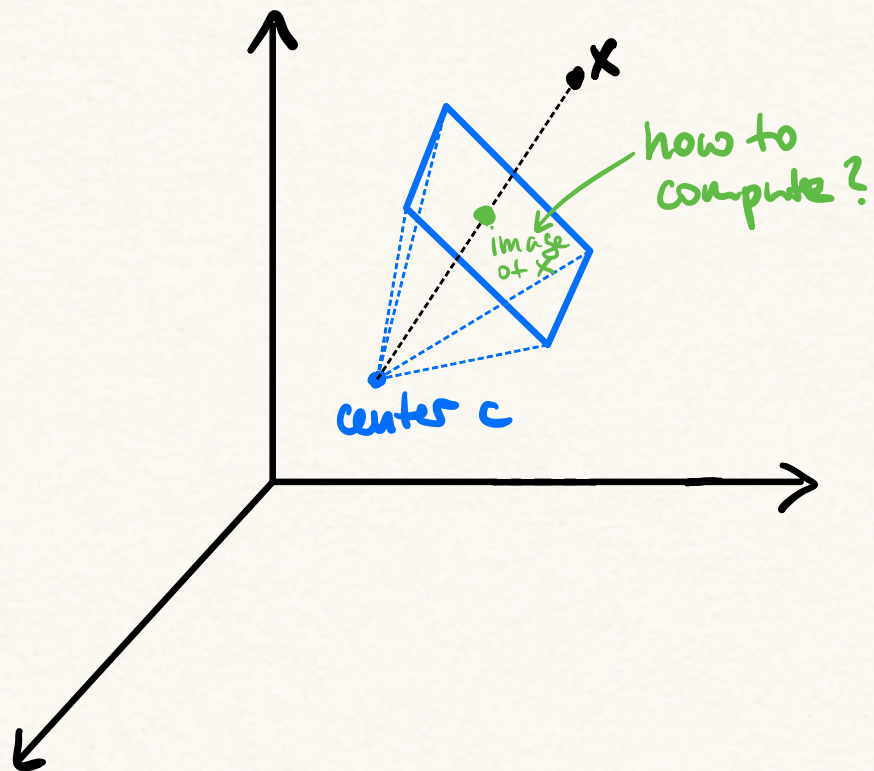
every camera is obtained
from the standard one
by **rotation** & **translation**

given by $[R \mid t]: \mathbb{P}^3 \dashrightarrow \mathbb{P}^2$

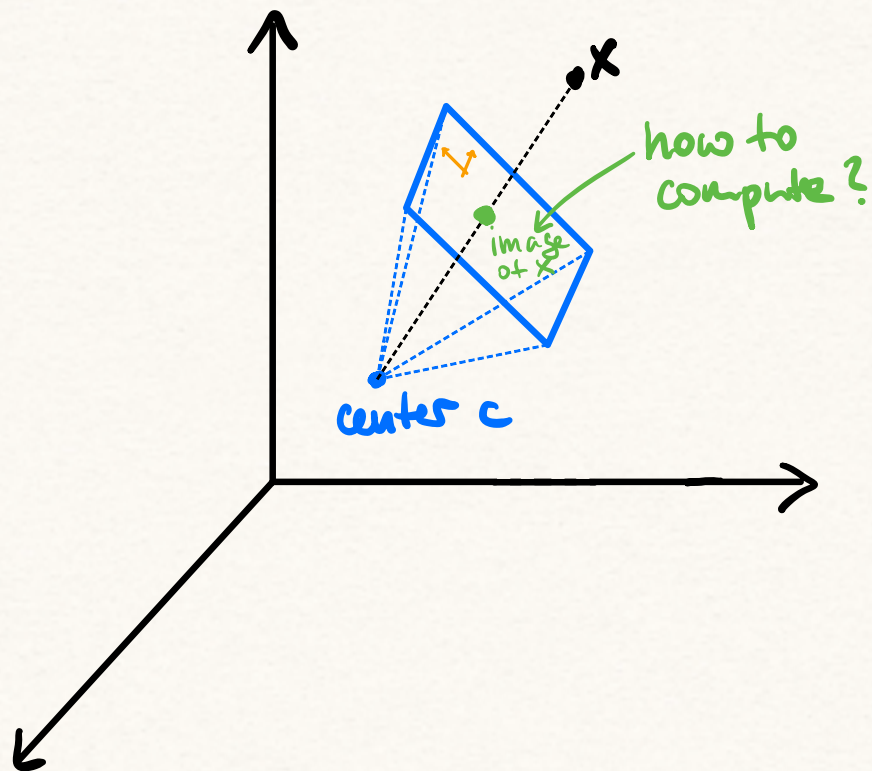
$R \in \text{SO}(3)$

$t \in \mathbb{R}^3$

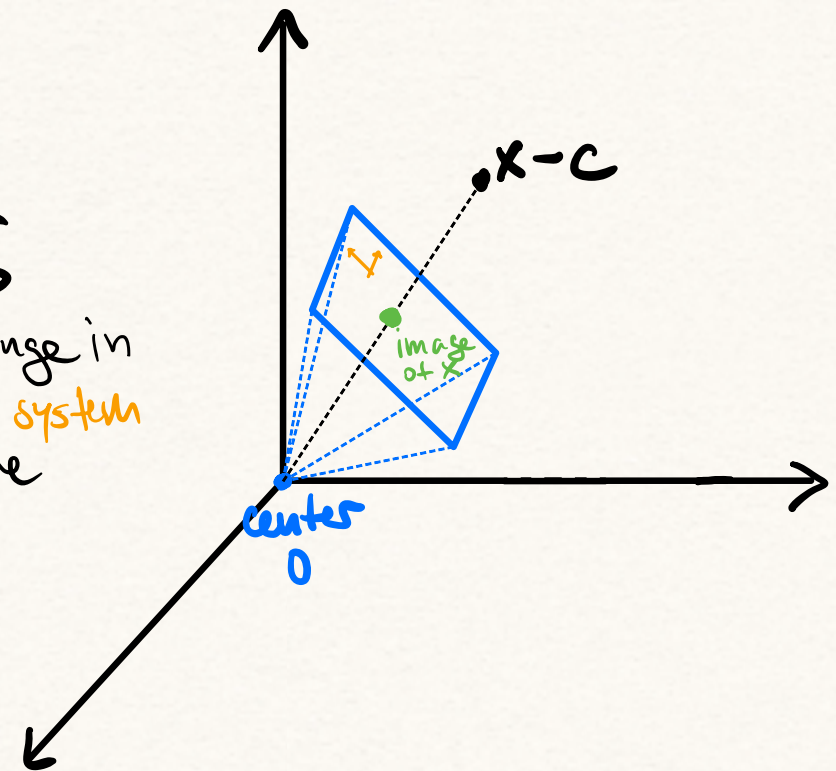
What is a camera?



What is a camera?

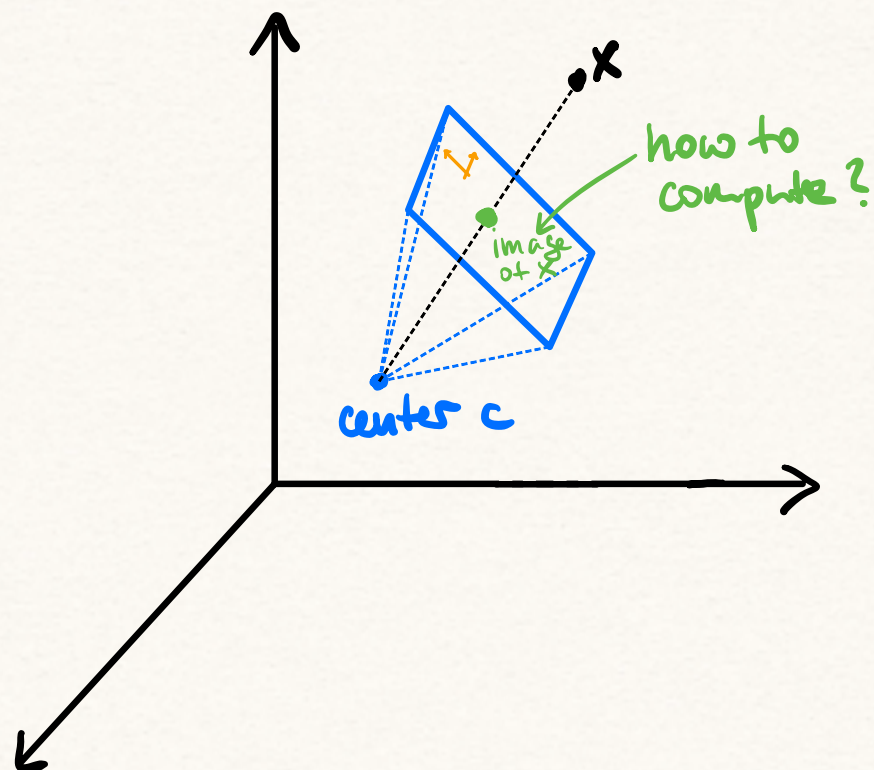


translate by $-c$
 ~~~~~  
 image( $x$ ) doesn't change in  
 the camera's coordinate system  
 of the image plane



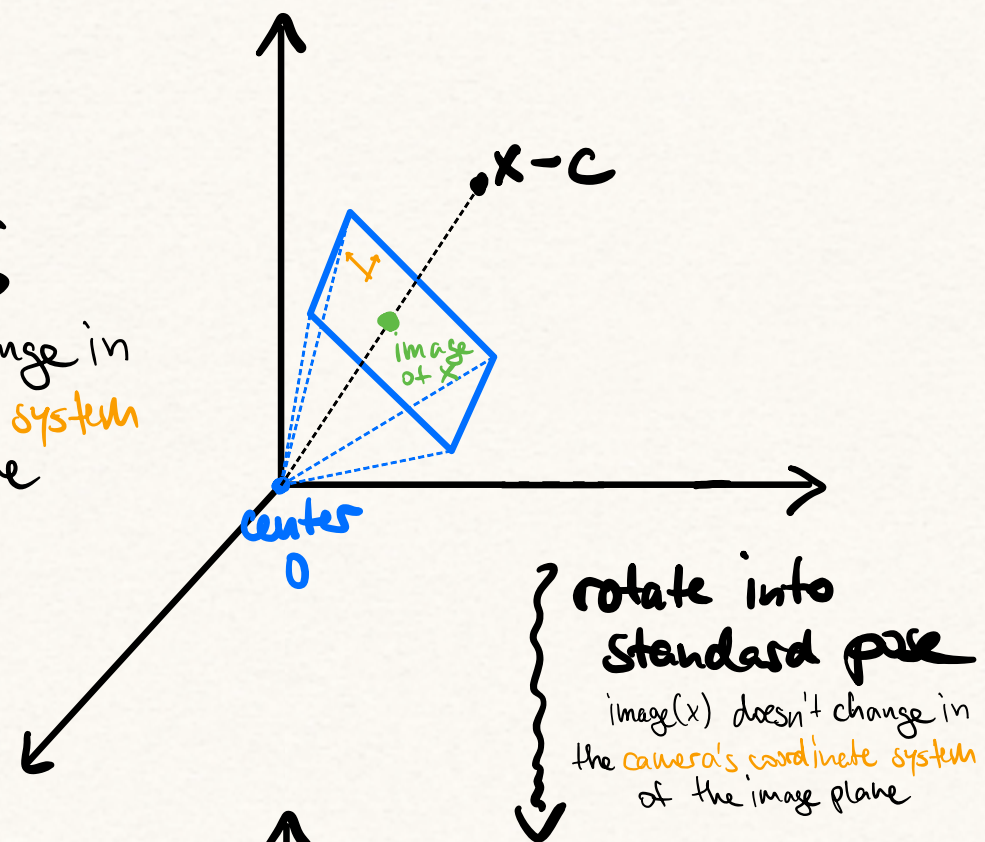


