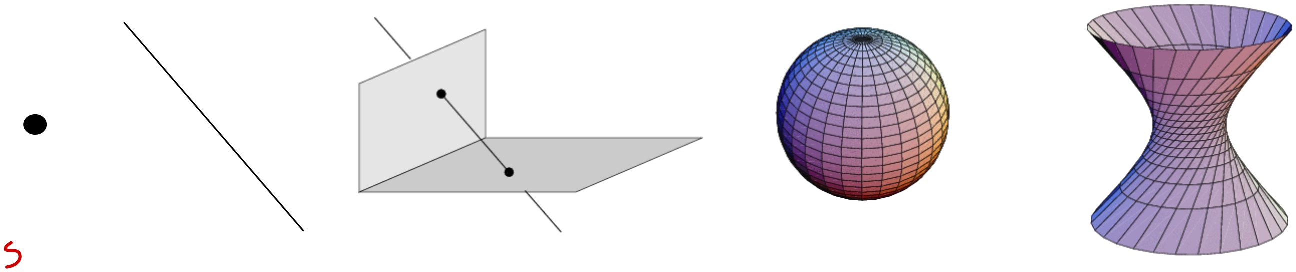
Space (3D) Projective Geometry

in most cases it is an extension of 3D geometry

3D Space Projective Geometry

- **Elements** 3D | 2D
  - **Points** ← points
  - Planes ← lines
  - Quadrics ← conics
  - (Dual quadrics) ← dual conics



the same as 2D TRANSFORMATIONS

- **Transformations**
  - Isometries
  - Similarities
  - Affinities
  - Projectivities

Points in the projective space

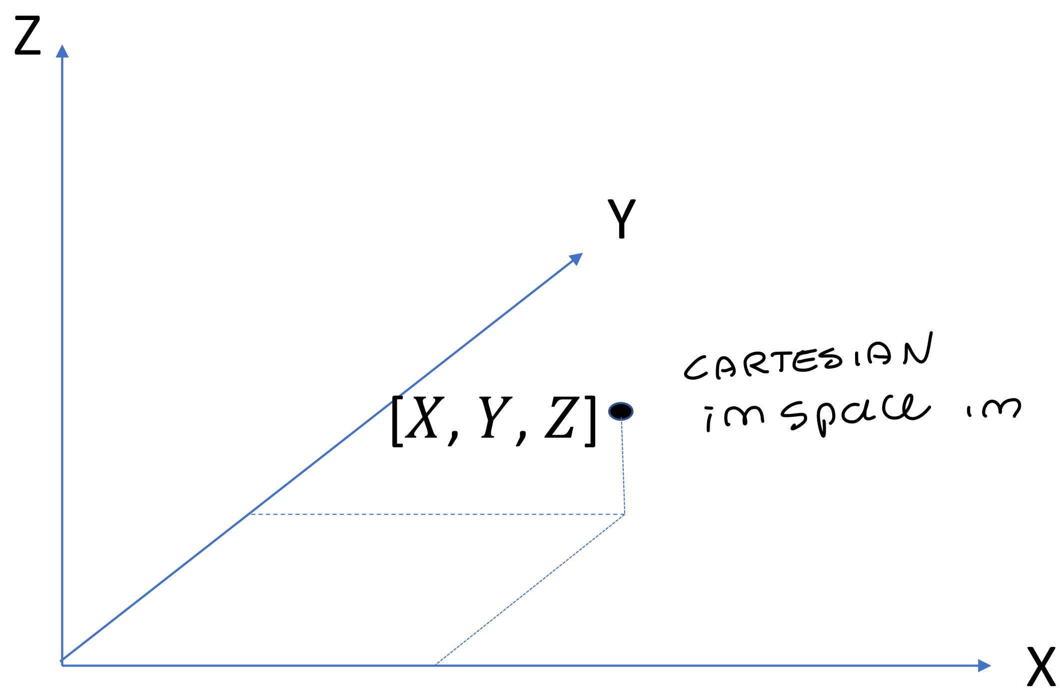
Euclidean space (3D) cartesian coordinates

as points in 2D, again in 3D we use

$p \in \mathbb{R}^3$ euclidean

$\downarrow$

homogeneous

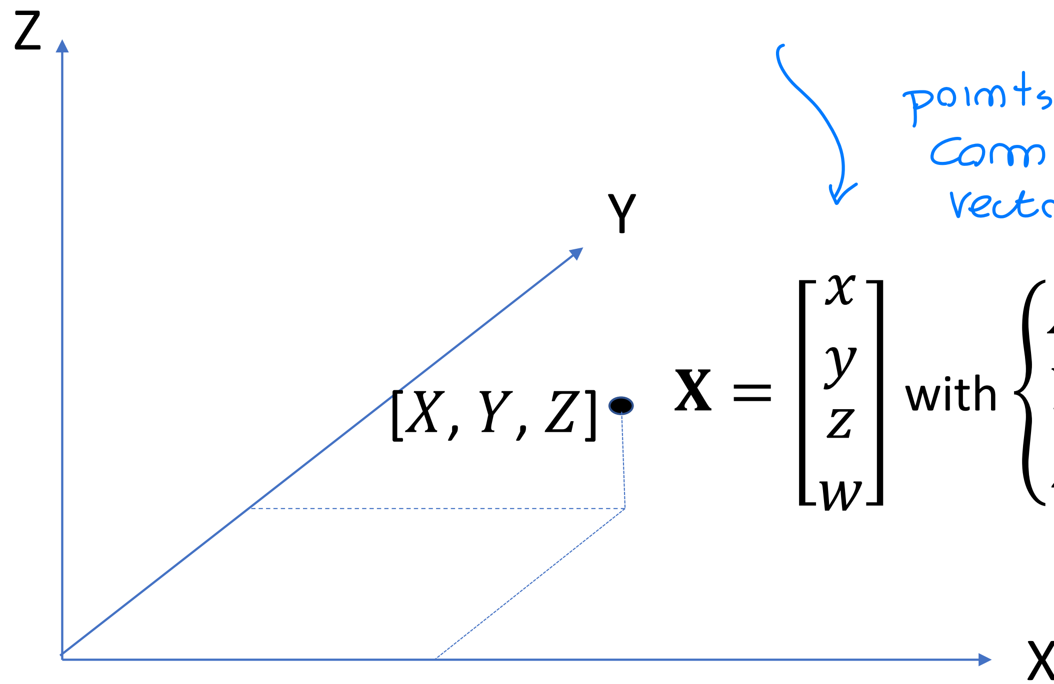


CARTESIAN

in space in XYZ reference

Projective space (3D)

4 homogeneous coordinates (same as 2D)

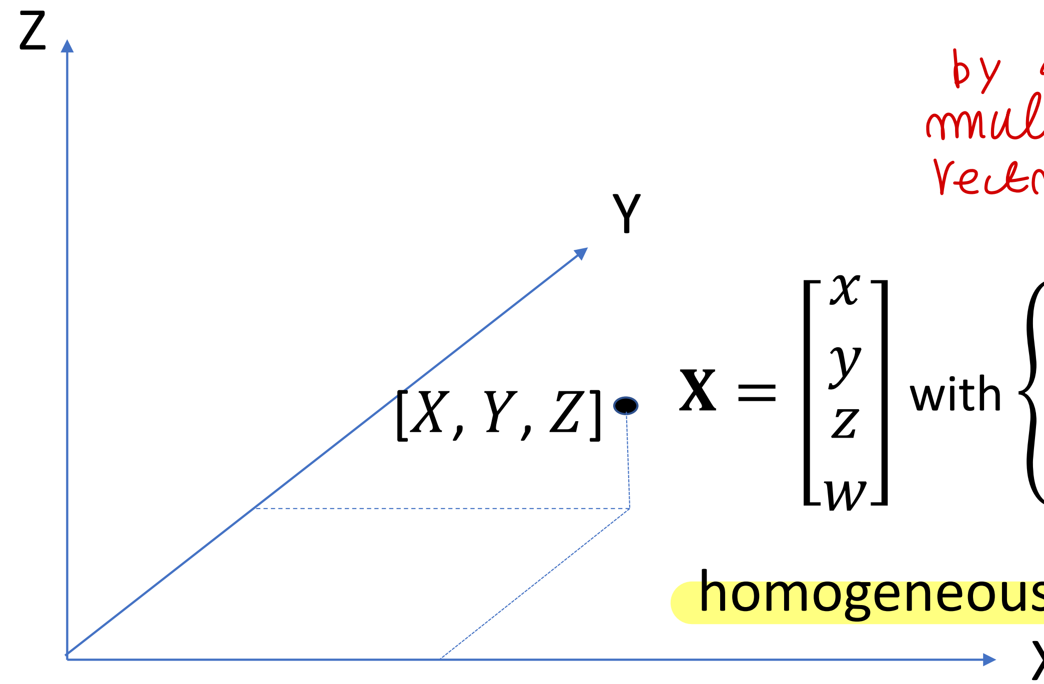


points by 4 components vector

$$\mathbf{X} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \text{ with } \begin{cases} X = x/w \\ Y = y/w \\ Z = z/w \end{cases}$$

EUCLIDEAN coordinate

Projective space (3D)
4 homogeneous coordinates



by common λ multiplication, the vector is the same...

$$\mathbf{X} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \text{ with } \begin{cases} X = x/w \\ Y = y/w \\ Z = z/w \end{cases}$$

homogeneous vector: $\lambda\mathbf{X} \Leftrightarrow \mathbf{X}$

same geometrical entity

Projective space $\mathbb{P}^3$: points at the ∞

($\mathbb{P}^2$ PROJECTIVE PLANE: Euclidean plane + X_∞)

$\downarrow$

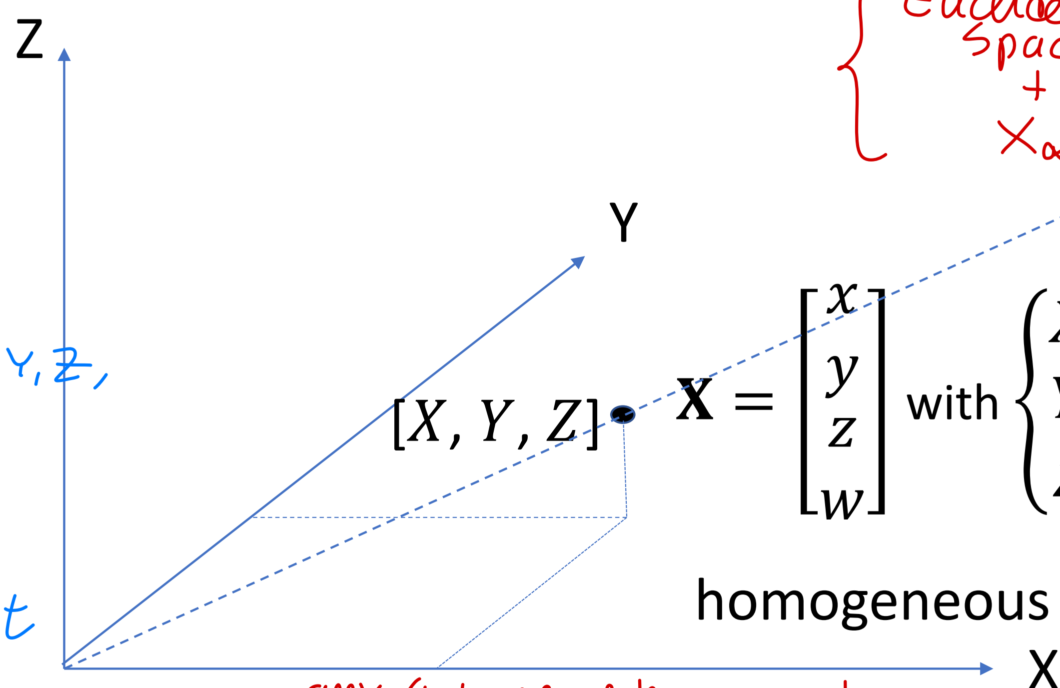
$\left\{ \begin{array}{c} \text{Euclidean space} \\ + \\ X_\infty \end{array} \right\}$