# What is a camera?



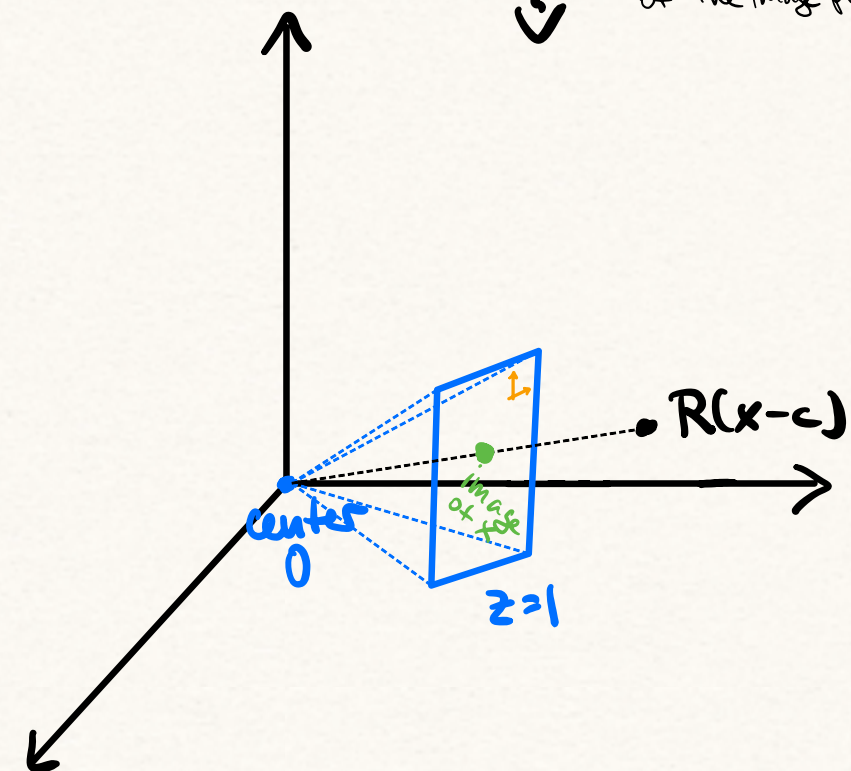
translate by  $-c$

$\text{image}(x)$  doesn't change in the camera's coordinate system of the image plane

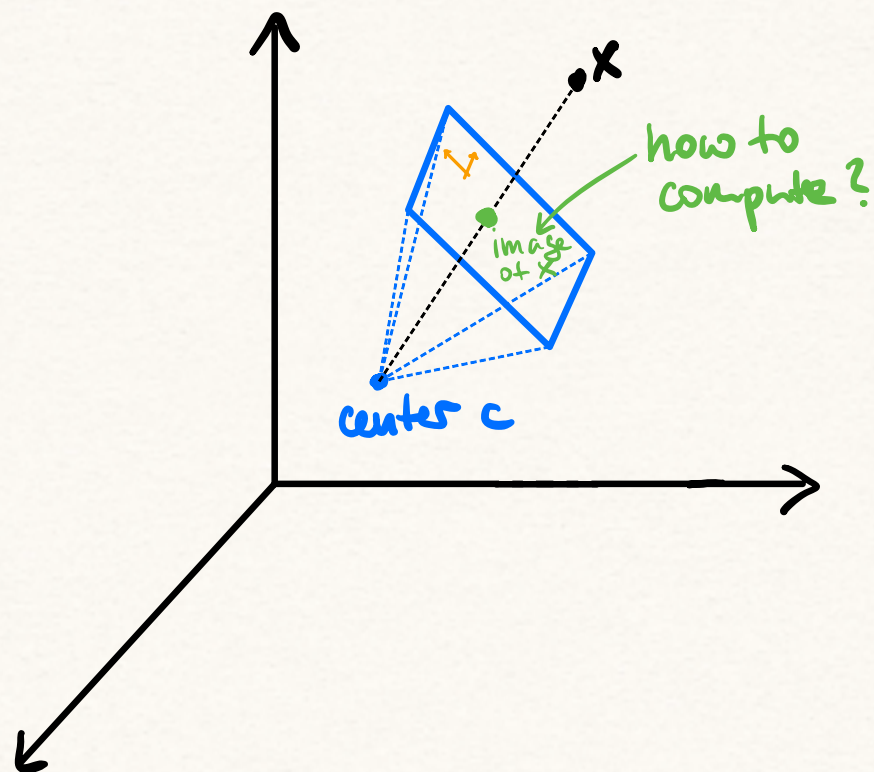


rotate into standard pose

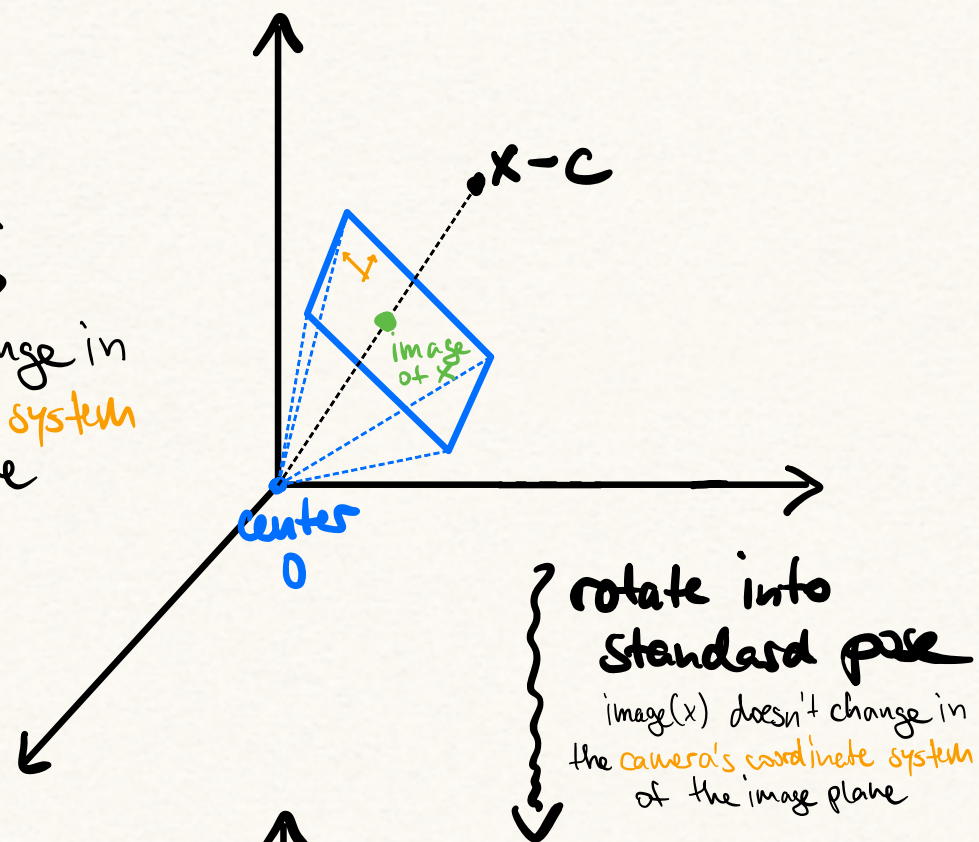
$\text{image}(x)$  doesn't change in the camera's coordinate system of the image plane



# What is a camera?



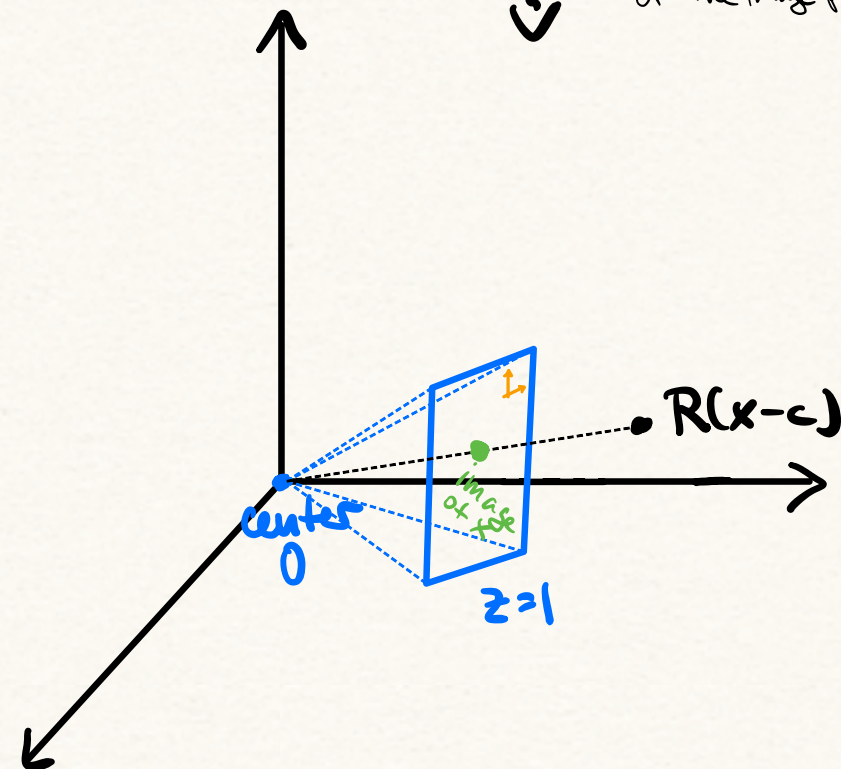
translate by  $-c$   
 $\rightarrow$   
 image( $x$ ) doesn't change in  
 the camera's coordinate system  
 of the image plane



rotate into  
 standard pose  
 $\downarrow$   
 image( $x$ ) doesn't change in  
 the camera's coordinate system  
 of the image plane

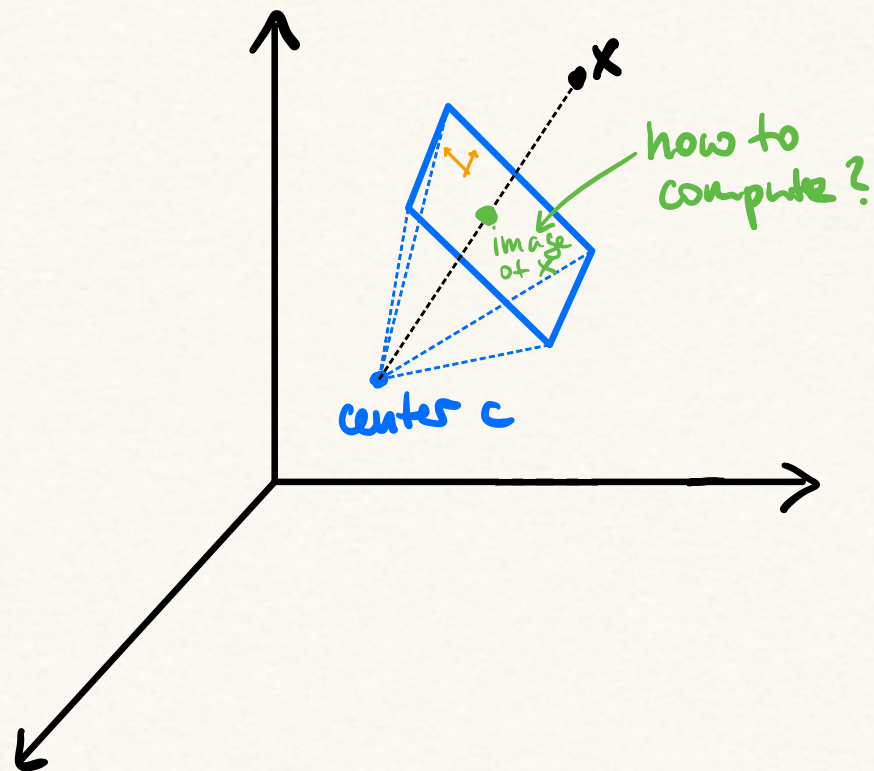
Hence: image of  $x \in \mathbb{R}^3$

$$= \begin{bmatrix} I_3 & | & 0 \end{bmatrix} \cdot \begin{bmatrix} R(x-c) \\ 1 \end{bmatrix}$$

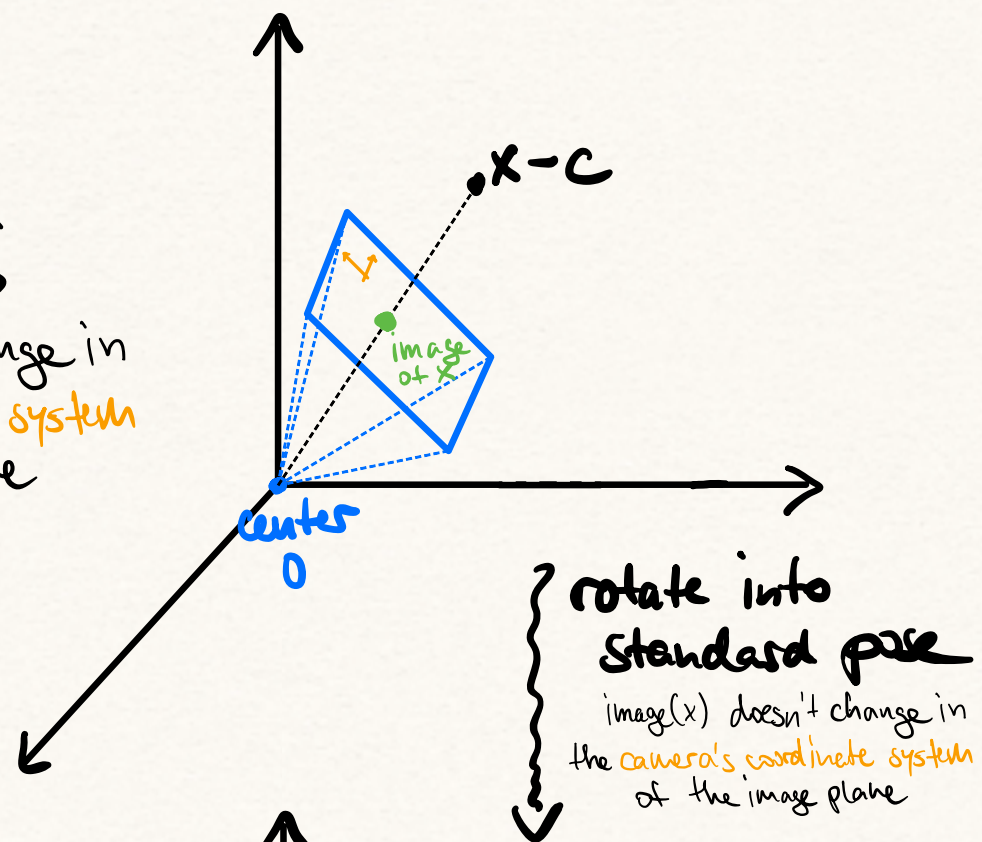