Far from origin

$$\mathbf{X}_\infty = \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}$$

points @ ∞ are the one with $w = 0$

having $[x, y, z] \to X_\infty$ has its direction x, y, z, as direction parameter of that point



$$\mathbf{X} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \text{ with } \begin{cases} X = x/w \to \infty \\ Y = y/w \to \infty \\ Z = z/w \to \infty \end{cases} \text{ when } w \to 0$$

homogeneous vector: $\lambda \mathbf{X} \Leftrightarrow \mathbf{X}$

any 4-touple of hom. coord

$$\mathbb{P}^3 = \{\mathbf{X} \in \mathbb{R}^4 - \{[0 \quad 0 \quad 0 \quad 0]^T\}\}$$

Except Null vector, which is NOTHING!

even if it seems to be a missmatch in DOF,

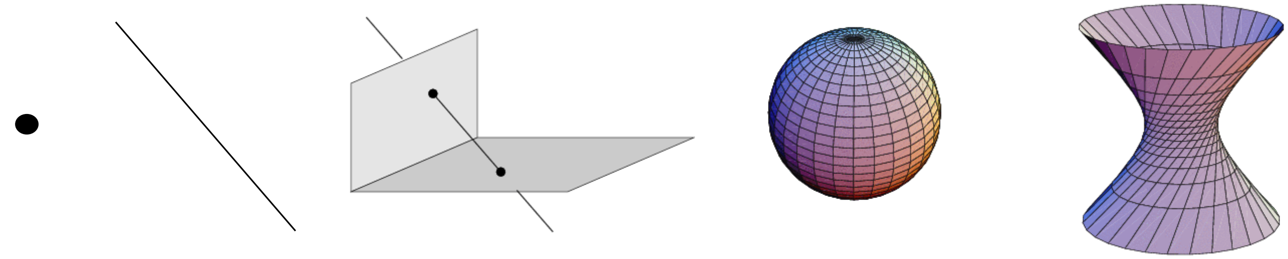
being $p \in \mathbb{R}^4$ BUT because of homogeneity

all x are such that λx is the same point

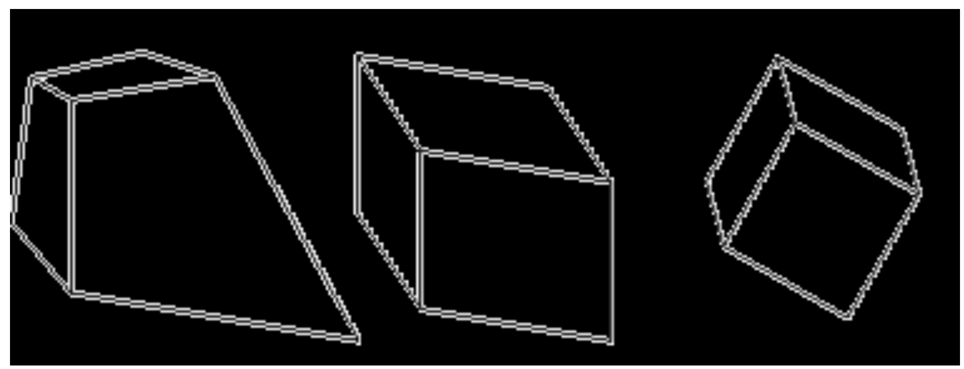
redundancy... 4 hom. instead of 3, NOT missmatch

3D Space Projective Geometry

- **Elements**
- Points
- **Planes**
- Quadrics
- (Dual quadrics)



- **Transformations**
- Isometries
- Similarities
- Affinities
- Projectivities

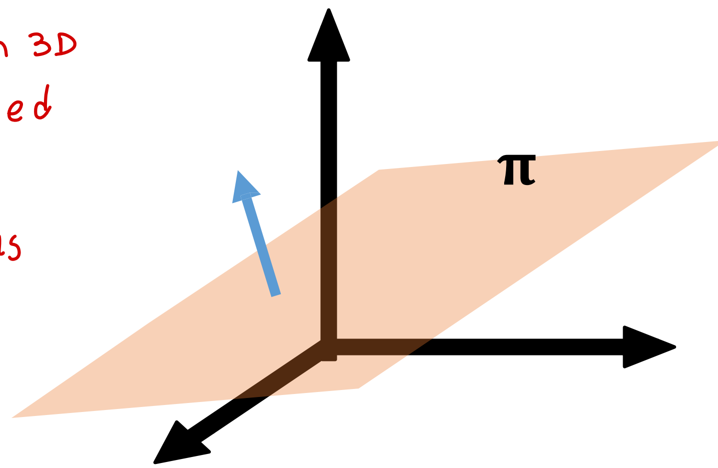


Planes in the projective space

(instead of "lines", we work with plane)

Planes in 3D Projective Geometry

plane in 3D represented by 4 parameters



given $(a, b, c, d)^T$ you place π in space as

even $\lambda\pi$ has same distance

$$\boldsymbol{\pi} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$ with direction $(a \quad b \quad c)$ normal to the plane,

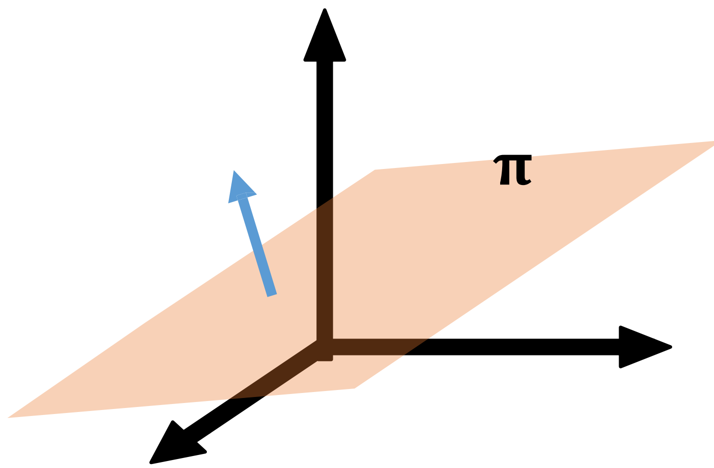
$d \sim$ distance between plane and origin $d = 0 \Rightarrow$ plane at O
$d \neq 0 \Rightarrow$ plane away

and $\frac{-d}{\sqrt{a^2+b^2+c^2}}$ is the distance between the origin and the plane

divided by modulus to be all homogeneous, NOT increasing if λ multiplicative factor

$\boldsymbol{\pi}$ is a homogeneous vector: $\lambda\boldsymbol{\pi} \Leftrightarrow \boldsymbol{\pi}$

Planes in 3D Projective Geometry



$\boldsymbol{\pi} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$ with direction $(a \quad b \quad c)$ normal to the plane,

$ax + by + cz + dw = 0$ then distance solving has (-) sign

and $\dfrac{-d}{\sqrt{a^2+b^2+c^2}}$ is the distance between the origin and the plane

$\boldsymbol{\pi}$ is a homogeneous vector: $\lambda\boldsymbol{\pi} \Leftrightarrow \boldsymbol{\pi}$

in cartesian coordinate

$$aX + bY + cZ + d = 0$$

in <u>homogeneus</u> $X = x/w$

$$\frac{a}{w}x + \frac{b}{w}y + \frac{c}{w}z + d = 0$$

than $ax + by + cz + dw = 0$

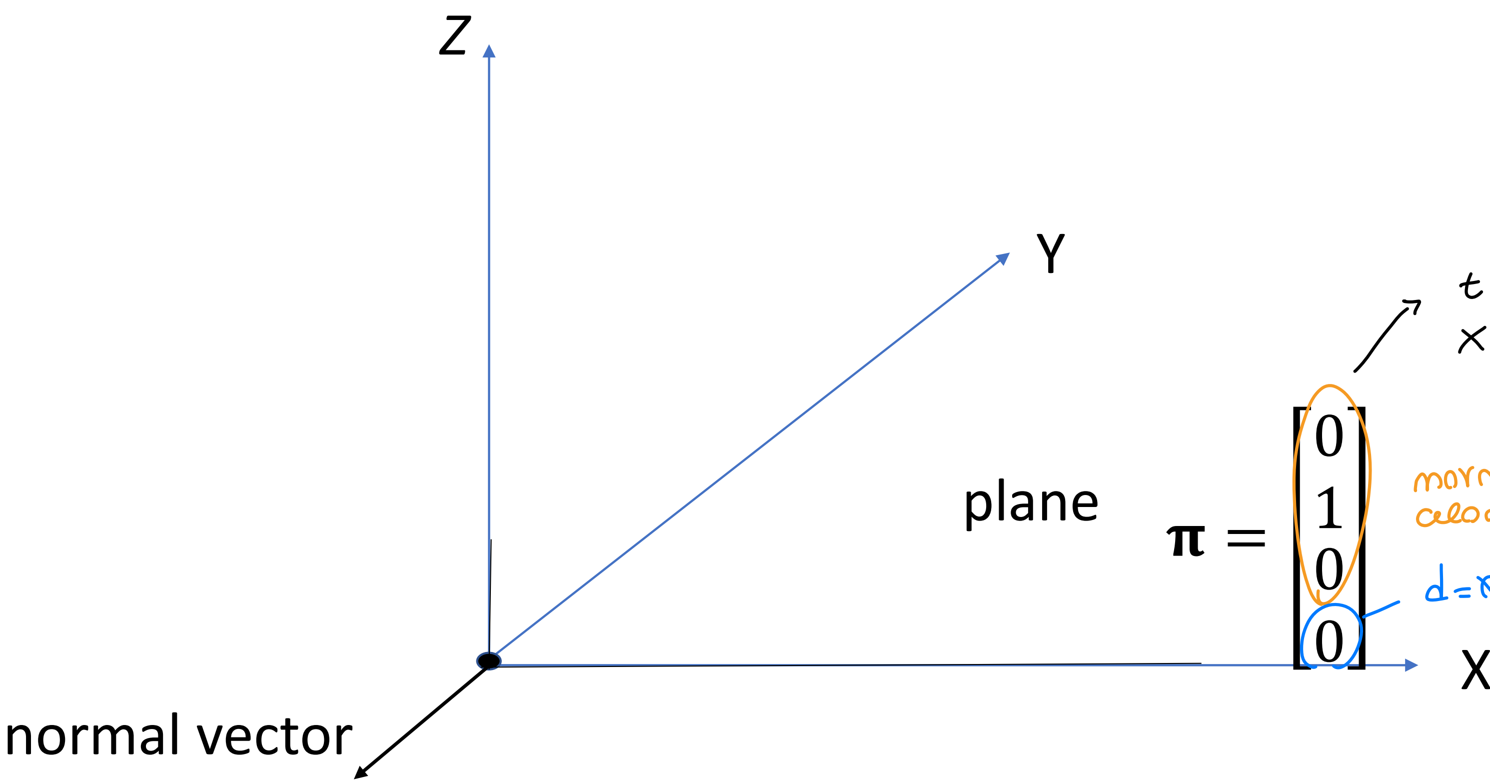
$[a \quad b \quad c \quad d]^T = \pi$

$[x \quad y \quad z \quad w]^T = X$ hom coord

$\pi X = 0$ means $x \in \pi$

OR π goes through x

Example: the X-Z plane



plane

$$\boldsymbol{\pi} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

this is XZ plane

normal along y

d=0 from origin

The incidence relation:

a point is on a plane, or a plane goes through a point

mathematical

Representation $\Rightarrow$

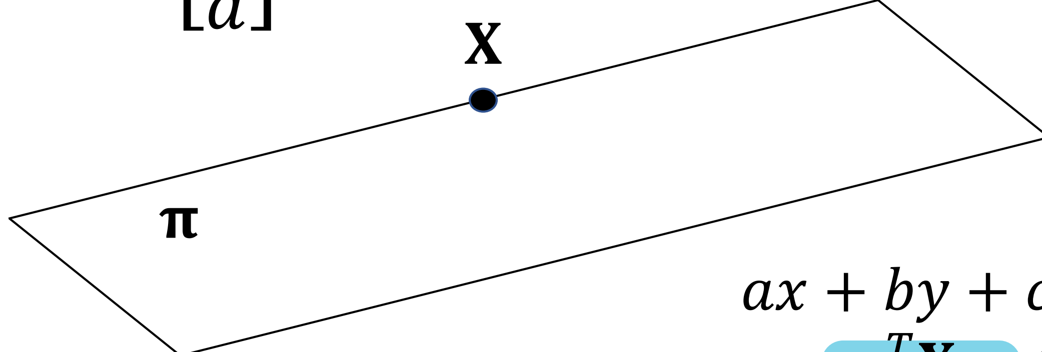
Incidence relation

- the point $\mathbf{X} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$ is on the plane $\boldsymbol{\pi} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$

same..

or

- the plane $\boldsymbol{\pi} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$ goes through the point $\mathbf{X} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$



$$ax + by + cz + dw = 0$$

$$\boldsymbol{\pi}^T \mathbf{X} = 0 = \mathbf{X}^T \boldsymbol{\pi}$$

Dividing by $\boldsymbol{w}$ we find the cartesian coordinates again

The plane at the infinity:
the locus of the points at the infinity

set of X_∞
(as extension of ℓ_∞ is π_∞ a plane)

The «locus» of the points at the infinity

As there are infinite points at the infinity (one for each direction), what is the aspect of the set of these points at the infinity?

Simply: $w = 0$

linear homog equation = PLANE

π_∞ as the set of points solving this

This set is a plane: $\begin{bmatrix} \boldsymbol{a} & \boldsymbol{b} & \boldsymbol{c} & \boldsymbol{d} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \\ \boldsymbol{w} \end{bmatrix} = 0$, actually $\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \\ \boldsymbol{w} \end{bmatrix} = 0$

homogeneus equation of degree one.. is an equation of a plane!

namely, **the plane at the infinity $\boldsymbol{\pi}_\infty$** $= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$

anything

NOTE: this plane has undefined normal direction

The duality principle between points and planes

duality

we didn't study lines in 3D because, for lines in space,
a line has 4 DoF as { 2 dof for intersection
{ +2 for direction

theorem stay
that: there is NO unique representation of line 3D.
you cannot represent line in unique way which only uses
4 parameters (NO minimal representation ∀ line)

Using 2 DOF + direction

↓

NOT valid for some lines...
NOT for some line that
doesn't have unique intersection...

↓

Also... extending planar geometry line you
get a PLANE!

↓

lines are complex object in 3D!

Duality comes from the fact that

Since dot product is commutative

→ incidence relation is commutative

$$\boldsymbol{\pi}^T\mathbf{X} = [\boldsymbol{a} \quad \boldsymbol{b} \quad \boldsymbol{c} \quad \boldsymbol{d}]\begin{bmatrix}\boldsymbol{x}\\\boldsymbol{y}\\\boldsymbol{z}\\\boldsymbol{w}\end{bmatrix} = 0 = [\boldsymbol{x} \quad \boldsymbol{y} \quad \boldsymbol{z} \quad \boldsymbol{w}]\begin{bmatrix}\boldsymbol{a}\\\boldsymbol{b}\\\boldsymbol{c}\\\boldsymbol{d}\end{bmatrix} = 0 = \mathbf{X}^T\boldsymbol{\pi} = \mathbf{0}$$

point $\mathbf{X}$ is on plane $\boldsymbol{\pi}$

$\Leftrightarrow$

point $\boldsymbol{\pi}$ is on **plane** $\mathbf{X}$

DUALITY! you can interchange the roles!

point $\mathbf{X}$ is on plane $\boldsymbol{\pi}$ (i.e. plane $\boldsymbol{\pi}$ goes through point $\mathbf{X}$)

$\Longleftrightarrow$

point $\boldsymbol{\pi}$ is on **line** $\mathbf{X}$ (i.e. line $\mathbf{X}$ goes through point $\boldsymbol{\pi}$)

no need to proof ... this holds since product is scalar, and $(.)^T$ is itself ... than $(\pi^T X)^T = 0 = X^T \pi$ same holds, DUALITY!

Principle of **duality** between points and planes in 3D Projective Geometry

For any true sentence containing the words

- point
- plane
- is on
- goes through

there is a DUAL sentence -also true- obtained by substituting, in the previous one, each occurence of

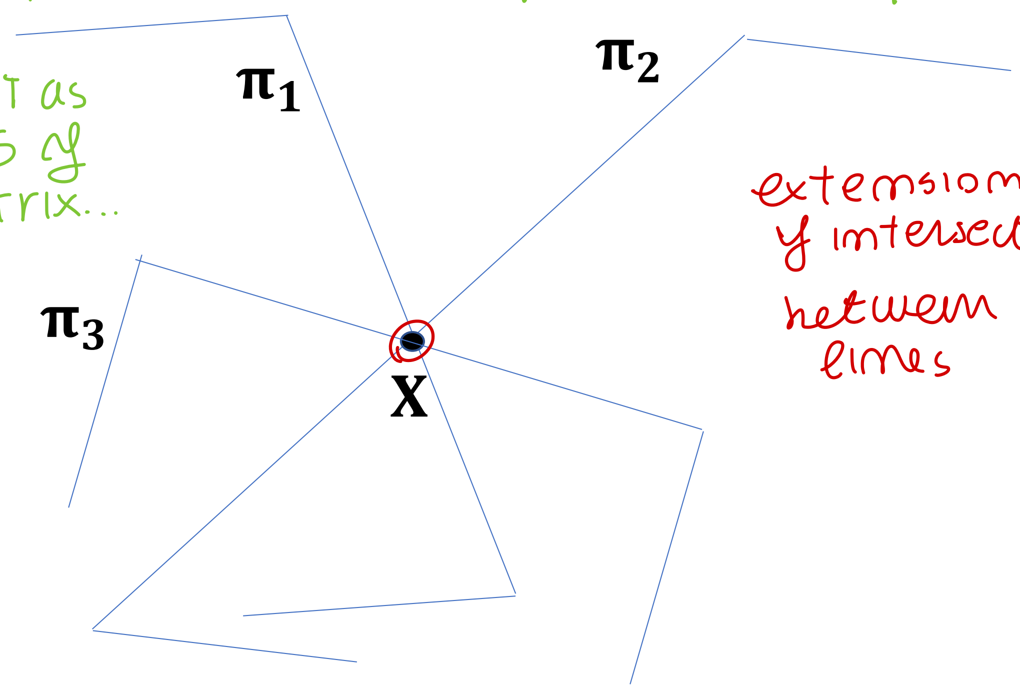
- point	by	- plane
- plane	by	- point
- is on	by	- goes through
- goes through	by	- is on

The point on three planes

The point on three planes

while intersection between lines was found by CROSS PRODUCT, here no special vector product...

↳ just as RNS of matrix...



extension of intersection between lines

intersection of three planes is a POINT

↕

by three points only one plane going through

$$\begin{cases} \boldsymbol{\pi}_1{}^T\mathbf{X} = 0 \\ \boldsymbol{\pi}_2{}^T\mathbf{X} = 0 \\ \boldsymbol{\pi}_3{}^T\mathbf{X} = 0 \end{cases}$$

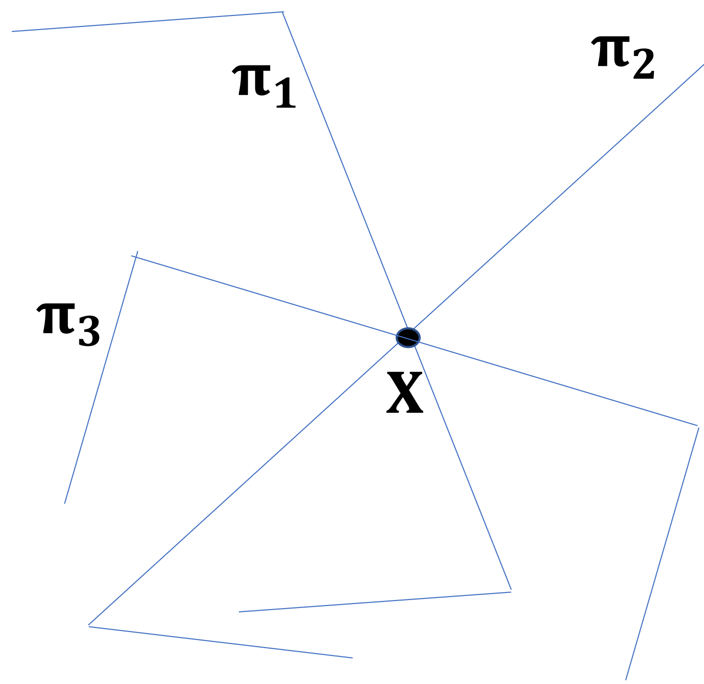
$$\underbrace{\begin{bmatrix} \boldsymbol{\pi}_1{}^T \\ \boldsymbol{\pi}_2{}^T \\ \boldsymbol{\pi}_3{}^T \end{bmatrix}}_{3 \times 4} \mathbf{X} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{X} = \mathbf{RNS}\left(\begin{bmatrix} \boldsymbol{\pi}_1{}^T \\ \boldsymbol{\pi}_2{}^T \\ \boldsymbol{\pi}_3{}^T \end{bmatrix}_{3 \times 4}\right)$$

set of vectors that multiplied matrix give $\underline{0}$

a solution vector + all its mutiples

The point on three planes



$$\begin{cases} \boldsymbol{\pi_1}^T \mathbf{X} = 0 \\ \boldsymbol{\pi_2}^T \mathbf{X} = 0 \\ \boldsymbol{\pi_3}^T \mathbf{X} = 0 \end{cases}$$

$$\underbrace{\begin{bmatrix} \boldsymbol{\pi_1}^T \\ \boldsymbol{\pi_2}^T \\ \boldsymbol{\pi_3}^T \end{bmatrix}}_{3 \times 4} \mathbf{X} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{X} = \mathbf{RNS}\left(\underbrace{\begin{bmatrix} \boldsymbol{\pi_1}^T \\ \boldsymbol{\pi_2}^T \\ \boldsymbol{\pi_3}^T \end{bmatrix}}_{3 \times 4} \right)$$

a solution vector + all its mutiples

having system of 4 unkn and 3 eq... ∞ solution

Algebrically! BUT being homogeneus solution, you have 1 vector and all multiples

↓

JUST one point

x and λx all homogeneus ∞ algebraic

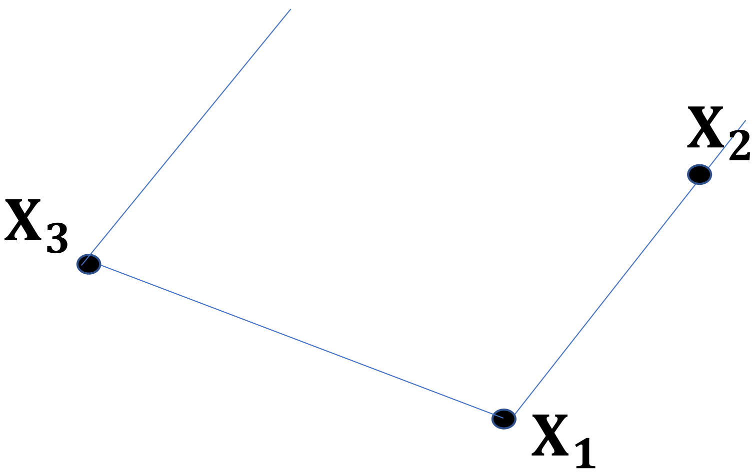
DUAL

The plane through three points

The plane through three points: DUAL of the point on three planes

only one plane going through 3 points

The plane through three points



dual of before

$$\begin{cases} \mathbf{X_1}^T \boldsymbol{\pi} = 0 \\ \mathbf{X_2}^T \boldsymbol{\pi} = 0 \\ \mathbf{X_3}^T \boldsymbol{\pi} = 0 \end{cases}$$

IF $X_i \in \Pi$
than $X_i^T \Pi = 0$
same $\forall X_i$
$i = 1,2,3$

MATRIX

$$\begin{bmatrix} \mathbf{X_1}^T \\ \mathbf{X_2}^T \\ \mathbf{X_3}^T \end{bmatrix}_{3 \times 4} \boldsymbol{\pi} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

representing same plane up to a multiplicative factor

$$\boldsymbol{\pi} = \mathbf{RNS}\left(\begin{bmatrix} \mathbf{X_1}^T \\ \mathbf{X_2}^T \\ \mathbf{X_3}^T \end{bmatrix}_{3 \times 4} \right)$$

a solution vector + all its mutiples

Lines

WHAT ABOUT LINES ?