# What is a camera?



translate by  $-c$   
 $\rightarrow$   
 image( $x$ ) doesn't change in  
 the camera's coordinate system  
 of the image plane

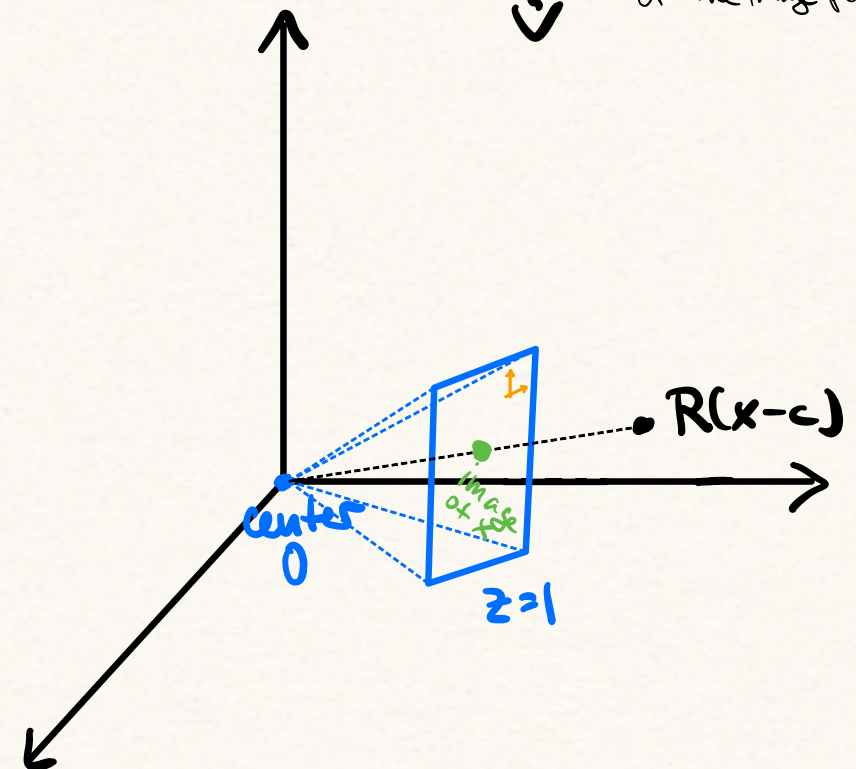


rotate into  
 standard pose  
 $\downarrow$   
 image( $x$ ) doesn't change in  
 the camera's coordinate system  
 of the image plane

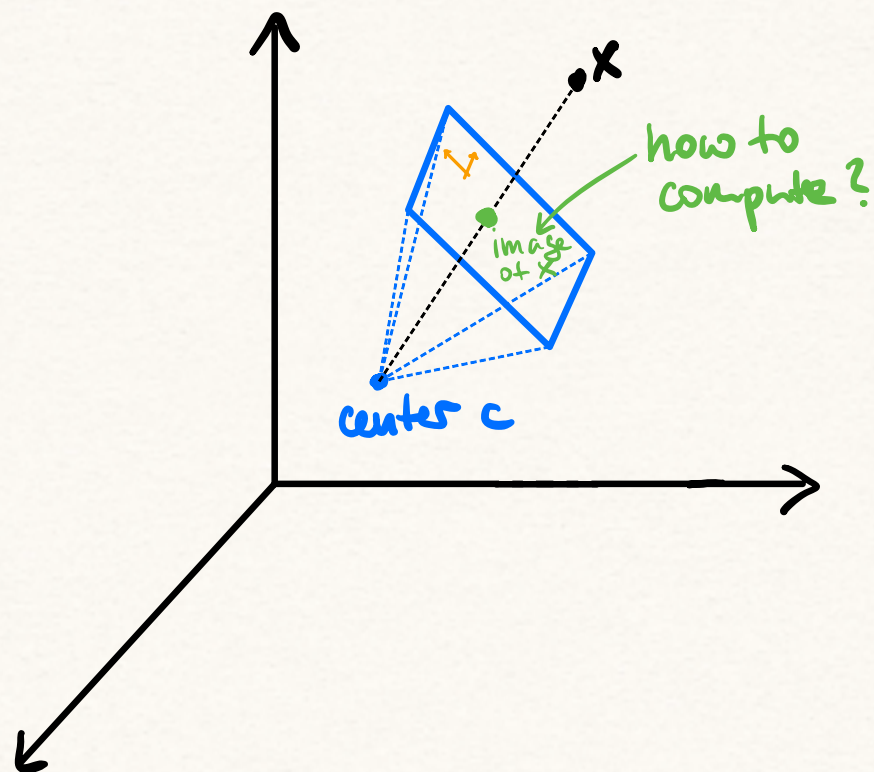
Hence: image of  $x \in \mathbb{R}^3$

$$= [I_3 | 0] \cdot \begin{bmatrix} R(x-c) \\ 1 \end{bmatrix}$$

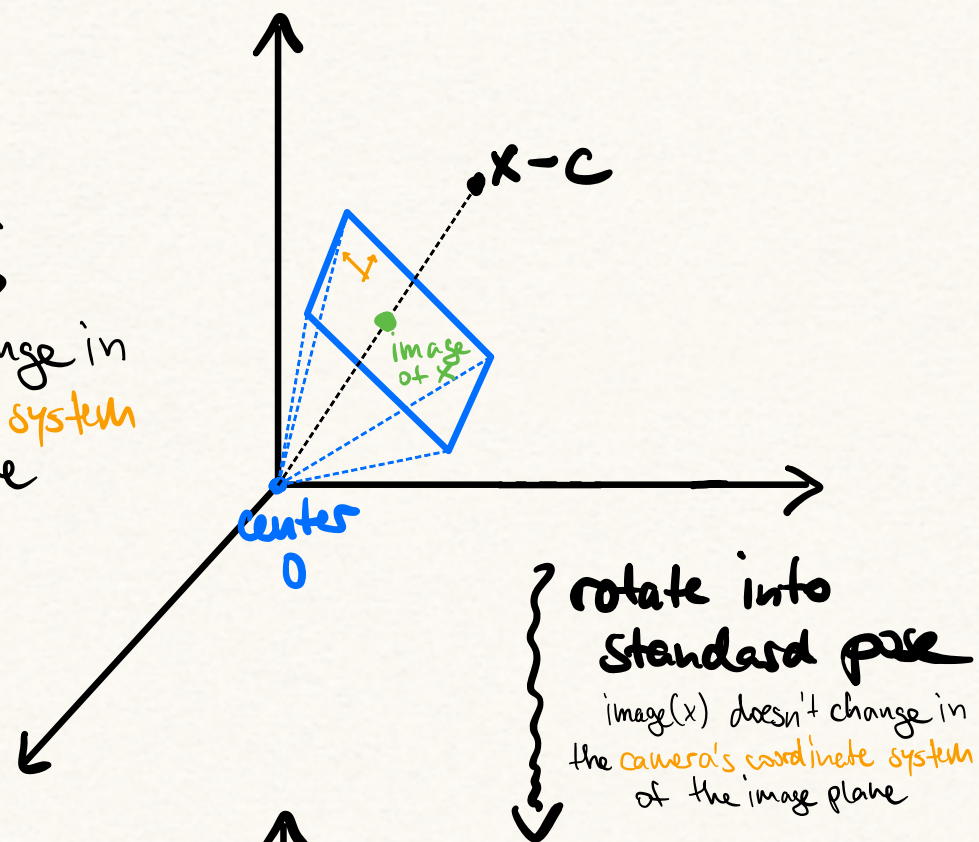
$$= Rx - Rc$$



# What is a camera?



translate by  $-c$   
 $\rightarrow$   
 image( $x$ ) doesn't change in  
 the camera's coordinate system  
 of the image plane



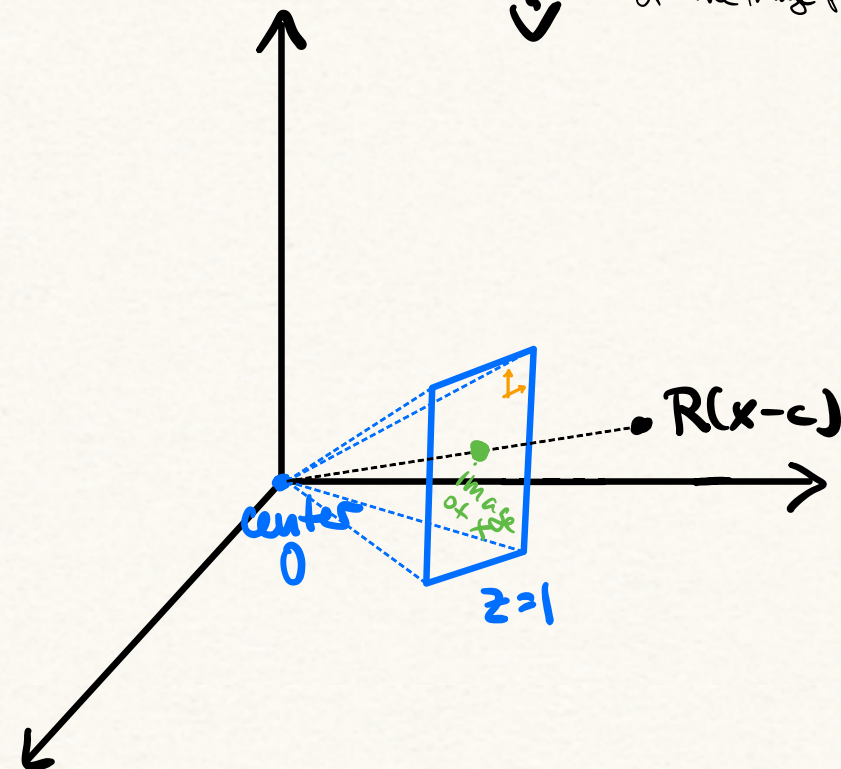
Hence: image of  $x \in \mathbb{R}^3$

$$= [I_3 | 0] \cdot \begin{bmatrix} R(x-c) \\ 1 \end{bmatrix}$$

$$= Rx - Rc$$

$$= [R | \underbrace{-Rc}] \begin{bmatrix} x \\ 1 \end{bmatrix}$$

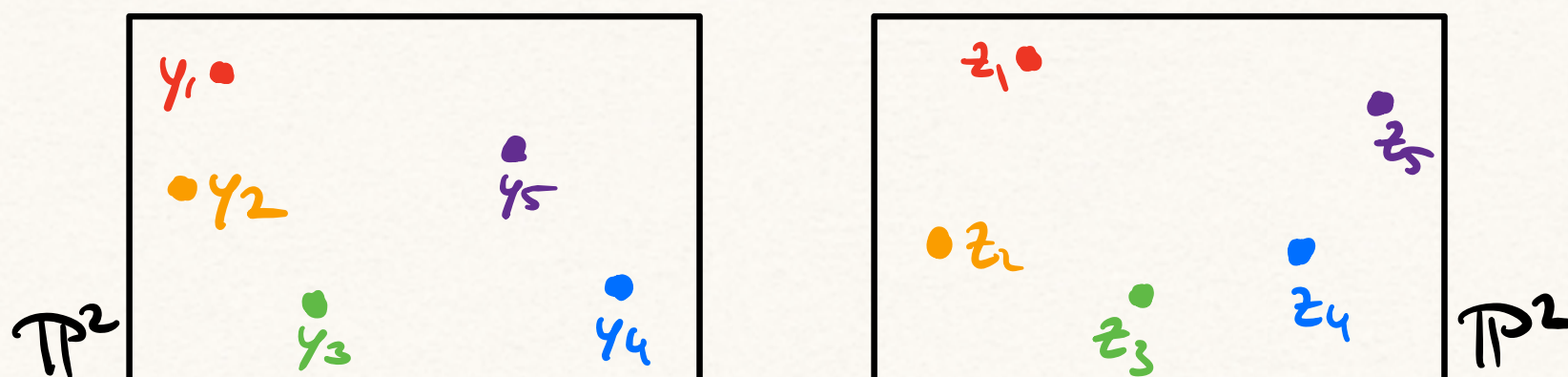
= + from previous slide





Example: 2 cameras took pictures of points in  $\mathbb{P}^3$ , yielding pairs of image points:

$$\begin{aligned} (y_1, z_1) &\in \mathbb{P}^2 \times \mathbb{P}^2 \\ (y_2, z_2) &\in \mathbb{P}^2 \times \mathbb{P}^2 \\ &\vdots \end{aligned}$$

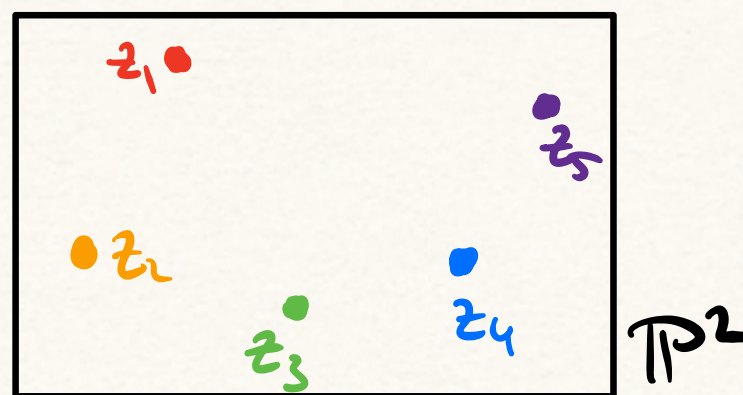


Goal: Find camera parameters  $R_1, R_2 \in \text{SO}(3)$ ,  $t_1, t_2 \in \mathbb{R}^3$ , and points  $x_1, x_2, \dots \in \mathbb{P}^3$  s.t.

$$[R_1 | t_1] x_i = y_i \quad [R_2 | t_2] x_i = z_i \quad (i=1, 2, \dots)$$

Example: 2 cameras took pictures of points in  $\mathbb{P}^3$ , yielding pairs of image points:

$$\begin{aligned} (y_1, z_1) &\in \mathbb{P}^2 \times \mathbb{P}^2 \\ (y_2, z_2) &\in \mathbb{P}^2 \times \mathbb{P}^2 \\ &\vdots \end{aligned}$$



Goal: Find camera parameters  $R_1, R_2 \in \text{SO}(3)$ ,  $t_1, t_2 \in \mathbb{R}^3$ , and points  $x_1, x_2, \dots \in \mathbb{P}^3$  s.t.

$$[R_1 | t_1] x_i = y_i \quad [R_2 | t_2] x_i = z_i \quad (i=1, 2, \dots)$$

polynomial equations!  $\smile$

$$\left( \begin{array}{l} \text{recall:} \\ \text{SO}(3) = \{ R \in \mathbb{R}^{3 \times 3} \mid \\ R^T R = I, \det(R) = 1 \} \end{array} \right)$$



Goal: Find camera parameters  $R_1, R_2 \in SO(3)$ ,  $t_1, t_2 \in \mathbb{R}^3$ ,  
and points  $x_1, x_2, \dots \in \mathbb{P}^3$  s.t.

$$[R_1 | t_1] x_i = y_i \quad [R_2 | t_2] x_i = z_i$$

Equivalently:

$$\text{For } \varphi: (SO(3) \times \mathbb{R}^3)^2 \times (\mathbb{P}^3)^k \rightarrow (\mathbb{P}^2)^k \times (\mathbb{P}^2)^k$$

$$([R_1 | t_1], [R_2 | t_2], x_1, \dots, x_k) \mapsto ([R_1 | t_1] x_1, \dots, [R_1 | t_1] x_k, [R_2 | t_2] x_1, \dots, [R_2 | t_2] x_k)$$

compute the fiber  $\varphi^{-1}(y_1, \dots, y_k, z_1, \dots, z_k)$ .

Goal: Find camera parameters  $R_1, R_2 \in SO(3)$ ,  $t_1, t_2 \in \mathbb{R}^3$ ,  
and points  $x_1, x_2, \dots \in \mathbb{P}^3$  s.t.

$$[R_1 | t_1] x_i = y_i \quad [R_2 | t_2] x_i = z_i$$

Equivalently:

$$\text{For } \varphi: (SO(3) \times \mathbb{R}^3)^2 \times (\mathbb{P}^3)^k \rightarrow (\mathbb{P}^2)^k \times (\mathbb{P}^2)^k$$

$$([R_1 | t_1], [R_2 | t_2], x_1, \dots, x_k) \mapsto ([R_1 | t_1] x_1, \dots, [R_1 | t_1] x_k, [R_2 | t_2] x_1, \dots, [R_2 | t_2] x_k)$$

compute the fiber  $\varphi^{-1}(y_1, \dots, y_k, z_1, \dots, z_k)$ .

Issue: The fibers are not finite



Goal: Find camera parameters  $R_1, R_2 \in SO(3)$ ,  $t_1, t_2 \in \mathbb{R}^3$ , and points  $x_1, x_2, \dots \in \mathbb{P}^3$  s.t.

$$[R_1 | t_1] x_i = y_i \quad [R_2 | t_2] x_i = z_i$$

Equivalently:

For  $\varphi: (SO(3) \times \mathbb{R}^3)^2 \times (\mathbb{P}^3)^k \rightarrow (\mathbb{P}^2)^k \times (\mathbb{P}^2)^k$  C<sub>1</sub> C<sub>2</sub>  
 $([R_1 | t_1], [R_2 | t_2], x_1, \dots, x_k) \mapsto ([R_1 | t_1] x_1, \dots, [R_1 | t_1] x_k, [R_2 | t_2] x_1, \dots, [R_2 | t_2] x_k)$   
 compute the fiber  $\varphi^{-1}(y_1, \dots, y_k, z_1, \dots, z_k)$ .

Issue: The fibers are not finite since the group

$$G := \left\{ \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \mid S \in SO(3), u \in \mathbb{R}^3, \lambda \in \mathbb{R}, \lambda \neq 0 \right\} \subseteq GL_4(\mathbb{R})$$

acts on them via  $\underset{G}{g} \cdot (C_1, C_2, x_1, \dots, x_k) := (C_1 g, C_2 g, \bar{g}^1 x_1, \dots, \bar{g}^1 x_k)$ .

8

$$\varphi: (SO(3) \times \mathbb{R}^3)^2 \times (\mathbb{P}^3)^k \rightarrow (\mathbb{P}^2)^k \times (\mathbb{P}^2)^k$$

$$([R_1 | t_1], [R_2 | t_2], x_1, \dots, x_k) \mapsto ([R_1 | t_1]x_1, \dots, [R_1 | t_1]x_k, [R_2 | t_2]x_1, \dots, [R_2 | t_2]x_k)$$

**Exercise:** a)  $G = \left\{ \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \mid \begin{array}{l} S \in SO(3), u \in \mathbb{R}^3, \\ \lambda \in \mathbb{R}, \lambda \neq 0 \end{array} \right\}$  is the largest subgroup of  $GL_4(\mathbb{R})$  acting on cameras  $C \in SO(3) \times \mathbb{R}^3$  via  $g \cdot C := Cg$ .