Lines are primitive elements in the planar geometry

but they are **not** primitive elements in the **space** geometry

in SPACE primitive are POINTS, PLANES

Lines can be defined through points or through planes.

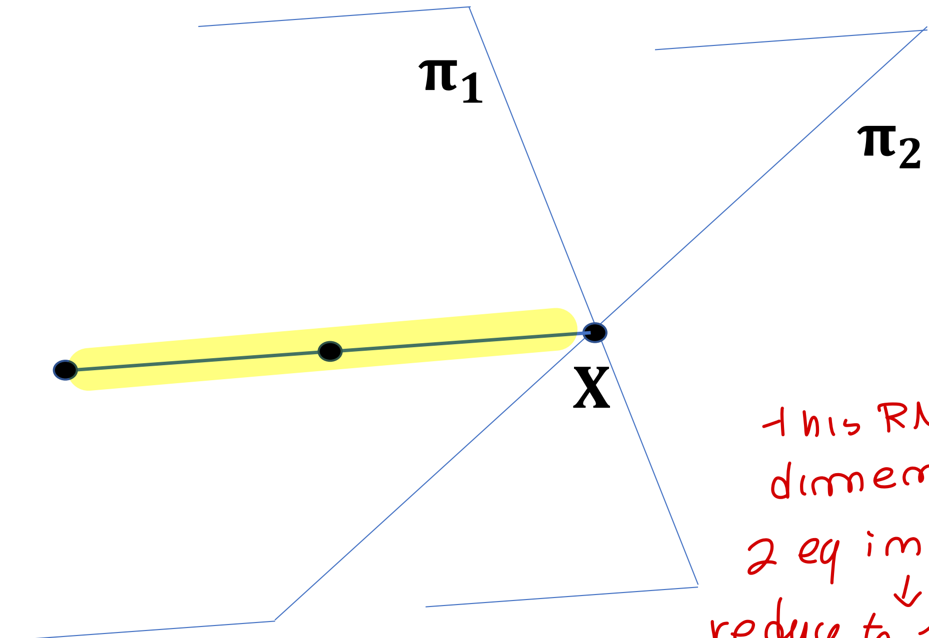
Lines are intermediate entities between points and planes

(they are self-dual)

dual of a line is itself ! they are intermediate between points & planes

Line: the set of points $\mathbf{X}$ on **two** planes

(as intersection of two planes)



$$\begin{cases} \boldsymbol{\pi}_1{}^T\mathbf{X} = 0 \\ \boldsymbol{\pi}_2{}^T\mathbf{X} = 0 \end{cases}$$

intersection π_1, π_2

$$\underbrace{\begin{bmatrix} \boldsymbol{\pi}_1{}^T \\ \boldsymbol{\pi}_2{}^T \end{bmatrix}}_{2 \times 4} \mathbf{X} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

this RNS has dimension 2, being 2 eq in 4 unkn

↓

reduce to 1D infinite!

set of sol as 1D space

$$\mathbf{L} = \mathbf{RNS}\left(\underbrace{\begin{bmatrix} \boldsymbol{\pi}_1{}^T \\ \boldsymbol{\pi}_2{}^T \end{bmatrix}}_{2 \times 4}\right)$$

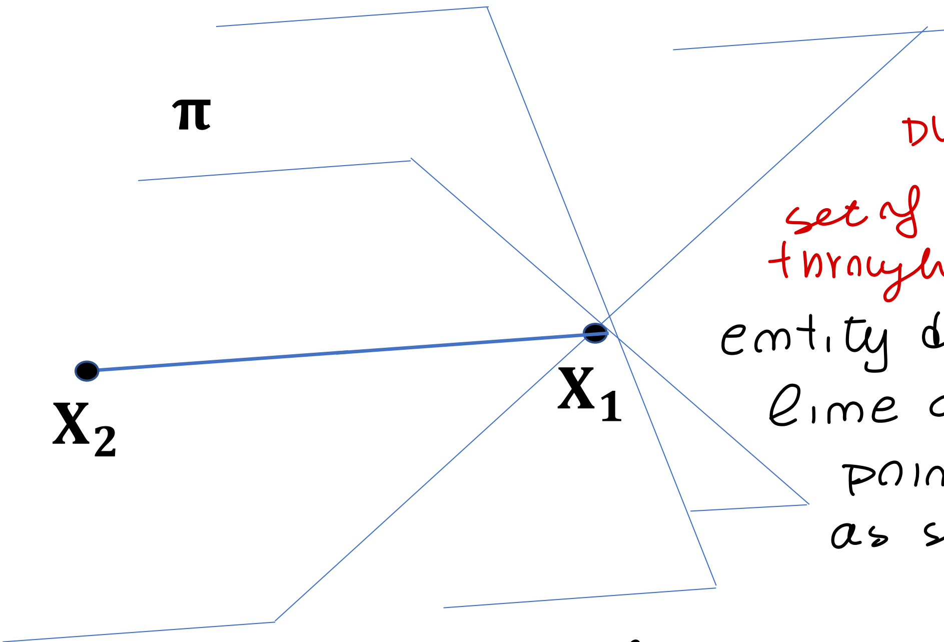
2D set of solution vectors: two points and all their linear combinations

→ due to homogeneity: 1D set of points (parameter abscissa)

Line: the set of planes $\boldsymbol{\pi}$ through **two** points

dual line representation, as set π plane goin through 2 points

$$\begin{cases} \mathbf{X_1}^T \boldsymbol{\pi} = 0 \\ \mathbf{X_2}^T \boldsymbol{\pi} = 0 \end{cases}$$



DUAL of before!

set of planes π through 2 points

entity different...

$$\underbrace{\begin{bmatrix} \mathbf{X_1}^T \\ \mathbf{X_2}^T \end{bmatrix}}_{2 \times 4} \boldsymbol{\pi} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

line as set of points satisfy that constraint OR as set of planes π

$$\mathbf{L}^* = \mathbf{RNS}\left(\begin{bmatrix} \mathbf{X_1}^T \\ \mathbf{X_2}^T \end{bmatrix}_{2 \times 4}\right)$$

algebraically again 1D space

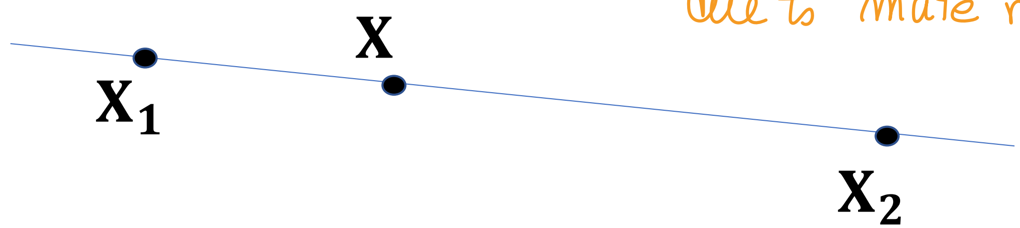
2D set of vector solutions: two planes and all their linear combinations
→ due to homogeneity: 1D set of planes (parameter: rotation angle)

5/11

linear combination of two points

Property: the point $\mathbf{X}$ given by the linear combination $\mathbf{X} = \propto \mathbf{X}_1 + \beta \mathbf{X}_2$ of two points $\mathbf{X}_1$ and $\mathbf{X}_2$ is on the line $\mathbf{L}$ through $\mathbf{X}_1$ and $\mathbf{X}_2$

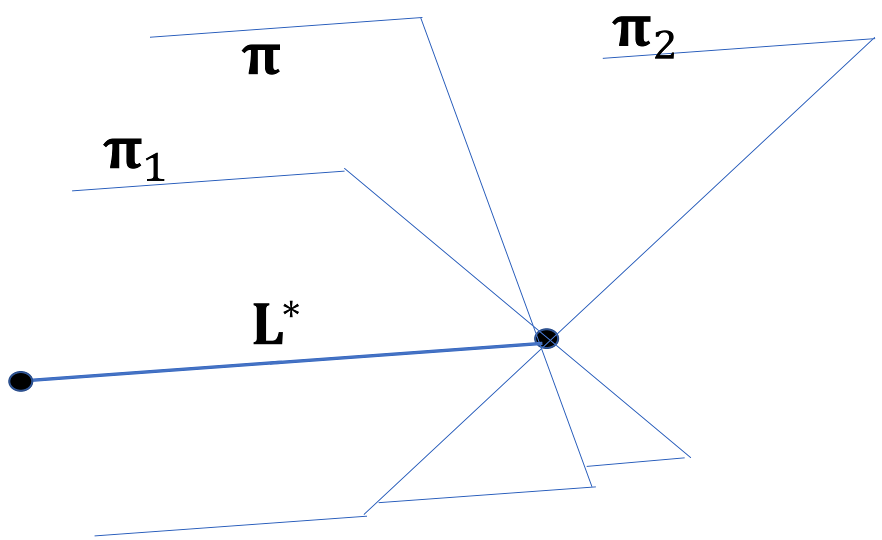
$\boxed{31/10/2023}$ ... this PART is explained in $\boxed{\text{IACV-06}}$ (2023) due to mute recording (2024)



A **line $\mathbf{L}$** can also be defined as the set of all points, that are linear combinations of two given points: $\mathbf{X} = \propto \mathbf{X}_1 + \beta \mathbf{X}_2$

DUAL: linear combination of two planes

Dual property: the plane $\boldsymbol{\pi}$**,** given by the linear combination $\boldsymbol{\pi} = \propto \boldsymbol{\pi}_1 + \beta \boldsymbol{\pi}_2$ of two planes $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$, goes through the line $\mathbf{L}^*$ on $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$