$$\varphi: (SO(3) \times \mathbb{R}^3)^2 \times (\mathbb{P}^3)^k \rightarrow (\mathbb{P}^3)^k \times (\mathbb{P}^2)^k$$

$$([R_1 | t_1], [R_2 | t_2], x_1, \dots, x_k) \mapsto ([R_1 | t_1]x_1, \dots, [R_1 | t_1]x_k, [R_2 | t_2]x_1, \dots, [R_2 | t_2]x_k)$$

**Exercise:** a)  $G = \left\{ \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \mid \begin{array}{l} S \in SO(3), u \in \mathbb{R}^3, \\ \lambda \in \mathbb{R}, \lambda \neq 0 \end{array} \right\}$  is the largest subgroup of  $GL_4(\mathbb{R})$  acting on cameras  $C \in SO(3) \times \mathbb{R}^3$  via  $g \cdot C := Cg$ .

$$\begin{array}{c} [R | t] \\ \uparrow \\ SO(3) \times \mathbb{R}^3 \end{array} \cdot \begin{array}{c} \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \\ \uparrow \\ G \end{array} =$$

$$\varphi: (SO(3) \times \mathbb{R}^3)^2 \times (\mathbb{P}^3)^k \rightarrow (\mathbb{P}^3)^k \times (\mathbb{P}^2)^k$$

$$([R_1 | t_1], [R_2 | t_2], x_1, \dots, x_k) \mapsto ([R_1 | t_1]x_1, \dots, [R_1 | t_1]x_k, [R_2 | t_2]x_1, \dots, [R_2 | t_2]x_k)$$

**Exercise:** a)  $G = \left\{ \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \mid \begin{array}{l} S \in SO(3), u \in \mathbb{R}^3, \\ \lambda \in \mathbb{R}, \lambda \neq 0 \end{array} \right\}$  is the largest subgroup of  $GL_4(\mathbb{R})$  acting on cameras  $C \in SO(3) \times \mathbb{R}^3$  via  $g \cdot C := Cg$ .

$$\underbrace{[R | t]}_{\substack{\uparrow \\ SO(3) \times \mathbb{R}^3}} \cdot \underbrace{\begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix}}_{\substack{\uparrow \\ G}} = \underbrace{[RS | Ru + \lambda t]}_{\text{this is again a camera}} \in SO(3) \times \mathbb{R}^3$$



$$\varphi: (SO(3) \times \mathbb{R}^3)^2 \times (\mathbb{P}^3)^k \rightarrow (\mathbb{P}^3)^k \times (\mathbb{P}^2)^k$$

$$([R_1 | t_1], [R_2 | t_2], x_1, \dots, x_k) \mapsto ([R_1 | t_1]x_1, \dots, [R_1 | t_1]x_k, [R_2 | t_2]x_1, \dots, [R_2 | t_2]x_k)$$

$\underbrace{\hspace{10em}}_{C_1} \quad \underbrace{\hspace{10em}}_{C_2}$

**Exercise:** a)  $G = \left\{ \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \mid \begin{array}{l} S \in SO(3), u \in \mathbb{R}^3, \\ \lambda \in \mathbb{R}, \lambda \neq 0 \end{array} \right\}$  is the largest subgroup of  $GL_4(\mathbb{R})$  acting on cameras  $C \in SO(3) \times \mathbb{R}^3$  via  $g \cdot C := Cg$ .

$$\underbrace{[R | t]}_{\substack{\uparrow \\ SO(3) \times \mathbb{R}^3}} \cdot \underbrace{\begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix}}_{\substack{\uparrow \\ G}} = \underbrace{[RS | Ru + \lambda t]}_{\text{this is again a camera}} \in SO(3) \times \mathbb{R}^3$$

but why is there no larger group that transforms cameras into cameras?

$$\varphi: (SO(3) \times \mathbb{R}^3)^2 \times (\mathbb{P}^3)^k \rightarrow (\mathbb{P}^3)^k \times (\mathbb{P}^2)^k$$

$$([R_1 | t_1], [R_2 | t_2], x_1, \dots, x_k) \mapsto ([R_1 | t_1]x_1, \dots, [R_1 | t_1]x_k, [R_2 | t_2]x_1, \dots, [R_2 | t_2]x_k)$$

$\underbrace{\hspace{10em}}_{C_1} \quad \underbrace{\hspace{10em}}_{C_2}$

**Exercise:** a)  $G = \left\{ \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \mid \begin{array}{l} S \in SO(3), u \in \mathbb{R}^3, \\ \lambda \in \mathbb{R}, \lambda \neq 0 \end{array} \right\}$  is the largest subgroup of  $GL_4(\mathbb{R})$  acting on cameras  $C \in SO(3) \times \mathbb{R}^3$  via  $g \cdot C := Cg$ .

$$\underbrace{[R | t]}_{\substack{\uparrow \\ SO(3) \times \mathbb{R}^3}} \cdot \underbrace{\begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix}}_{\substack{\uparrow \\ G}} = \underbrace{[RS | Ru + \lambda t]}_{\text{this is again a camera}} \in SO(3) \times \mathbb{R}^3$$

but why is there no larger group that transforms cameras into cameras?

$$b) (C_1, C_2, x) \in \vec{\varphi}'(y, z), g \in G \Rightarrow g \cdot (C_1, C_2, x) \in \vec{\varphi}'(y, z)$$

We can only hope to reconstruct cameras & 3D points up to global rotation, global translation & global scaling.



$$G = \left\{ \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \mid \begin{array}{l} S \in SO(3), u \in \mathbb{R}^3, \\ \lambda \in \mathbb{R}, \lambda \neq 0 \end{array} \right\}, \quad \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix}^{-1} = \begin{bmatrix} S^{-1} & -\lambda^{-1} S^{-1} u \\ 0 & \lambda^{-1} \end{bmatrix} \quad 9$$

$$\begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \text{ acts on cameras } [R|t] \text{ via } [R|t] \cdot \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} = [RS | Ru + \lambda t]$$

$$\text{and on points } \mathbb{P}^3 \ni x = \begin{bmatrix} x^{\text{aff}} \\ \overline{x^\infty} \end{bmatrix} \text{ via } \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix}^{-1} \cdot x = \begin{bmatrix} S^{-1} x^{\text{aff}} - x^\infty \lambda^{-1} u \\ \lambda^{-1} x^\infty \end{bmatrix}$$

Let  $x$  be a finite point, i.e., we may assume  $x^\infty = 1$ .

global rotation:  $\begin{bmatrix} S & 0 \\ 0 & 1 \end{bmatrix} \cdot ([R|t], x) = ([RS | t], \begin{bmatrix} S^{-1} x^{\text{aff}} \\ 1 \end{bmatrix})$

$$G = \left\{ \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \mid \begin{array}{l} S \in SO(3), u \in \mathbb{R}^3, \\ \lambda \in \mathbb{R}, \lambda \neq 0 \end{array} \right\}, \quad \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix}^{-1} = \begin{bmatrix} S^{-1} & -\lambda^{-1} S^{-1} u \\ 0 & \lambda^{-1} \end{bmatrix} \quad 9$$

$$\begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \text{ acts on cameras } [R|t] \text{ via } [R|t] \cdot \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} = [RS | Ru + \lambda t]$$

$$\text{and on points } \mathbb{P}^3 \ni x = \begin{bmatrix} x^{\text{aff}} \\ \overline{x^\infty} \end{bmatrix} \text{ via } \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix}^{-1} \cdot x = \begin{bmatrix} S^{-1} x^{\text{aff}} - x^\infty \lambda^{-1} u \\ \lambda^{-1} x^\infty \end{bmatrix}$$

Let  $x$  be a finite point, i.e., we may assume  $x^\infty = 1$ .

$$\text{global rotation: } \begin{bmatrix} S & 0 \\ 0 & 1 \end{bmatrix} \cdot ([R|t], x) = ([RS | t], \begin{bmatrix} S^{-1} x^{\text{aff}} \\ 1 \end{bmatrix})$$

$$\text{global translation: } \begin{bmatrix} I & u \\ 0 & 1 \end{bmatrix} \cdot ([R|t], x) = ([R | Ru + t], \begin{bmatrix} x^{\text{aff}} - u \\ 1 \end{bmatrix})$$



$$G = \left\{ \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \mid \begin{array}{l} S \in SO(3), u \in \mathbb{R}^3, \\ \lambda \in \mathbb{R}, \lambda \neq 0 \end{array} \right\}, \quad \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix}^{-1} = \begin{bmatrix} S^{-1} & -\lambda^{-1} S^{-1} u \\ 0 & \lambda^{-1} \end{bmatrix} \quad 9$$

$$\begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \text{ acts on cameras } [R|t] \text{ via } [R|t] \cdot \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} = [RS | Ru + \lambda t]$$

$$\text{and on points } \mathbb{P}^3 \ni x = \begin{bmatrix} x^{\text{aff}} \\ \overline{x^\infty} \end{bmatrix} \text{ via } \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix}^{-1} \cdot x = \begin{bmatrix} S^{-1} x^{\text{aff}} - x^\infty \lambda^{-1} u \\ \lambda^{-1} x^\infty \end{bmatrix}$$

Let  $x$  be a finite point, i.e., we may assume  $x^\infty = 1$ .

$$\text{global rotation: } \begin{bmatrix} S & 0 \\ 0 & 1 \end{bmatrix} \cdot ([R|t], x) = ([RS | t], \begin{bmatrix} S^{-1} x^{\text{aff}} \\ 1 \end{bmatrix})$$

$$\text{global translation: } \begin{bmatrix} I & u \\ 0 & 1 \end{bmatrix} \cdot \underbrace{([R|t], x)}_{\substack{\text{has center (=kernel)} \\ \begin{bmatrix} -R^{-1} t \\ 1 \end{bmatrix}}} = \underbrace{([R | Ru + t], \begin{bmatrix} x^{\text{aff}} - u \\ 1 \end{bmatrix})}_{\substack{\text{has center} \\ \begin{bmatrix} -R^{-1} t - u \\ 1 \end{bmatrix}}}$$

$$G = \left\{ \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \mid \begin{array}{l} S \in SO(3), u \in \mathbb{R}^3, \\ \lambda \in \mathbb{R}, \lambda \neq 0 \end{array} \right\}, \quad \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix}^{-1} = \begin{bmatrix} S^{-1} & -\lambda^{-1} S^{-1} u \\ 0 & \lambda^{-1} \end{bmatrix} \quad 9$$

$$\begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \text{ acts on cameras } [R|t] \text{ via } [R|t] \cdot \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} = [RS | Ru + \lambda t]$$

$$\text{and on points } \mathbb{P}^3 \ni x = \begin{bmatrix} x^{\text{aff}} \\ \overline{x^\infty} \end{bmatrix} \text{ via } \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix}^{-1} \cdot x = \begin{bmatrix} S^{-1} x^{\text{aff}} - x^\infty \lambda^{-1} u \\ \lambda^{-1} x^\infty \end{bmatrix}$$

Let  $x$  be a finite point, i.e., we may assume  $x^\infty = 1$ .

$$\text{global rotation: } \begin{bmatrix} S & 0 \\ 0 & 1 \end{bmatrix} \cdot ([R|t], x) = ([RS | t], \begin{bmatrix} S^{-1} x^{\text{aff}} \\ 1 \end{bmatrix})$$

$$\text{global translation: } \begin{bmatrix} I & u \\ 0 & 1 \end{bmatrix} \cdot \underbrace{([R|t], x)}_{\substack{\text{has center (=kernel)} \\ [-R^{-1} | t]}} = \underbrace{([R | Ru + t])}_{\substack{\text{has center} \\ [-R^{-1} | t - u]}} \cdot \begin{bmatrix} x^{\text{aff}} - u \\ 1 \end{bmatrix}$$

simultaneous translation by  $-u$



$$G = \left\{ \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \mid \begin{array}{l} S \in SO(3), u \in \mathbb{R}^3, \\ \lambda \in \mathbb{R}, \lambda \neq 0 \end{array} \right\}, \quad \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix}^{-1} = \begin{bmatrix} S^{-1} & -\lambda^{-1} S^{-1} u \\ 0 & \lambda^{-1} \end{bmatrix} \quad 10$$

$$\begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \text{ acts on cameras } [R|t] \text{ via } [R|t] \cdot \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} = [RS \mid Ru + \lambda t]$$

$$\text{and on points } \mathbb{P}^3 \ni x = \begin{bmatrix} x^{\text{aff}} \\ \overline{x^\infty} \end{bmatrix} \text{ via } \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix}^{-1} \cdot x = \begin{bmatrix} S^{-1} x^{\text{aff}} - x^\infty \lambda^{-1} u \\ \lambda^{-1} x^\infty \end{bmatrix}$$

Let  $x$  be a finite point, i.e., we may assume  $x^\infty = 1$ .

global scaling:  $\begin{bmatrix} I & 0 \\ 0 & \lambda \end{bmatrix} \cdot ([R|t], x) = ([R \mid \lambda t], \begin{bmatrix} x^{\text{aff}} \\ \lambda^{-1} \end{bmatrix})$

$$G = \left\{ \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \mid \begin{array}{l} S \in SO(3), u \in \mathbb{R}^3, \\ \lambda \in \mathbb{R}, \lambda \neq 0 \end{array} \right\}, \quad \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix}^{-1} = \begin{bmatrix} S^{-1} & -\lambda^{-1} S^{-1} u \\ 0 & \lambda^{-1} \end{bmatrix} \quad 10$$

$$\begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \text{ acts on cameras } [R|t] \text{ via } [R|t] \cdot \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} = [RS \mid Ru + \lambda t]$$

$$\text{and on points } \mathbb{P}^3 \ni x = \begin{bmatrix} x^{\text{aff}} \\ \overline{x^\infty} \end{bmatrix} \text{ via } \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix}^{-1} \cdot x = \begin{bmatrix} S^{-1} x^{\text{aff}} - x^\infty \lambda^{-1} u \\ \lambda^{-1} x^\infty \end{bmatrix}$$

Let  $x$  be a finite point, i.e., we may assume  $x^\infty = 1$ .