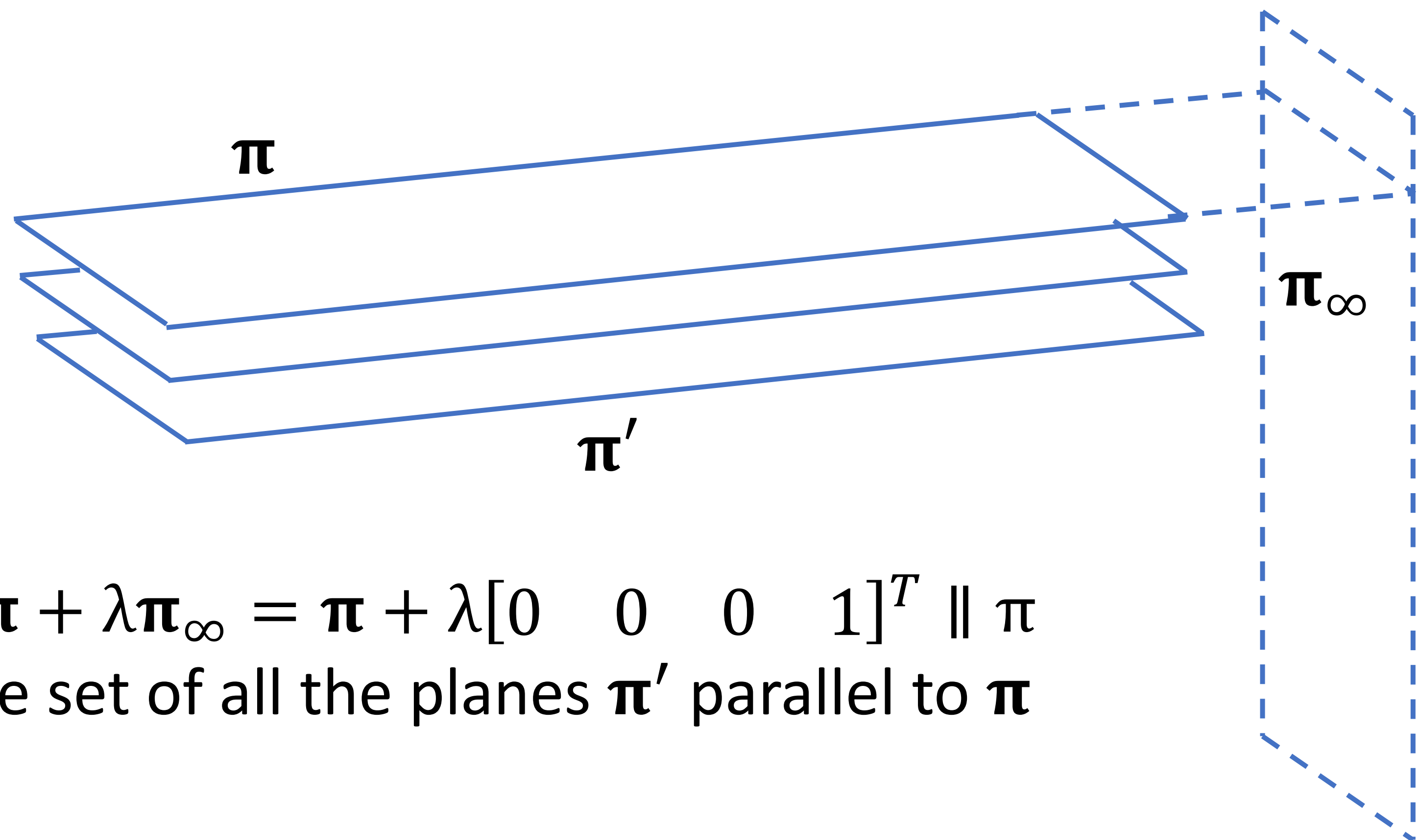


A **line** $\mathbf{L}^*$ can also be defined as the set of all planes, that are linear combinations of two given planes: $\boldsymbol{\pi} = \propto \boldsymbol{\pi}_1 + \beta\ \boldsymbol{\pi}_2$

pairs of DUALLY corresponding words

- point	→	- plane
- line	→	- line
- plane	→	- point
- is on	→	- goes through
- goes through	→	- is on

A special case:
linear combinations of a plane $\boldsymbol{\pi}$ and the plane $\boldsymbol{\pi}_{\infty}$



$$\boldsymbol{\pi}' = \boldsymbol{\pi} + \lambda\boldsymbol{\pi}_{\infty} = \boldsymbol{\pi} + \lambda[0 \quad 0 \quad 0 \quad 1]^T \parallel \pi$$

→ the set of all the planes $\boldsymbol{\pi}'$ parallel to $\boldsymbol{\pi}$

Remark: planes and lines at the infinity

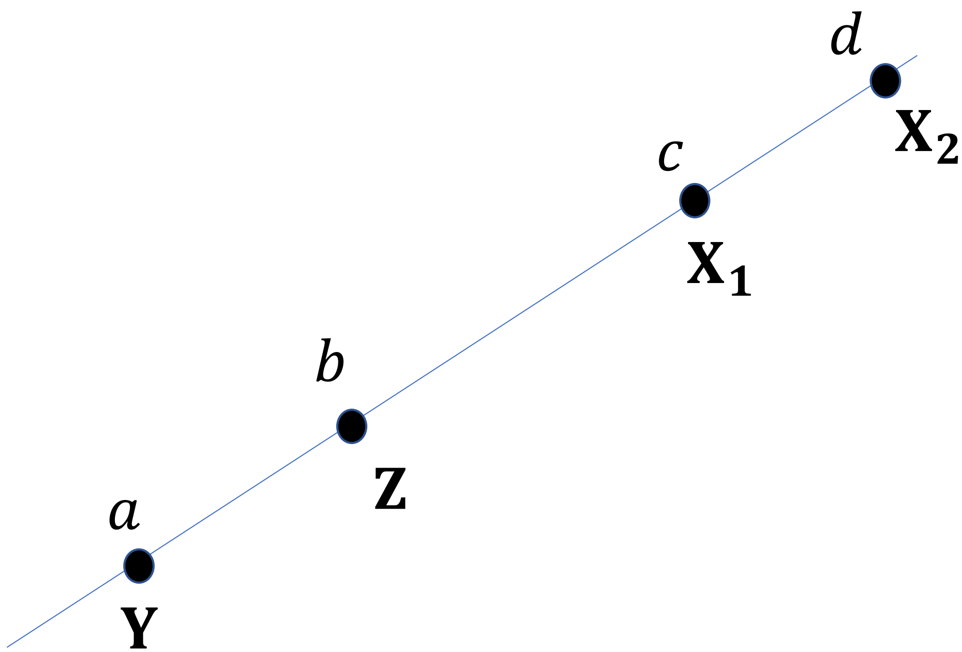
Each plane $\boldsymbol{\pi}$ has its own line at the infinity $\boldsymbol{l}_{\infty}(\boldsymbol{\pi})$
and also its own circular points $\boldsymbol{I}(\boldsymbol{\pi})$ and $\boldsymbol{J}(\boldsymbol{\pi})$

parallel planes share the same $\boldsymbol{l}_{\infty}$
and the same circular points $\boldsymbol{I}$ and $\boldsymbol{J}$

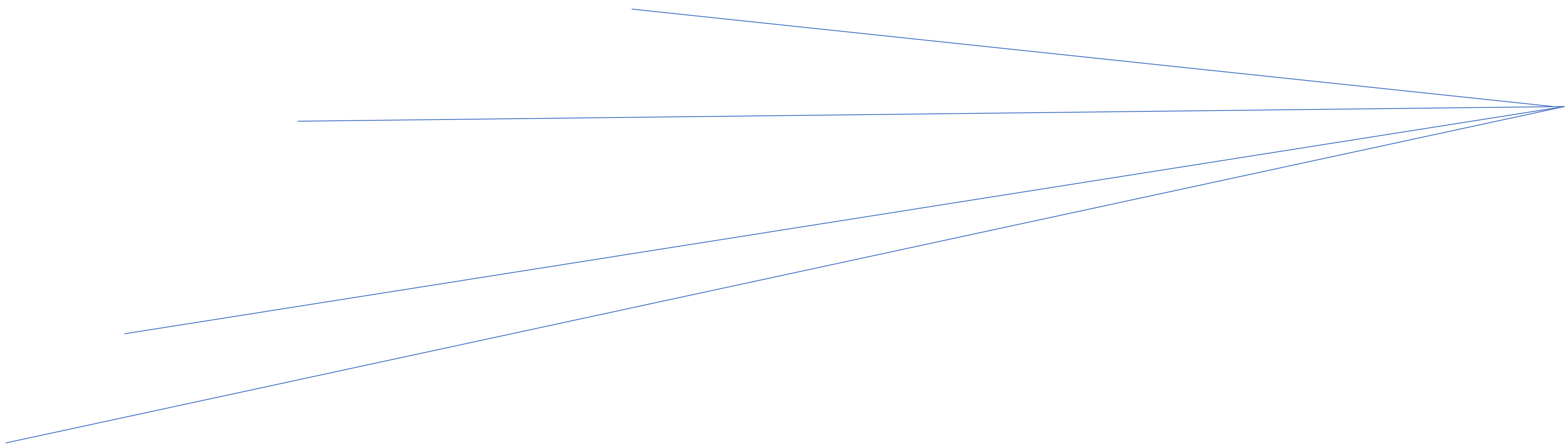
The cross ratio

1D cross ratio of a 4-tuple of colinear points

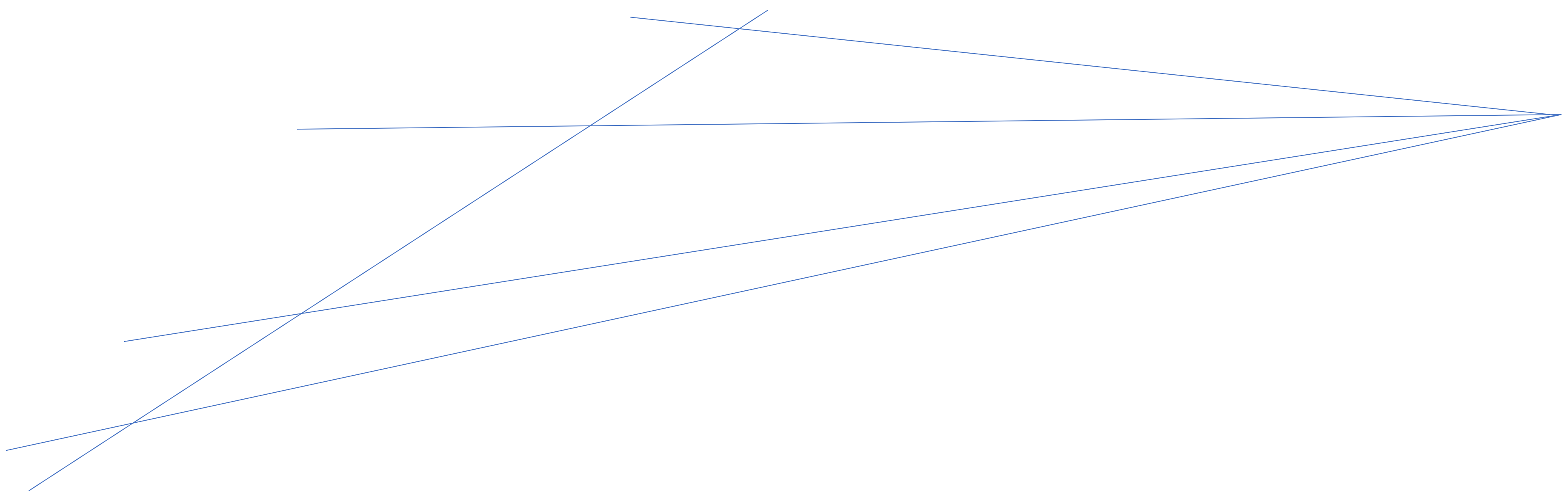
$$CR_{Y,Z,X_1,X_2} = \frac{c-a}{c-b} \Big/ \frac{d-a}{d-b}$$