global scaling:  $\begin{bmatrix} I & 0 \\ 0 & \lambda \end{bmatrix} \cdot (\underbrace{[R|t]}_{\substack{\text{has center (=kernel)} \\ [-R^{-1}t]}}, x) = (\underbrace{[R|\lambda t]}_{\substack{\text{has center} \\ [-\lambda R^{-1}t]}}, \underbrace{\begin{bmatrix} x^{\text{aff}} \\ \lambda^{-1} \end{bmatrix}}_{\substack{= \\ \begin{bmatrix} \lambda x^{\text{aff}} \\ 1 \end{bmatrix}}})$



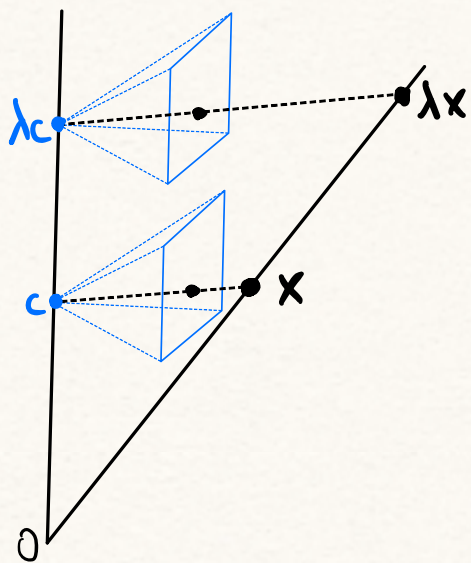
$$G = \left\{ \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \mid \begin{array}{l} S \in SO(3), u \in \mathbb{R}^3, \\ \lambda \in \mathbb{R}, \lambda \neq 0 \end{array} \right\}, \quad \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix}^{-1} = \begin{bmatrix} S^{-1} & -\lambda^{-1} S^{-1} u \\ 0 & \lambda^{-1} \end{bmatrix} \quad 10$$

$$\begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} \text{ acts on cameras } [R|t] \text{ via } [R|t] \cdot \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix} = [RS | Ru + \lambda t]$$

$$\text{and on points } \mathbb{P}^3 \ni x = \begin{bmatrix} x^{\text{aff}} \\ \overline{x^\infty} \end{bmatrix} \text{ via } \begin{bmatrix} S & u \\ 0 & \lambda \end{bmatrix}^{-1} \cdot x = \begin{bmatrix} S^{-1} x^{\text{aff}} - x^\infty \lambda^{-1} u \\ \lambda^{-1} x^\infty \end{bmatrix}$$

Let  $x$  be a finite point, i.e., we may assume  $x^\infty = 1$ .

global scaling:  $\begin{bmatrix} I & 0 \\ 0 & \lambda \end{bmatrix} \cdot (\underbrace{[R|t]}_{\text{has center } (= \text{kernel})} \underbrace{, x}_{\text{has center } \underbrace{\begin{bmatrix} x^{\text{aff}} \\ \lambda^{-1} \end{bmatrix}}_{\text{" } \begin{bmatrix} \lambda x^{\text{aff}} \\ 1 \end{bmatrix}}}) = (\underbrace{[R|\lambda t]}_{\text{has center } \underbrace{\begin{bmatrix} -\lambda R^{-1} t \\ 1 \end{bmatrix}}_{\text{simultaneous scaling by } \lambda}}, \underbrace{\begin{bmatrix} x^{\text{aff}} \\ \lambda^{-1} \end{bmatrix}}_{\text{" } \begin{bmatrix} \lambda x^{\text{aff}} \\ 1 \end{bmatrix}})$



11

We wish to "mod out" the group  $G$  out of the fibers

**Exercise:** There is a Zariski closed proper subset  $\Delta \subsetneq (SO(3) \times \mathbb{R}^3)^2$  such that all camera pairs  $(C_1, C_2) \notin \Delta$  can be written as

$$(C_1, C_2) = ([I | 0], [R | \check{c}]) \cdot g$$

for some  $R \in SO(3)$ ,  $v \in \mathbb{R}$ ,  $g \in G$ .



We wish to "mod out" the group  $G$  out of the fibers

**Exercise:** There is a Zariski closed proper subset  $\Delta \subsetneq (SO(3) \times \mathbb{R}^3)^2$  such that all camera pairs  $(C_1, C_2) \notin \Delta$  can be written as

$$(C_1, C_2) = ([I | 0], [R | \check{c}]) \cdot g$$

for some  $R \in SO(3)$ ,  $v, w \in \mathbb{R}$ ,  $g \in G$ .

As this works for almost all camera pairs, we focus on those now & deal with  $\Delta$  later:

Modify  $\varphi: (SO(3) \times \mathbb{R}^3)^2 \times (\mathbb{P}^3)^k \rightarrow (\mathbb{P}^2)^k \times (\mathbb{P}^2)^k$

$$(C_1, C_2, x_1, \dots, x_k) \mapsto (C_1 x_1, \dots, C_1 x_k, C_2 x_1, \dots, C_2 x_k)$$

to  $\tilde{\varphi}: SO(3) \times \mathbb{R}^2 \times (\mathbb{P}^3)^k \rightarrow (\mathbb{P}^2)^k \times (\mathbb{P}^2)^k$

$$(R, v, w, x_1, \dots, x_k) \mapsto \varphi([I | 0], [R | \check{c}], x_1, \dots, x_k)$$

In order to compute fibers, we move to the complex numbers:

$$\begin{aligned} \tilde{\varphi} : SO(3) \times \mathbb{R}^2 \times (\mathbb{P}^3)^k &\rightarrow (\mathbb{P}^3)^k \times (\mathbb{P}^2)^k \\ (R, v, w, x_1, \dots, x_k) &\mapsto \varphi([I|0], [R|\check{v}], x_1, \dots, x_k) \end{aligned}$$

$$\left\{ \begin{array}{l} \text{complexify} \\ \downarrow \end{array} \right. \{ R \in \mathbb{C}^{3 \times 3} \mid R^T R = I, \det(R) = 1 \}$$

$$\begin{aligned} \tilde{\varphi}_{\mathbb{C}} : SO(3, \mathbb{C}) \times \mathbb{C}^2 \times (\mathbb{P}_{\mathbb{C}}^3)^k &\rightarrow (\mathbb{P}_{\mathbb{C}}^3)^k \times (\mathbb{P}_{\mathbb{C}}^2)^k \\ (R, v, w, x_1, \dots, x_k) &\mapsto ([I|0]x_1, \dots, [I|0]x_k, [R|\check{v}]x_1, \dots, [R|\check{v}]x_k) \end{aligned}$$



In order to compute fibers, we move to the complex numbers:

$$\begin{aligned} \tilde{\varphi} : SO(3) \times \mathbb{R}^2 \times (\mathbb{P}^3)^k &\rightarrow (\mathbb{P}^3)^k \times (\mathbb{P}^2)^k \\ (R, v, w, x_1, \dots, x_k) &\mapsto \varphi([I|0], [R|\check{v}], x_1, \dots, x_k) \end{aligned}$$

$$\left\{ \begin{array}{l} \text{complexify} \\ \downarrow \end{array} \right. \{ R \in \mathbb{C}^{3 \times 3} \mid R^T R = I, \det(R) = 1 \}$$

$$\begin{aligned} \tilde{\varphi}_{\mathbb{C}} : SO(3, \mathbb{C}) \times \mathbb{C}^2 \times (\mathbb{P}_{\mathbb{C}}^3)^k &\rightarrow (\mathbb{P}_{\mathbb{C}}^3)^k \times (\mathbb{P}_{\mathbb{C}}^2)^k \\ (R, v, w, x_1, \dots, x_k) &\mapsto ([I|0]x_1, \dots, [I|0]x_k, [R|\check{v}]x_1, \dots, [R|\check{v}]x_k) \end{aligned}$$

**Exercise:** For almost all  $(y, z) \in (\mathbb{P}_{\mathbb{C}}^3)^k \times (\mathbb{P}_{\mathbb{C}}^2)^k$ ,

$$|\tilde{\varphi}_{\mathbb{C}}^{-1}(y, z)| = \begin{cases} 0 & \text{if } k \geq 6 \\ \text{finite and } > 0 & \text{if } k = 5 \\ \infty & \text{if } k \leq 4 \end{cases}$$

In order to compute fibers, we move to the complex numbers:

$$\begin{aligned} \tilde{\varphi} : SO(3) \times \mathbb{R}^2 \times (\mathbb{P}^3)^k &\rightarrow (\mathbb{P}^2)^k \times (\mathbb{P}^2)^k \\ (R, v, w, x_1, \dots, x_k) &\mapsto \varphi([I|0], [R|\check{v}], x_1, \dots, x_k) \end{aligned}$$

$\left\{ \begin{array}{l} \text{complexify} \\ \downarrow \end{array} \right. \{ R \in \mathbb{C}^{3 \times 3} \mid R^T R = I, \det(R) = 1 \}$

$$\begin{aligned} \tilde{\varphi}_{\mathbb{C}} : SO(3, \mathbb{C}) \times \mathbb{C}^2 \times (\mathbb{P}_{\mathbb{C}}^3)^k &\rightarrow (\mathbb{P}_{\mathbb{C}}^2)^k \times (\mathbb{P}_{\mathbb{C}}^2)^k \\ (R, v, w, x_1, \dots, x_k) &\mapsto ([I|0]x_1, \dots, [I|0]x_k, [R|\check{v}]x_1, \dots, [R|\check{v}]x_k) \end{aligned}$$

**Exercise:** For almost all  $(y, z) \in (\mathbb{P}_{\mathbb{C}}^2)^k \times (\mathbb{P}_{\mathbb{C}}^2)^k$ ,

$$|\tilde{\varphi}_{\mathbb{C}}^{-1}(y, z)| = \begin{cases} 0 & \text{if } k \geq 6 \\ \text{finite and } > 0 & \text{if } k = 5 \\ \infty & \text{if } k \leq 4 \end{cases}$$



$$\tilde{\varphi}_{\mathbb{C}} : \mathrm{SO}(3, \mathbb{C}) \times \mathbb{C}^2 \times (\mathbb{P}_{\mathbb{C}}^3)^k \rightarrow (\mathbb{P}_{\mathbb{C}}^2)^k \times (\mathbb{P}_{\mathbb{C}}^2)^k$$

$$(\mathbb{R}, v, w, x_1, \dots, x_k) \mapsto ([I|0]x_1, \dots, [I|0]x_k, [R|\check{v}]x_1, \dots, [R|\check{v}]x_k)$$

Exercise: let  $k=5$ .

a) The Zariski closure of  $\varphi_{\mathbb{C}}(\underbrace{\Delta_{\mathbb{C}}}_{\substack{\text{complexification} \\ \text{of } \varphi}} \times \underbrace{(\mathbb{P}_{\mathbb{C}}^3)^k}_{\substack{\text{"bad"} \\ \text{camera pairs}}})$  is a proper subset of  $(\mathbb{P}_{\mathbb{C}}^2)^k \times (\mathbb{P}_{\mathbb{C}}^2)^k$ .

$$\tilde{\varphi}_{\mathbb{C}} : \mathrm{SO}(3, \mathbb{C}) \times \mathbb{C}^2 \times (\mathbb{P}_{\mathbb{C}}^3)^k \rightarrow (\mathbb{P}_{\mathbb{C}}^2)^k \times (\mathbb{P}_{\mathbb{C}}^2)^k$$

$$(R, v, w, x_1, \dots, x_k) \mapsto ([I|0]x_1, \dots, [I|0]x_k, [R|v]x_1, \dots, [R|v]x_k)$$

Exercise: let  $k=5$ .

a) The Zariski closure of  $\varphi_{\mathbb{C}}(\underbrace{\Delta}_{\text{complexification of } \varphi} \times \underbrace{(\mathbb{P}_{\mathbb{C}}^3)^k}_{\text{"bad" camera pairs}})$  is a proper subset of  $(\mathbb{P}_{\mathbb{C}}^2)^k \times (\mathbb{P}_{\mathbb{C}}^2)^k$ .

$\Rightarrow$  For almost all  $(y, z) \in (\mathbb{P}_{\mathbb{C}}^2)^k \times (\mathbb{P}_{\mathbb{C}}^2)^k$ :

$\rightarrow \varphi_{\mathbb{C}}^{-1}(y, z)$  consists of cameras outside of  $\Delta$

$\rightarrow$  We can safely compute  $\tilde{\varphi}_{\mathbb{C}}^{-1}(y, z)$  instead  $\smile$

$\rightarrow$  Formally:  $\varphi_{\mathbb{C}}^{-1}(y, z)$  is the  $G$ -orbit of  $\tilde{\varphi}_{\mathbb{C}}^{-1}(y, z)$ , i.e.

$$\varphi_{\mathbb{C}}^{-1}(y, z) = \{ g \cdot ([I|0], [R|v], x_1, \dots, x_k) \mid g \in G, (R, v, w, x_1, \dots, x_k) \in \tilde{\varphi}_{\mathbb{C}}^{-1}(y, z) \}$$

$\sim$



$$\tilde{\varphi}_{\mathbb{C}} : \mathrm{SO}(3, \mathbb{C}) \times \mathbb{C}^2 \times (\mathbb{P}_{\mathbb{C}}^3)^k \rightarrow (\mathbb{P}_{\mathbb{C}}^2)^k \times (\mathbb{P}_{\mathbb{C}}^2)^k$$

$$(R, v, w, x_1, \dots, x_k) \mapsto ([I|0]x_1, \dots, [I|0]x_k, [R|v]x_1, \dots, [R|v]x_k)$$

Exercise: let  $k=5$ .

a) The Zariski closure of  $\varphi_{\mathbb{C}}(\Delta \times (\mathbb{P}_{\mathbb{C}}^3)^k)$  is a proper subset of  $(\mathbb{P}_{\mathbb{C}}^2)^k \times (\mathbb{P}_{\mathbb{C}}^2)^k$ .

$\Delta$   $\xleftarrow{\text{complexification of } \varphi}$   $\Delta \times (\mathbb{P}_{\mathbb{C}}^3)^k$   $\xleftarrow{\text{"bad" camera pairs}}$

$\Rightarrow$  For almost all  $(y, z) \in (\mathbb{P}_{\mathbb{C}}^2)^k \times (\mathbb{P}_{\mathbb{C}}^2)^k$ :

$\rightarrow \varphi_{\mathbb{C}}^{-1}(y, z)$  consists of cameras outside of  $\Delta$

$\rightarrow$  We can safely compute  $\tilde{\varphi}_{\mathbb{C}}^{-1}(y, z)$  instead  $\smile$

$\rightarrow$  Formally:  $\varphi_{\mathbb{C}}^{-1}(y, z)$  is the  $G$ -orbit of  $\tilde{\varphi}_{\mathbb{C}}^{-1}(y, z)$ , i.e.