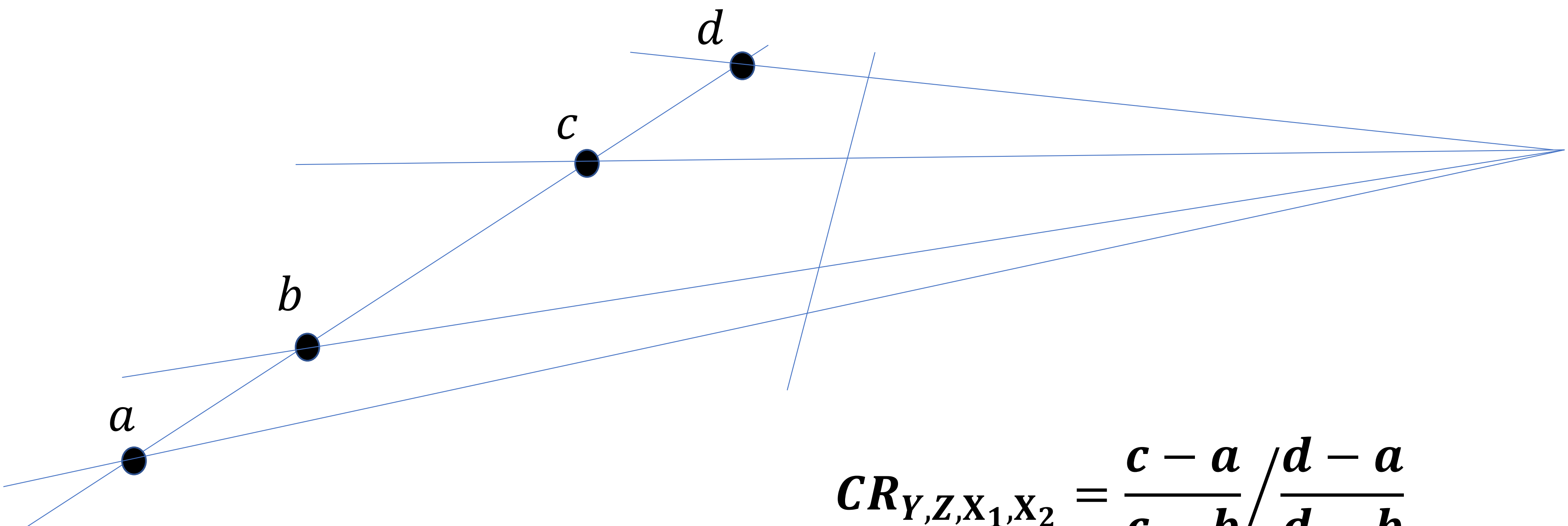
2D cross ratio of a 4-tuple of coplanar, concurrent lines



2D cross ratio of a 4-tuple of coplanar, concurrent lines: take any crossing line …



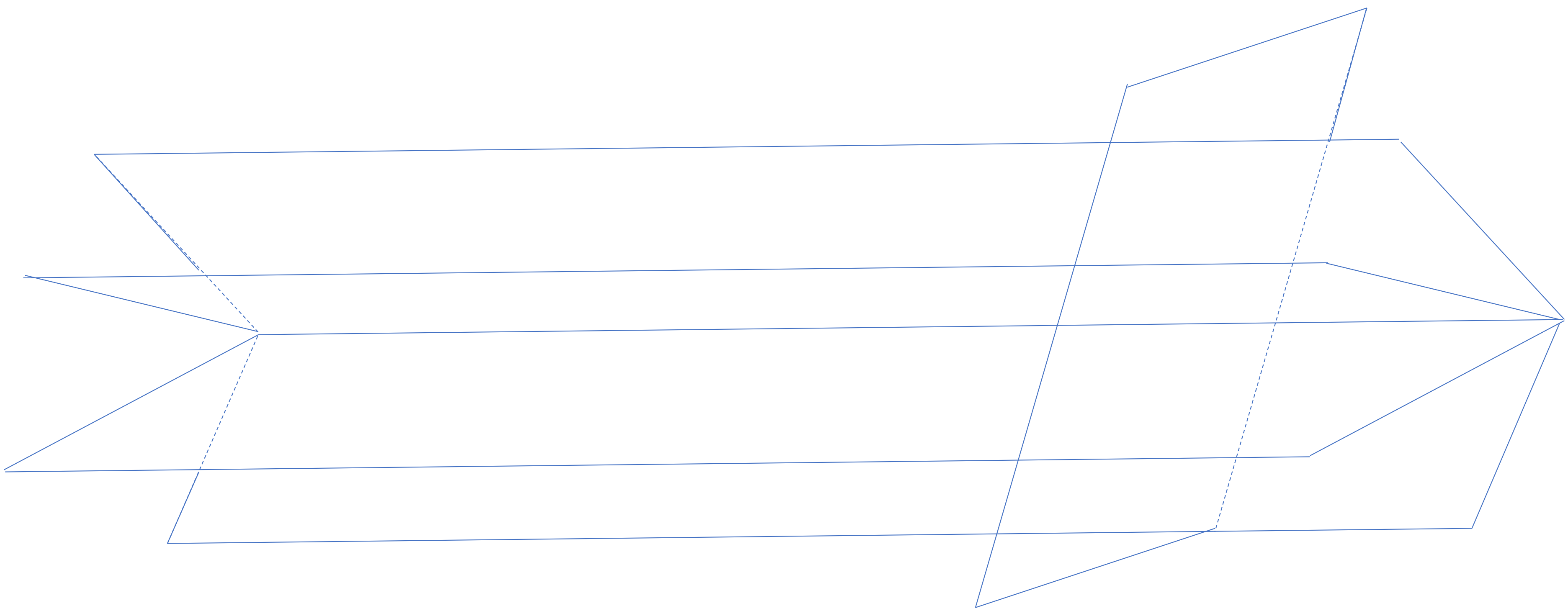
2D cross ratio of a 4-tuple of coplanar, concurrent lines: take any crossing line … compute the 1D cross ratio of intersection points



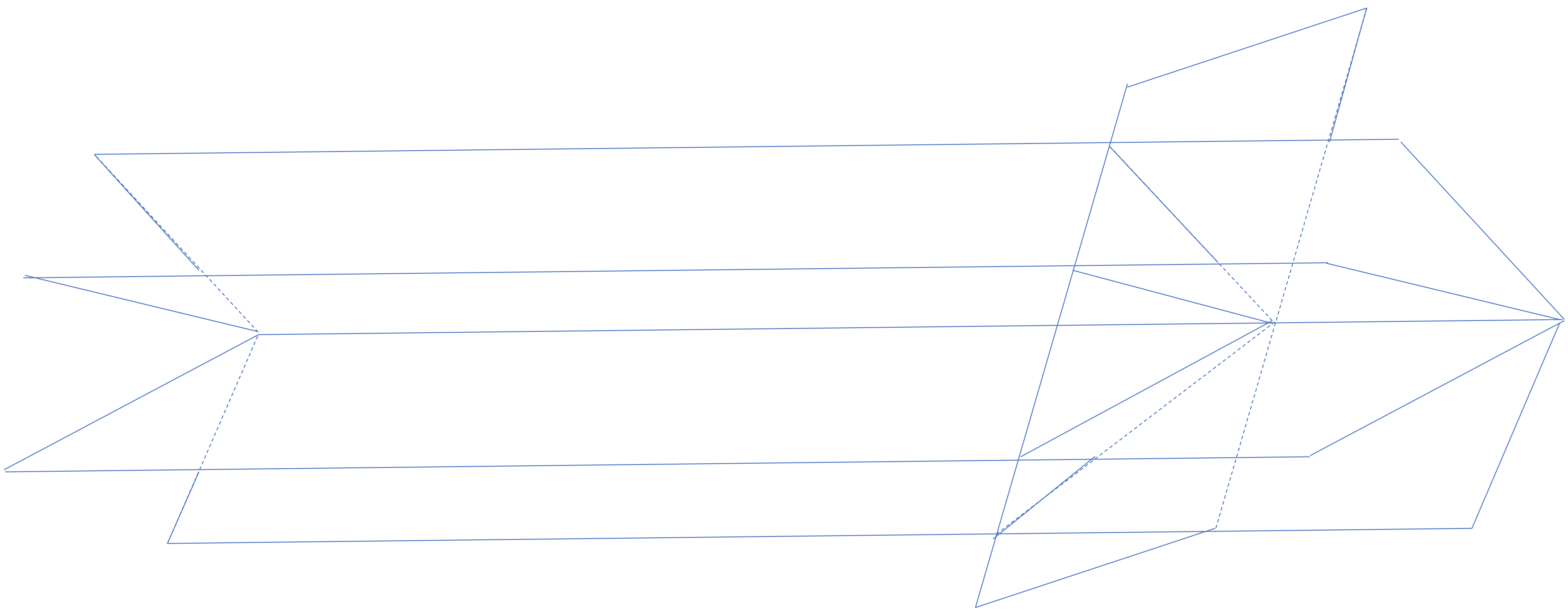
$$\boldsymbol{CR_{Y,Z,X_1,X_2}} = \frac{\boldsymbol{c-a}}{\boldsymbol{c-b}} \Big/ \frac{\boldsymbol{d-a}}{\boldsymbol{d-b}}$$

3D cross ratio of a 4-tuple of coaxial planes:

3D cross ratio of a 4-tuple of coaxial planes: take any crossing plane …

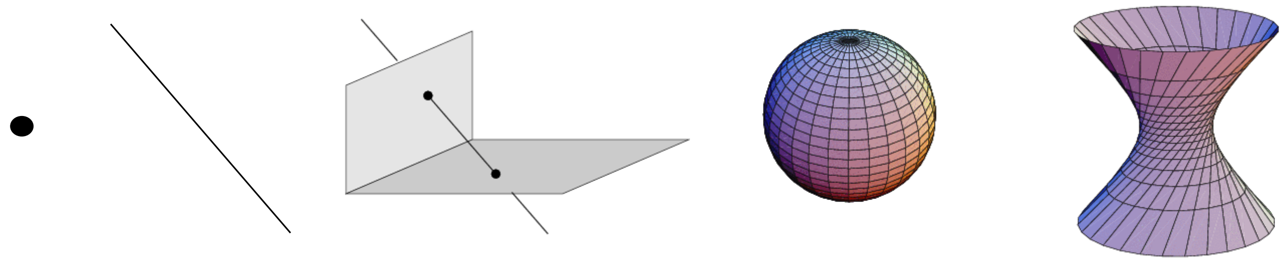


3D cross ratio of a 4-tuple of coaxial planes: take any crossing plane … compute the 2D cross ratio of intersection lines

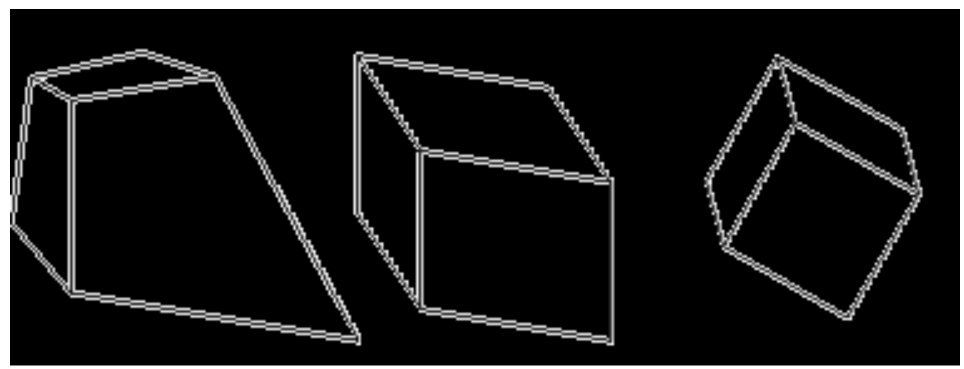


3D Space Projective Geometry

- **Elements**
  - Points
  - Planes
  - **Quadrics**
  - (Dual quadrics)



- **Transformations**
  - Isometries
  - Similarities
  - Affinities
  - Projectivities



QUADRICS

Quadric: a point $\mathbf{X}$ is on a quadric $\boldsymbol{Q}$ if it satisfies a homogeneous *quadratic* equation, namely

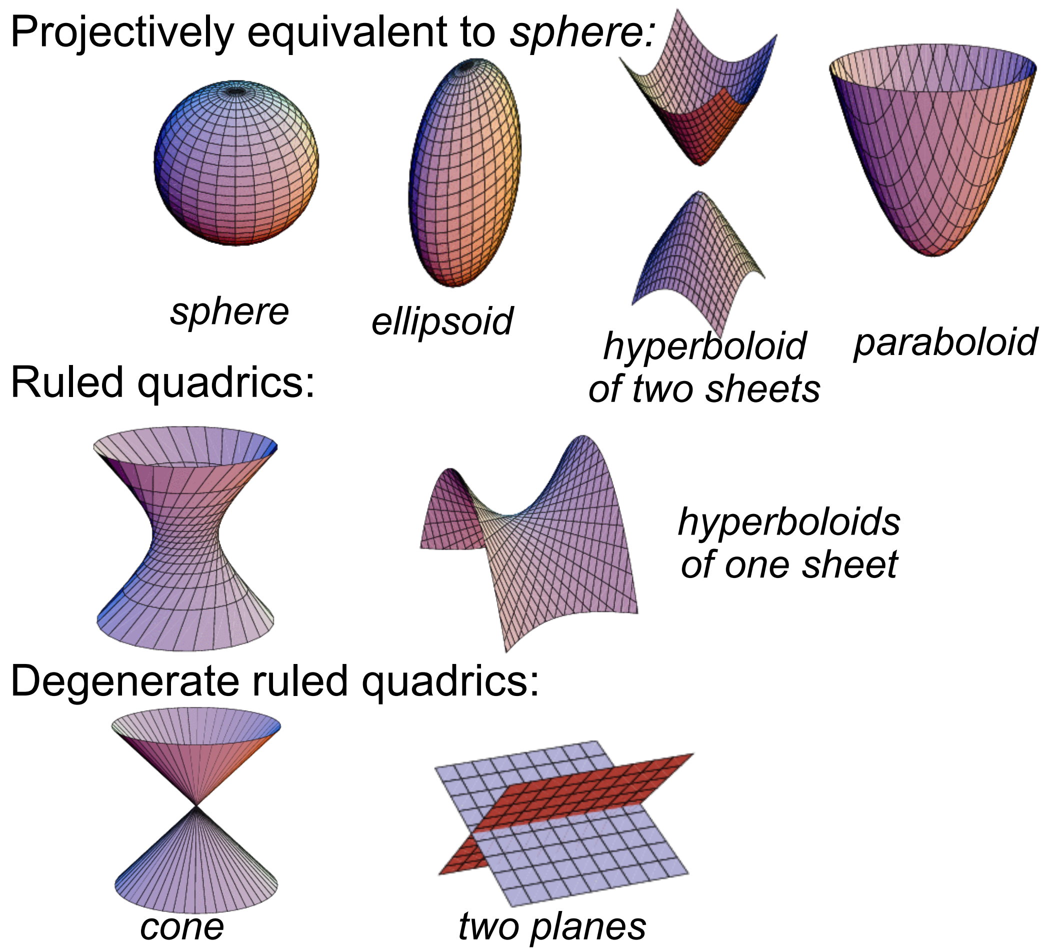
$$\mathbf{X}^T \boldsymbol{Q}\, \mathbf{X} = \mathbf{0}$$

where $\boldsymbol{Q}$ is a 4x4 symmetric matrix.

$$\mathrm{Q} = \begin{bmatrix} \bullet & \bullet & \bullet & \bullet \\ \circ & \bullet & \bullet & \bullet \\ \circ & \circ & \bullet & \bullet \\ \circ & \circ & \circ & \bullet \end{bmatrix}$$

- $\boldsymbol{Q}$ is a homogeneous matrix: $\lambda\boldsymbol{Q} \Leftrightarrow \boldsymbol{Q}$
- 9 degrees of freedom
- 9 points in general positions define a quadric

Quadric classification

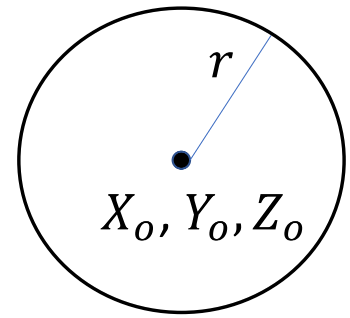


Example: the sphere

First in cartesian coordinates:

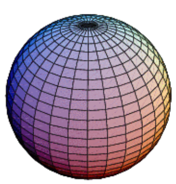
$$(X - X_o)^2 + (Y - Y_o)^2 + (Z - Z_o)^2 - r^2 = 0$$

X_o, Y_o, Z_o are the center coordinates, r is the radius.



… then in homogeneous coordinates:

$$\begin{bmatrix} x & y & z & w \end{bmatrix} \underbrace{\begin{bmatrix} 1 & 0 & 0 & -X_o \\ 0 & 1 & 0 & -Y_o \\ 0 & 0 & 1 & -Z_o \\ -X_o & -Y_o & -Z_o & X_o^2 + Y_o^2 - r^2 \end{bmatrix}}_{} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = 0$$

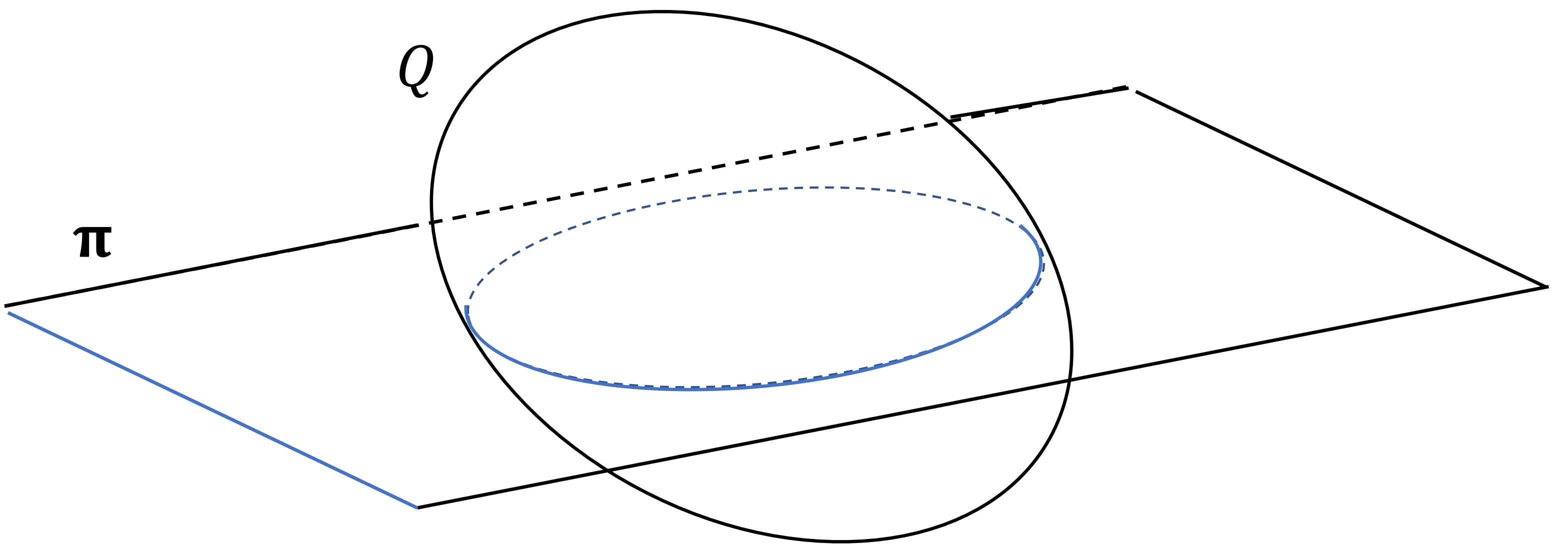


$$\mathbf{X}^\top \, \boldsymbol{Q} \, \mathbf{X} = 0$$

Intersection of a quadric and a plane

plane – quadric intersection: quadratic equation → a conic

.



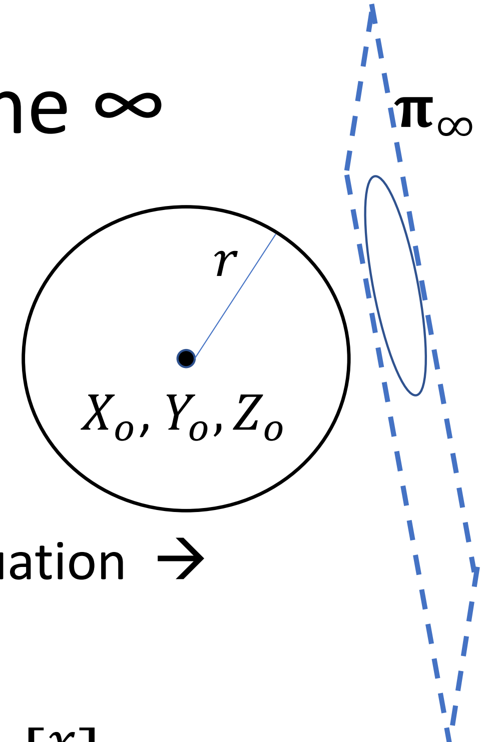
The **absolute conic**: an extension of the circular points

A noteworthy example: intersecting a sphere and the plane at the ∞

$$\begin{cases} (x - X_o w)^2 + (y - Y_o w)^2 + (z - Z_o w)^2 - r^2 w^2 = 0 \\ w = 0 \end{cases}$$

$$\rightarrow$$

$$\begin{cases} x^2 + y^2 + z^2 = 0 \\ w = 0 \end{cases}$$



The sphere parameters (center and radius) disappear from the equation $\rightarrow$ the intersection **conic** is the **same for all** spheres:

$$x^2 + y^2 + z^2 = \begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x & y & z \end{bmatrix} \Omega_\infty \begin{bmatrix} x \\ y \\ z \end{bmatrix} = 0$$

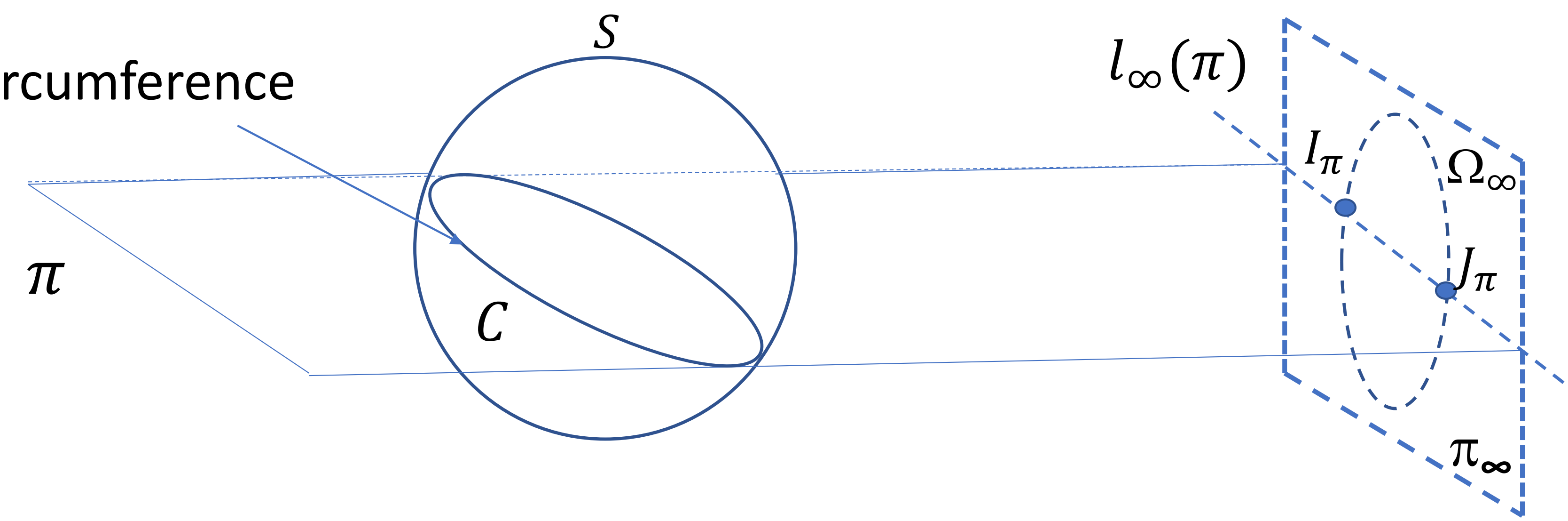
A conic within $\boldsymbol{\pi}_\infty$: $\Omega_\infty = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ **ABSOLUTE CONIC**

Property:

The absolute conic Ω_∞ contains the circular points I_π, J_π of any plane π

Proof.