$$\varphi_{\mathbb{C}}^{-1}(y, z) = \{ g \cdot ([I|0], [R|v], x_1, \dots, x_k) \mid g \in G, (R, v, w, x_1, \dots, x_k) \in \tilde{\varphi}_{\mathbb{C}}^{-1}(y, z) \}$$

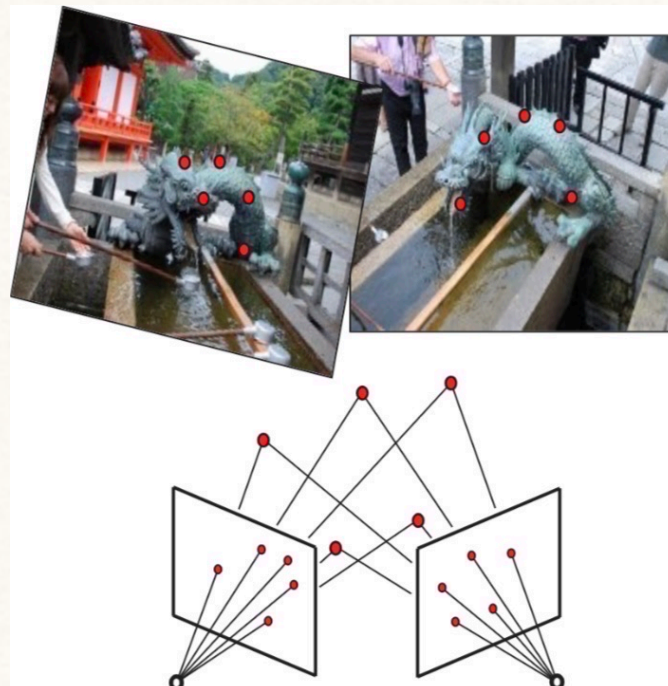
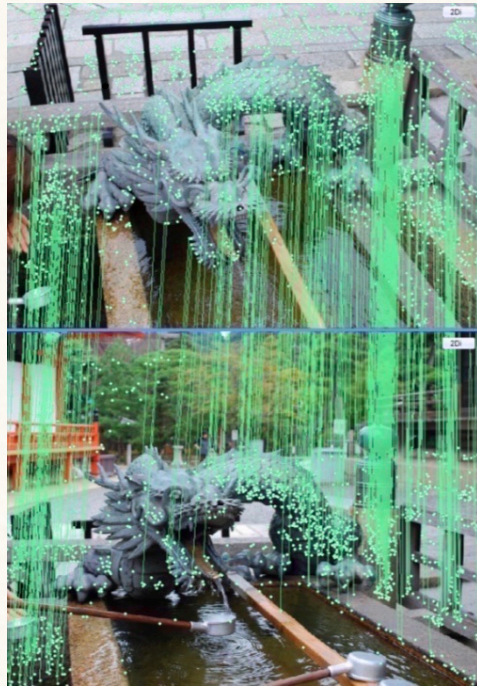
b) For almost all  $(y, z) \in (\mathbb{P}_{\mathbb{C}}^2)^k \times (\mathbb{P}_{\mathbb{C}}^2)^k$ :  $|\tilde{\varphi}_{\mathbb{C}}^{-1}(y, z)| = 20$ .



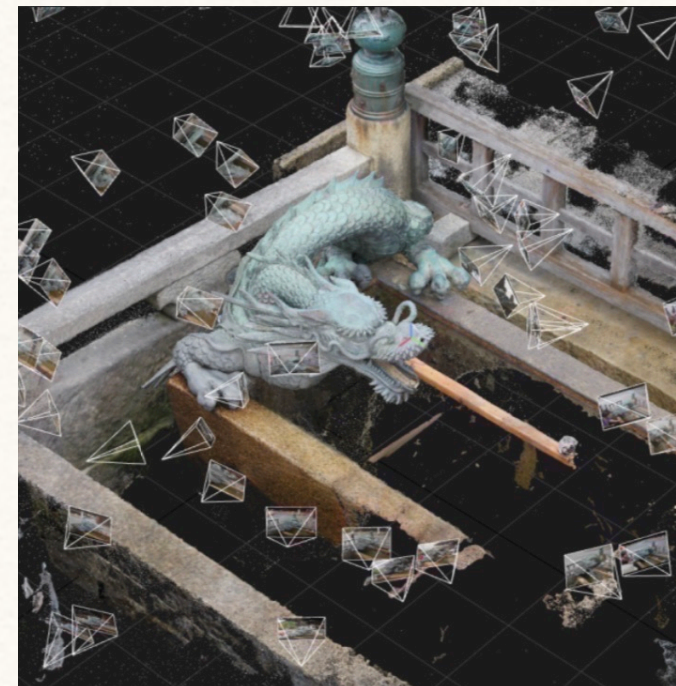
# SfM pipeline

14

Given: pictures



Output: 3D model  
+ camera poses



Step 1:

Identify corresponding points

Typically many matching points on many images are found!

Step 2:

forget pictures,  
solve for the 3D points & the cameras

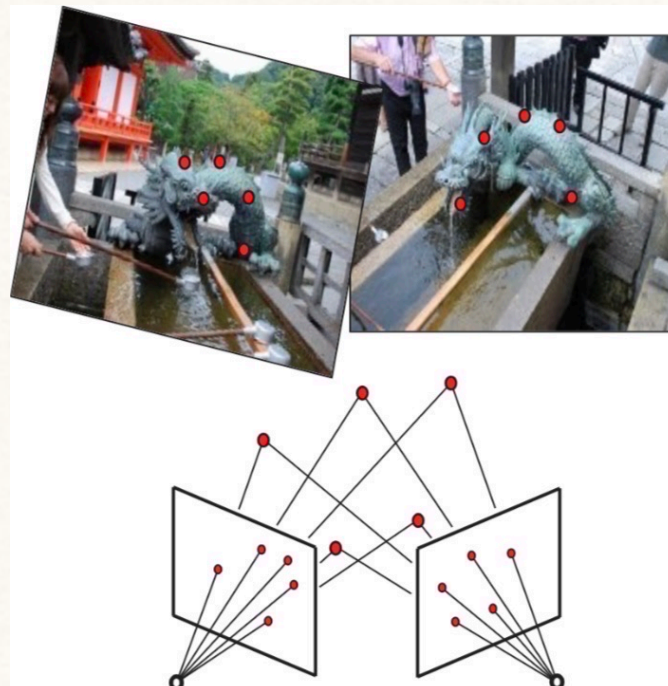
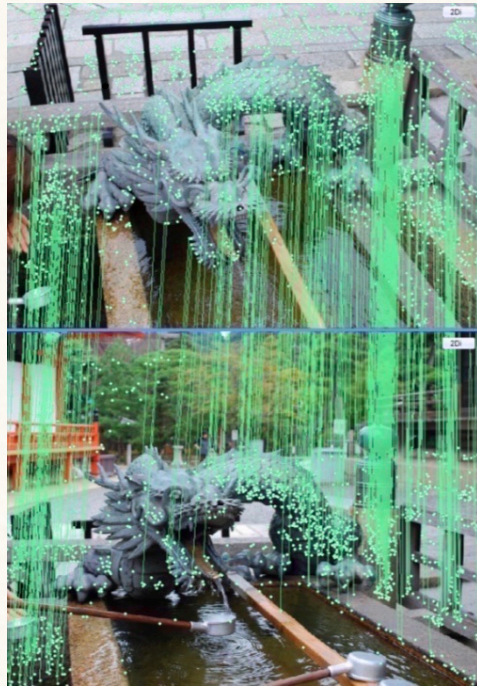
☺ algebraic inverse problem!



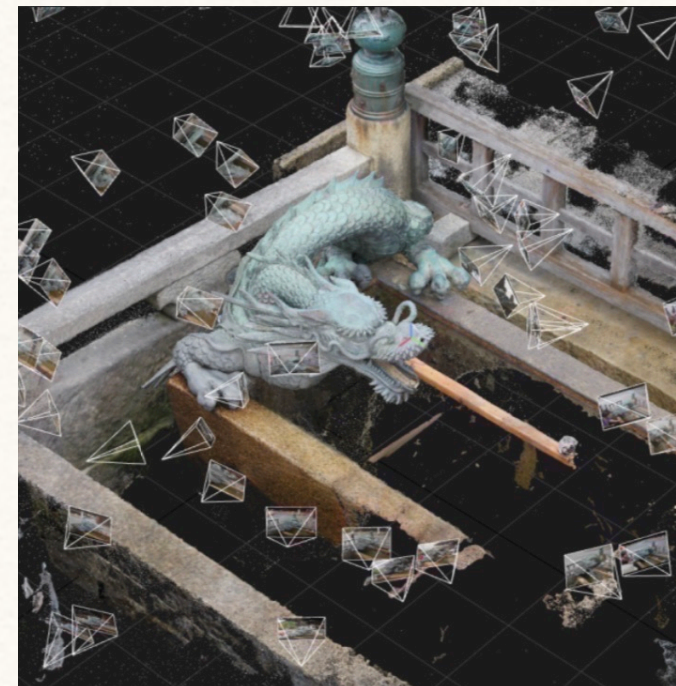
# SfM pipeline

14

Given: pictures



Output: 3D model  
+ camera poses



Step 1:

Identify corresponding points

Typically many matching points on many images are found!

Step 2:

forget pictures,  
solve for the 3D points & the cameras

☺ algebraic inverse problem!

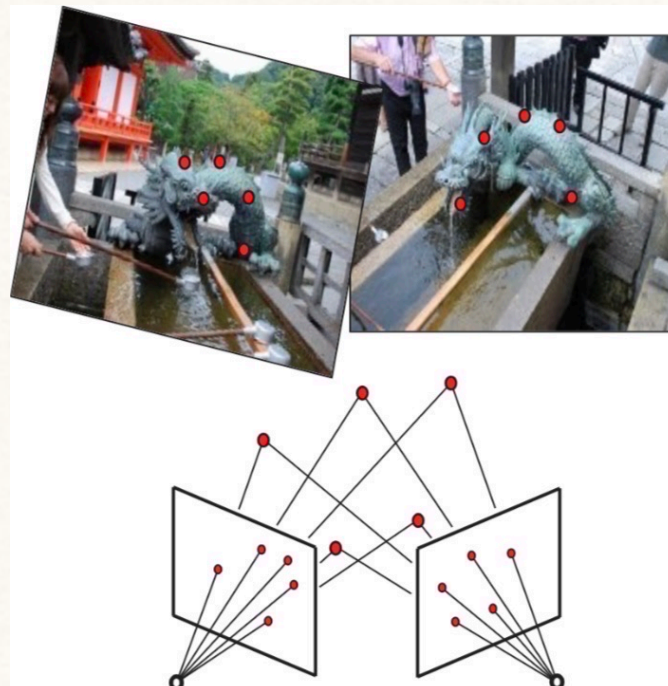
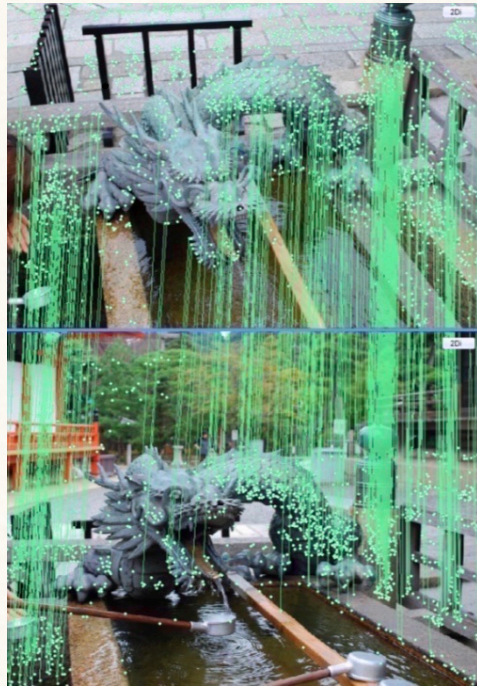
So, why did I propose to use 5 point matches on 2 images only (?)



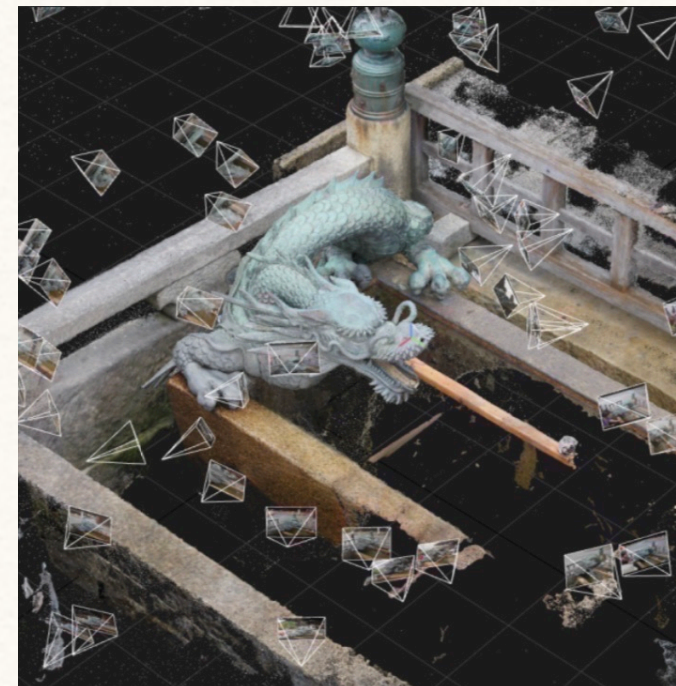
# SfM pipeline

14

Given: pictures



Output: 3D model  
+ camera poses



Step 1:

Identify corresponding points

Typically many matching points on many images are found!

Step 2:

forget pictures,  
solve for the 3D points & the cameras

😊 algebraic inverse problem!

Problem: point correspondences are **noisy** & contain **outliers** 😞



## Problem 1: Noise

Measurements are always noisy!

## Problem 1: Noise

Measurements are always noisy!

Example:  $k=6$

$$\tilde{\varphi}_c : \mathcal{SO}(3, \mathbb{C}) \times \mathbb{C}^2 \times (\mathbb{P}_{\mathbb{C}}^3)^k \rightarrow (\mathbb{P}_{\mathbb{C}}^2)^k \times (\mathbb{P}_{\mathbb{C}}^2)^k$$

$$(R, v, w, \underbrace{x_1, \dots, x_k}_{=X}) \mapsto ([I|0]x_1, \dots, [I|0]x_k, [R|v]x_1, \dots, [R|v]x_k)$$

- Fix camera parameters  $R, v, w$  & 3D points  $X$ .
- Compute image points  $(y, z) = \tilde{\varphi}_c(R, v, w, X)$
- Measure image points  $\rightsquigarrow$  obtain perturbation  $(\tilde{y}, \tilde{z})$  of  $(y, z)$
- There is no solution to  $(\tilde{y}, \tilde{z})$ ! (i.e.,  $\tilde{\varphi}_c^{-1}(\tilde{y}, \tilde{z}) = \emptyset$ ) previous exercise



# Problem 1: Noise

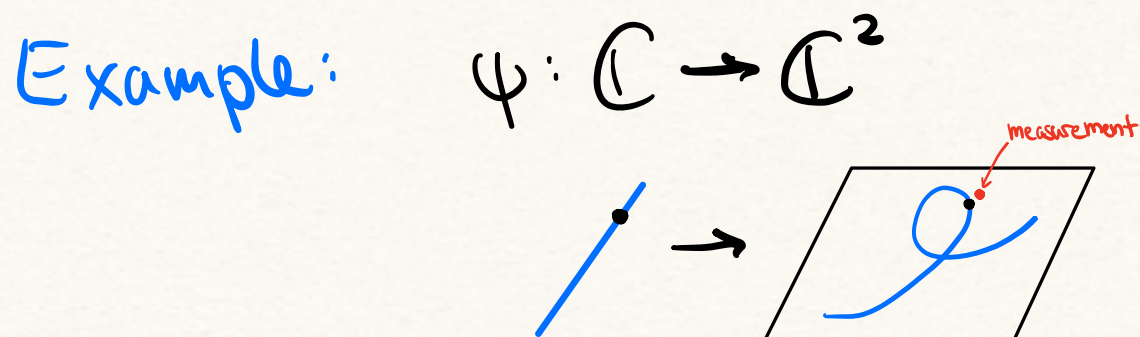
Measurements are always noisy!

Example:  $k=6$

$$\tilde{\varphi}_c : \mathcal{SO}(3, \mathbb{C}) \times \mathbb{C}^2 \times (\mathbb{P}_{\mathbb{C}}^3)^k \rightarrow (\mathbb{P}_{\mathbb{C}}^2)^k \times (\mathbb{P}_{\mathbb{C}}^2)^k$$

$$(R, v, w, \underbrace{x_1, \dots, x_k}_{=X}) \mapsto ([I|0]x_1, \dots, [I|0]x_k, [R|v]x_1, \dots, [R|v]x_k)$$

- Fix camera parameters  $R, v, w$  & 3D points  $X$ .
- Compute image points  $(y, z) = \tilde{\varphi}_c(R, v, w, X)$
- Measure image points  $\rightsquigarrow$  obtain perturbation  $(\tilde{y}, \tilde{z})$  of  $(y, z)$
- There is no solution to  $(\tilde{y}, \tilde{z})$ ! (i.e.,  $\tilde{\varphi}_c^{-1}(\tilde{y}, \tilde{z}) = \emptyset$ ) previous exercise



# Problem 1: Noise

Measurements are always noisy!

Example:  $k=6$

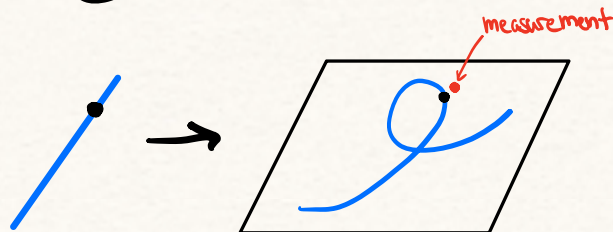
$$\tilde{\varphi}_c : \mathcal{SO}(3, \mathbb{C}) \times \mathbb{C}^2 \times (\mathbb{P}_{\mathbb{C}}^3)^k \rightarrow (\mathbb{P}_{\mathbb{C}}^2)^k \times (\mathbb{P}_{\mathbb{C}}^2)^k$$

$$(R, v, w, \underbrace{x_1, \dots, x_k}_{=X}) \mapsto ([I|0]x_1, \dots, [I|0]x_k, [R|v]x_1, \dots, [R|v]x_k)$$

- Fix camera parameters  $R, v, w$  & 3D points  $X$ .
- Compute image points  $(y, z) = \tilde{\varphi}_c(R, v, w, X)$
- Measure image points  $\rightsquigarrow$  obtain perturbation  $(\tilde{y}, \tilde{z})$  of  $(y, z)$
- There is no solution to  $(\tilde{y}, \tilde{z})$ ! (i.e.,  $\tilde{\varphi}_c^{-1}(\tilde{y}, \tilde{z}) = \emptyset$ ) previous exercise

Example:

$$\psi : \mathbb{C} \rightarrow \mathbb{C}^2$$



In these examples: Given measurements, you first need to denoise them to get a point on the image variety before you can solve the inverse problem.



# Problem 1: Noise

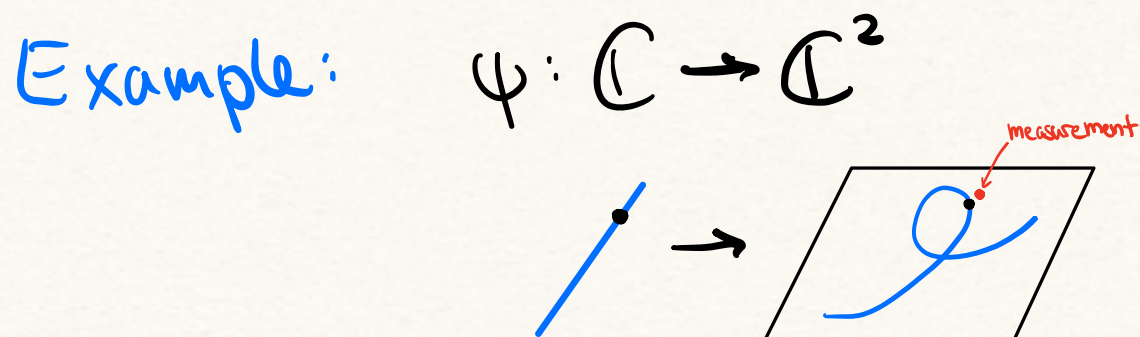
Measurements are always noisy!

Example:  $k=6$

$$\tilde{\varphi}_c : \mathcal{SO}(3, \mathbb{C}) \times \mathbb{C}^2 \times (\mathbb{P}_{\mathbb{C}}^3)^k \rightarrow (\mathbb{P}_{\mathbb{C}}^2)^k \times (\mathbb{P}_{\mathbb{C}}^2)^k$$

$$(R, v, w, \underbrace{x_1, \dots, x_k}_{=X}) \mapsto ([I|0]x_1, \dots, [I|0]x_k, [R|v]x_1, \dots, [R|v]x_k)$$

- Fix camera parameters  $R, v, w$  & 3D points  $X$ .
- Compute image points  $(y, z) = \tilde{\varphi}_c(R, v, w, X)$
- Measure image points  $\rightsquigarrow$  obtain perturbation  $(\tilde{y}, \tilde{z})$  of  $(y, z)$
- There is no solution to  $(\tilde{y}, \tilde{z})$ ! (i.e.,  $\tilde{\varphi}_c^{-1}(\tilde{y}, \tilde{z}) = \emptyset$ ) previous exercise



In these examples: Given measurements, you first need to **denoise** them to get a point on the image variety before you can solve the inverse problem.

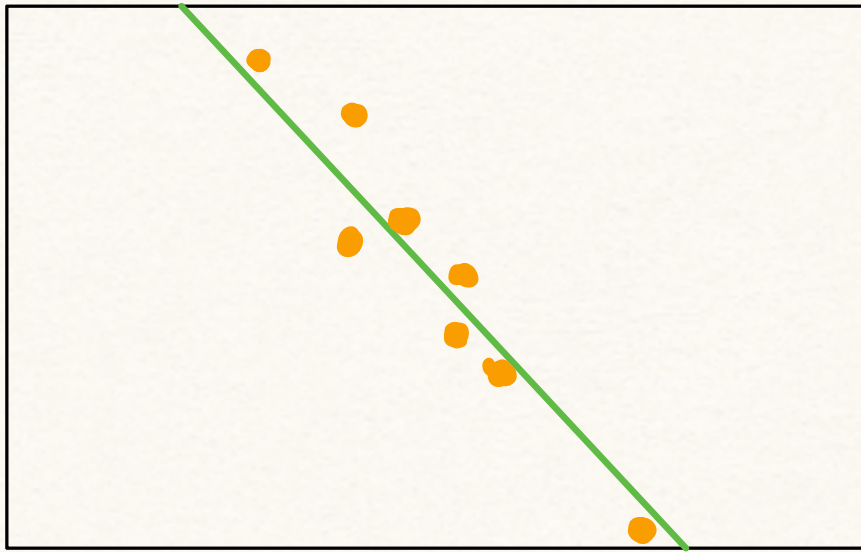
optimization problem: find closest point on image variety

## Problem 1: Noise

Measurements are always noisy!

Example: fitting a line to **noisy** points

linear  
regression



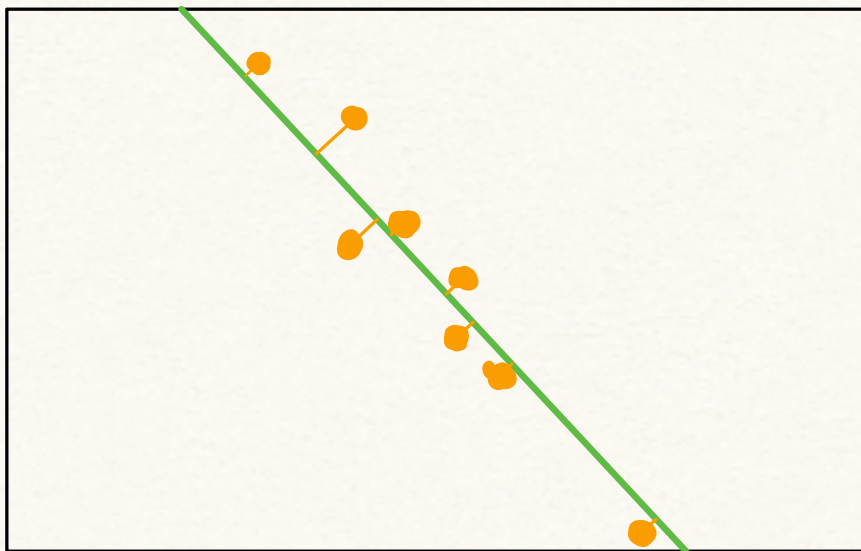


# Problem 1: Noise

Measurements are always noisy!

Example: fitting a line to **noisy** points

linear  
regression



$$\operatorname{argmin}_{L \subseteq \mathbb{R}^2 \text{ line}} \sum_{i=1}^k \operatorname{dist}(x_i, L)^2$$

$$\left. \begin{array}{l} \{(L, x_1, \dots, x_k) \mid L \subseteq \mathbb{R}^2 \text{ line}, x_i \in L\} \longrightarrow (\mathbb{R}^2)^k \\ (L, x_1, \dots, x_k) \longmapsto (x_1, \dots, x_k) \end{array} \right\} \begin{array}{l} \text{image variety} \\ = \{\text{collinear points}\} \end{array}$$

the bigger  
problem! →

## Problem 2: Outliers

17

Outliers in data are common in general.

In Structure-from-Motion, 30% of the data being outliers is not unusual.



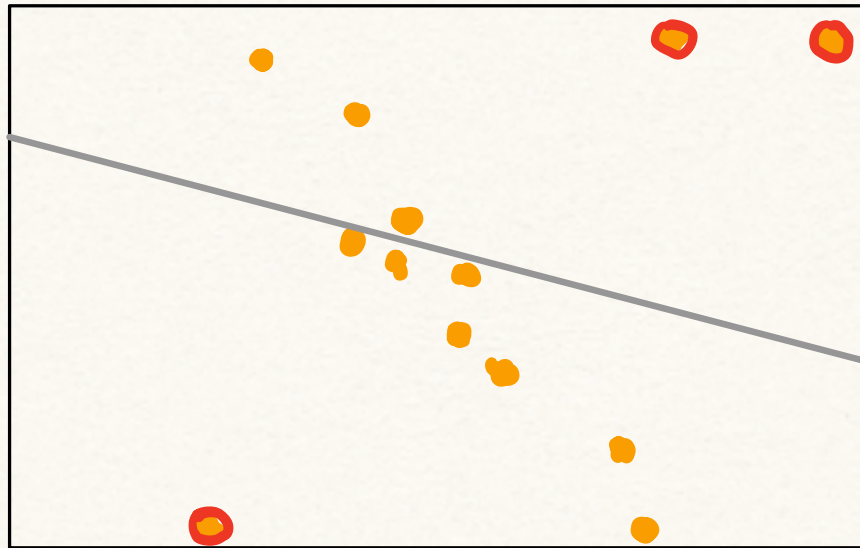
the bigger  
problem! →

## Problem 2: Outliers

Outliers in data are common in general.

In Structure-from-Motion, 30% of the data being outliers is not unusual.

Example:



linear regression  
does not find the  
best line anymore...

What to do  
with outliers  
?

## Problem 2: Outliers

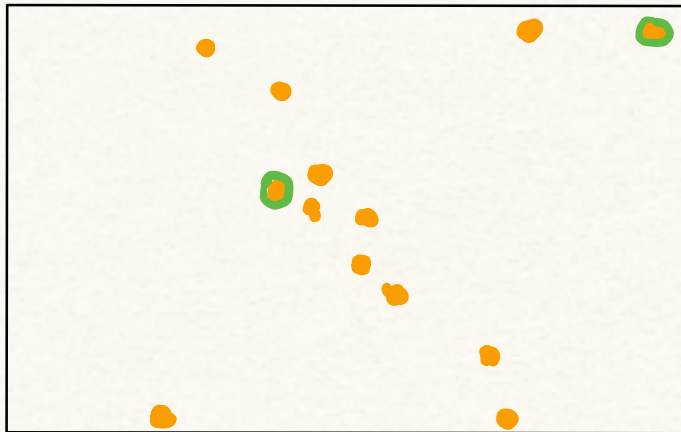
RANSAC (RANDOM Sample Consensus) provides robust estimation!



## Problem 2: Outliers

RANSAC (RANDOM Sample Consensus) provides robust estimation!

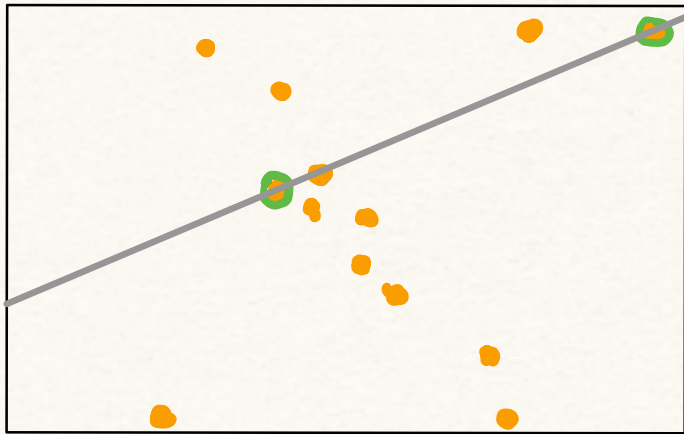
① Randomly select a subset of the data



## Problem 2: Outliers

RANSAC (RANDOM Sample Consensus) provides robust estimation!

- ① Randomly select a subset of the data
- ② Fit a model to the selected subset

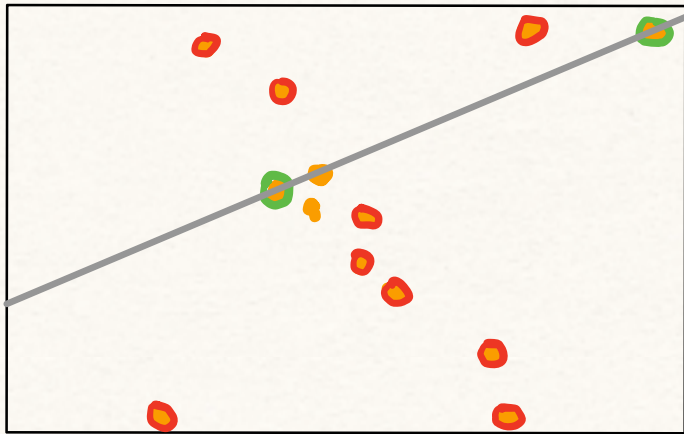




## Problem 2: Outliers

RANSAC (RANDOM Sample Consensus) provides robust estimation!

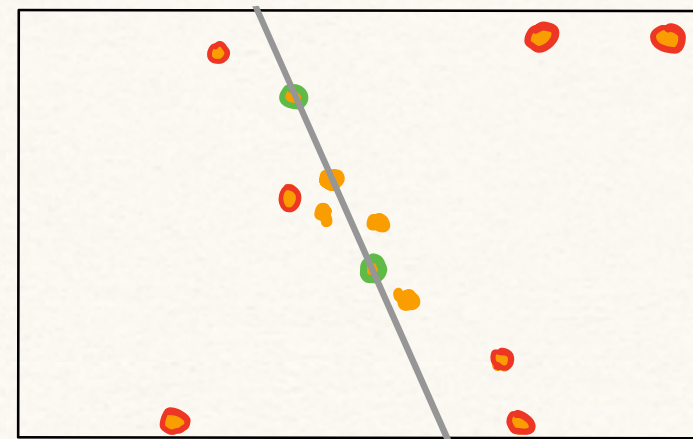
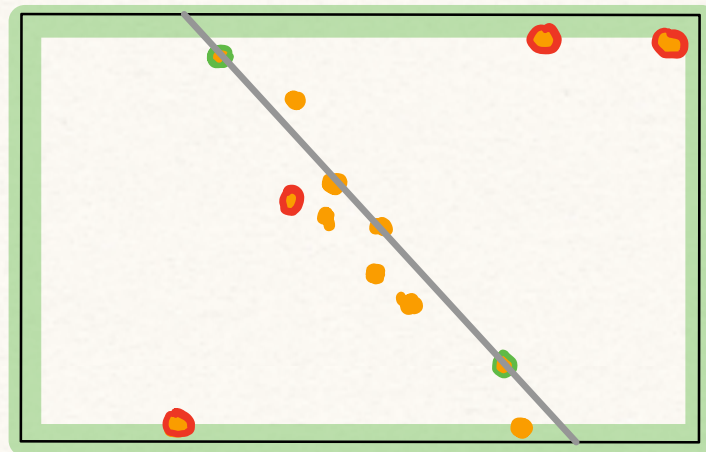
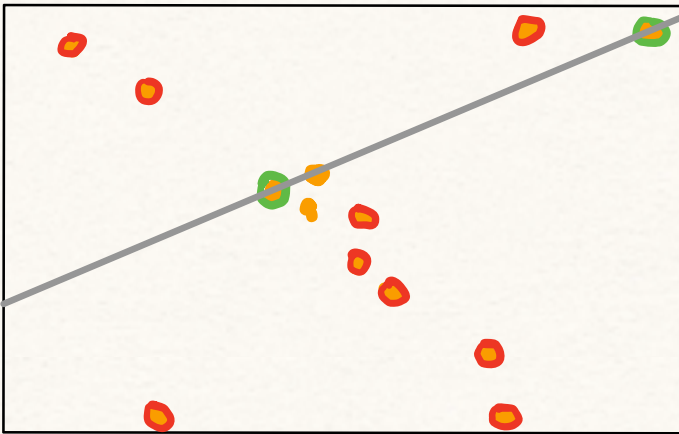
- ① Randomly select a subset of the data
- ② Fit a model to the selected subset
- ③ Determine the number of outliers (= data points further away than a fixed threshold from the model found in ②)



## Problem 2: Outliers

RANSAC (RANDOM SAMPLE CONSENSUS) provides robust estimation!

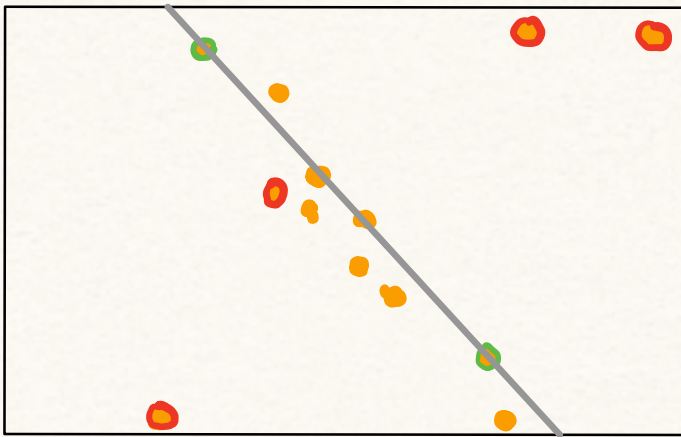
- ① Randomly select a subset of the data
- ② Fit a model to the selected subset
- ③ Determine the number of outliers (= data points further away than a fixed threshold from the model found in ②)
- ④ Repeat ①-③ until you find a model with few number of outliers





# RANSAC...

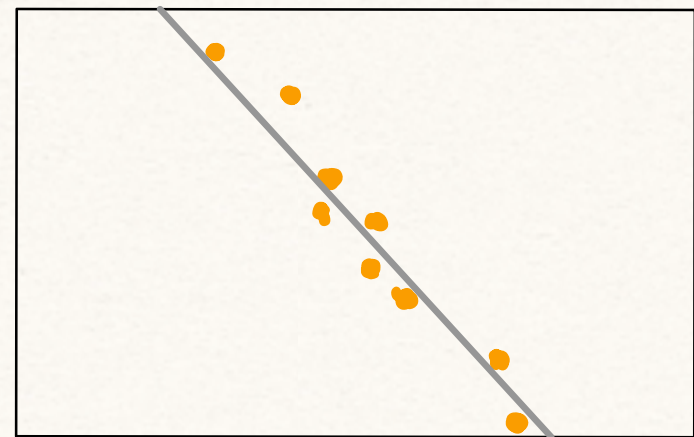
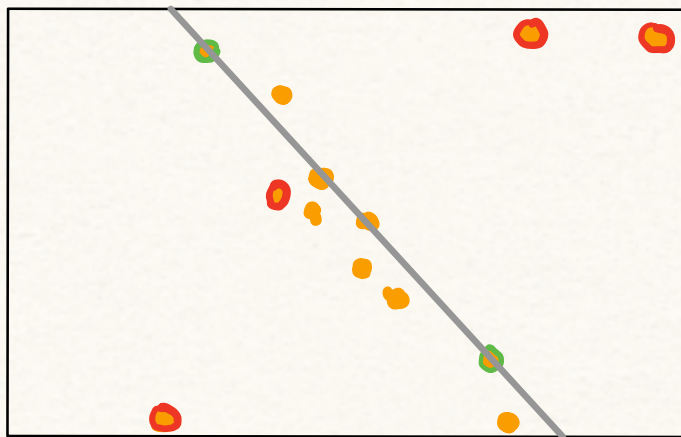
... determines a good model & outliers in the data



# RANSAC...

... determines a good model & outliers in the data

Post processing: Delete the determined outliers & and optimize the found model over all remaining data





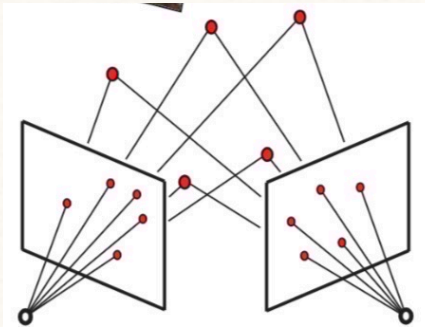
# RANSAC for overconstrained polynomial systems

more non-redundant equations than variables

- ① Randomly select a square sub-system with finitely many solutions
- ② Compute the solutions
- ③ For each solution: Determine how many of the equations discarded in ① are far from vanishing on the solution
- ④ Repeat ①-③ until you find a solution that almost satisfies many equations

# RANSAC for Structure from Motion

① Randomly select a subset of the corresponding points (or lines etc.) such that the resulting polynomial system has finitely many solutions



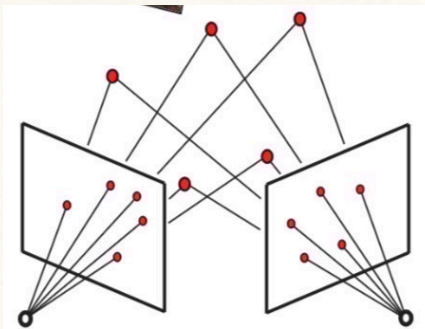
in our example with 2 cameras and  $k$  3D-points:

select 5 point pairs and get a system with 20 solutions



# RANSAC for Structure from Motion

- ① Randomly select a subset of the corresponding points (or lines etc.) such that the resulting polynomial system has finitely many solutions



in our example with 2 cameras and  $k$  3D-points:

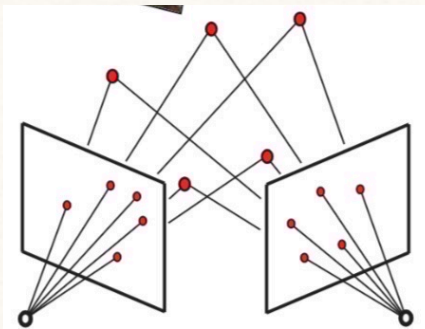
select 5 point pairs and get a system with 20 solutions

- ② Compute the solutions

each solution consists of 2 cameras & 5 3D-points

# RANSAC for Structure from Motion

- ① Randomly select a subset of the corresponding points (or lines etc.) such that the resulting polynomial system has finitely many solutions



in our example with 2 cameras and  $k$  3D-points:

select 5 point pairs and get a system with 20 solutions

- ② Compute the solutions

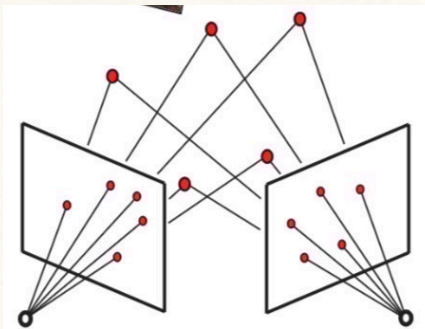
each solution consists of 2 cameras & 5 3D-points

- ③ For each solution: Determine how many given point pairs are far away from being a feasible point pair of the 2 cameras in the solution



# RANSAC for Structure from Motion

- ① Randomly select a subset of the corresponding points (or lines etc.) such that the resulting polynomial system has finitely many solutions



in our example with 2 cameras and  $k$  3D-points:

select 5 point pairs and get a system with 20 solutions

- ② Compute the solutions

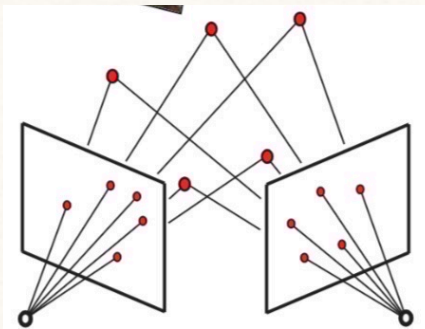
each solution consists of 2 cameras & 5 3D-points

- ③ For each solution: Determine how many given point pairs are far away from being a feasible point pair of the 2 cameras in the solution

- ④ Repeat ①-③ until you find a camera pair with many consistent point pairs  
 → find a good camera pair & which point matches were outliers

# RANSAC for Structure from Motion

- ① Randomly select a subset of the corresponding points (or lines etc.) such that the resulting polynomial system has finitely many solutions



in our example with 2 cameras and  $k$  3D-points:

select 5 point pairs and get a system with 20 solutions

- ② Compute the solutions

each solution consists of 2 cameras & 5 3D-points

- ③ For each solution: Determine how many given point pairs are far away from being a feasible point pair of the 2 cameras in the solution

- ④ Repeat ①-③ until you find a camera pair with many consistent point pairs

→ find a good camera pair & which point matches were outliers

→ post processing: optimization over all inlier data (=analog to linear regression)



(3) For each camera pair: Determine how many of the given point pairs are far away from being a feasible point pair of the 2 cameras

How do you do that?

(3) For each camera pair: Determine how many of the given point pairs are far away from being a feasible point pair of the 2 cameras

How do you do that?

Given cameras  $C_1$  &  $C_2$ , all feasible point pairs are in the image of

$$\begin{aligned} \varphi_{C_1, C_2}: \mathbb{P}^3 &\longrightarrow \mathbb{P}^2 \times \mathbb{P}^2 \\ x &\longmapsto (C_1 x, C_2 x). \end{aligned}$$



(3) For each camera pair: Determine how many of the given point pairs are far away from being a feasible point pair of the 2 cameras

How do you do that?

Given cameras  $C_1$  &  $C_2$ , all feasible point pairs are in the image of

$$\begin{aligned} \varphi_{C_1, C_2}: \mathbb{P}^3 &\rightarrow \mathbb{P}^2 \times \mathbb{P}^2 \\ x &\mapsto (C_1 x, C_2 x). \end{aligned}$$

Proposition:  $\overline{\text{im}(\varphi_{C_1, C_2})} = \left\{ (y, z) \in \mathbb{P}^2 \times \mathbb{P}^2 \mid \det \begin{bmatrix} C_1 & y & 0 \\ C_2 & 0 & z \end{bmatrix} = 0 \right\}$

(3) For each camera pair: Determine how many of the given point pairs are far away from being a feasible point pair of the 2 cameras

How do you do that?  $\leftarrow$  Given cameras  $C_1$  &  $C_2$  and image points  $y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$  &  $z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$ , compute  $\left| \det \begin{bmatrix} C_1 & y & 0 \\ C_2 & 0 & z \end{bmatrix} \right|$ .

Given cameras  $C_1$  &  $C_2$ , all feasible point pairs are in the image of

$$\begin{aligned} \varphi_{C_1, C_2}: \mathbb{P}^3 &\rightarrow \mathbb{P}^2 \times \mathbb{P}^2 \\ x &\mapsto (C_1 x, C_2 x). \end{aligned}$$

Proposition:  $\overline{\text{im}(\varphi_{C_1, C_2})} = \left\{ (y, z) \in \mathbb{P}^2 \times \mathbb{P}^2 \mid \det \begin{bmatrix} C_1 & y & 0 \\ C_2 & 0 & z \end{bmatrix} = 0 \right\}$



④ Repeat ①-③ until you find a camera pair with many consistent point pairs



How often?

23  
④ Repeat ①-③ until you find a camera pair with many consistent point pairs

↑  
How often?

Say 30% of the given point matches are outliers  
and in each repetition you select 5 matched point pairs.

How many repetitions do you need to be 99.9% sure  
that one of the selections did not contain any outliers?



④ Repeat ①-③ until you find a camera pair with many consistent point pairs

↑  
How often?

Say 30% of the given point matches are outliers  
and in each repetition you select 5 matched point pairs.

How many repetitions do you need to be 99.9% sure  
that one of the selections did not contain any outliers?

Exercise: 38 repetitions

⇒ Since Structure from Motion should run in (almost) real time (e.g. self-driving robots, ...) and each repetition involves solving a polynomial system, these systems have to be solved really fast!  
 < 1ms