Cutting sphere $\cap\ \boldsymbol{\pi}_\infty = \Omega_\infty$ with plane $\boldsymbol{\pi}$ gives circle $\cap\ \boldsymbol{l}_\infty(\boldsymbol{\pi}) = \{I_\pi, J_\pi\} \subset \Omega_\infty$



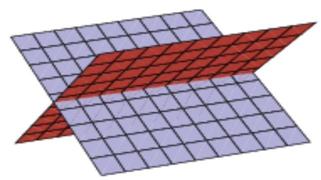
Degenerate Quadrics

DEGENERATE QUADRICS

$$\mathbf{X}^T \boldsymbol{Q}\, \mathbf{X} = \mathbf{0}$$

- rank $\boldsymbol{Q} = \mathbf{1}$ → $\boldsymbol{Q} = \mathbf{A}\mathbf{A}^T$ repeated plane $\mathbf{A}$

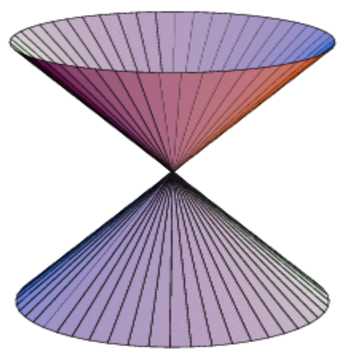
- rank $\boldsymbol{Q} = \mathbf{2}$ → $\boldsymbol{Q} = \mathbf{A}\mathbf{B}^T + \mathbf{B}\mathbf{A}^T$



two planes $\mathbf{A}$ and $\mathbf{B}$

- rank $\boldsymbol{Q} = \mathbf{3}$ a **cone**

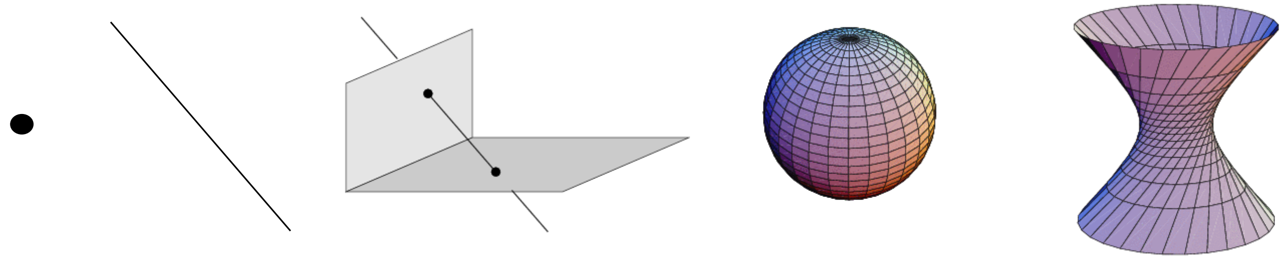
  vertex = RNS($\boldsymbol{Q}$)



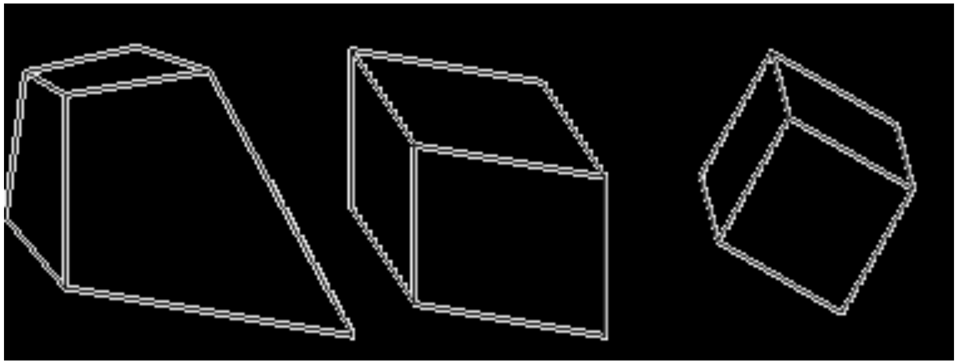
a cylinder is a cone
whose vertex is at the infinity

3D Space Projective Geometry

- **Elements**
- Points
- Planes
- **Quadrics**
- (Dual quadrics)



- **Transformations**
- Isometries
- Similarities
- Affinities
- Projectivities



Projective 3D Geometry: Projective Transformations

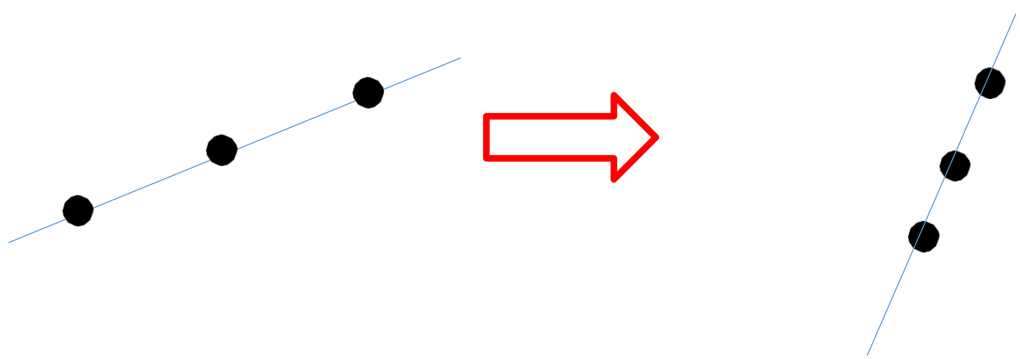
Projective mappings

Def. *A **projective mapping** between a projective space* $\mathbb{P}^3$ *and an other projective space* $\mathbb{P}'^3$ *is an **invertible** mapping which preserves colinearity:*

$$h\colon \mathbb{P}^3 \rightarrow \mathbb{P}'^3, \mathbf{X}' = h(\mathbf{X}),\ \mathbf{X}_1,\ \mathbf{X}_2, \mathbf{X}_3 \textit{ are colinear}$$

$$\leftrightarrow$$

$$\mathbf{X}'_1 = h(\mathbf{X}_1),\ \mathbf{X}'_2 = h(\mathbf{X}_2), \mathbf{X}'_3 = h(\mathbf{X}_3) \textit{ are colinear}$$



Alternative names:

- *Projectivity*

Fundamental Theorem of Projective Geometry

Theorem: *A mapping* $h : \mathbb{P}^3 \rightarrow \mathbb{P}'^3$ *is projective* if and only if *there exists an invertible 4x4 matrix* H *such that for any point in* $\mathbb{P}^3$ *represented by the vector* $\mathbf{X}$ *, is* $h(\mathbf{X}) = H\,\boldsymbol{X}$

i.e. projective mappings are LINEAR in the homogeneous coordinates

(they are not linear in cartesian coordinates)

Projectivity: 15 degrees of freedom

From the theorem

$$h(\mathbf{X}) = \mathbf{X}' = H\,\mathbf{X}$$

Therefore, if we multiply the matrix H by any nonzero scalar λ, the relation is satisfied by the same points

$$\mathbf{X}' = \lambda H\,\mathbf{X}$$

Thus any nonzero multiple of the matrix H represents the same projective mapping as H.

Hence H is a homogeneous matrix: in spite of its 16 entries,

H has only 15 degrees of freedom, namely the ratios between its elements.

Transformation of points, planes, quadrics, dual quadrics

Transformation rules for the space elements

A homography transforms **each point** $\boldsymbol{X}$ into a point $\boldsymbol{X}'$ such that:

$$\boldsymbol{X} \rightarrow H\boldsymbol{X} = \boldsymbol{X}'$$

A homography transforms **each plane $\boldsymbol{\pi}$** into a plane $\boldsymbol{\pi}'$ such that:

$$\boldsymbol{\pi} \rightarrow H^{-T}\boldsymbol{\pi} = \boldsymbol{\pi}'$$

A homography transforms **each quadric** $\boldsymbol{Q}$ into a quadric Q$'$ such that:

$$Q \rightarrow H^{-T}QH^{-1} = Q'$$

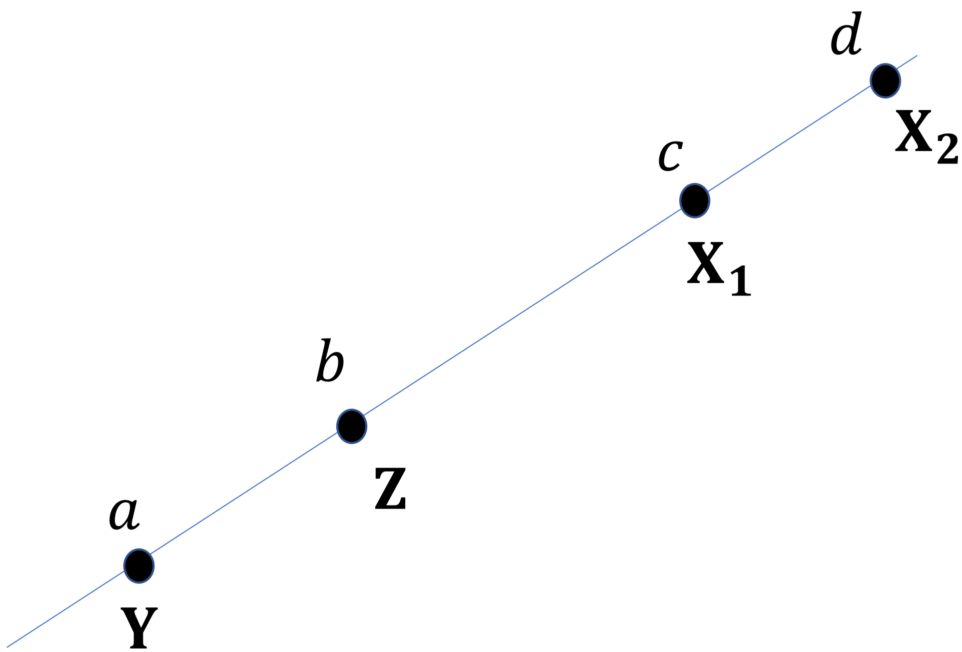
A homography transforms **each dual quadric** Q^* into a dual quadric $Q^{*\prime}$

$$Q^* \rightarrow HQ^*H^T = Q^{*\prime}$$

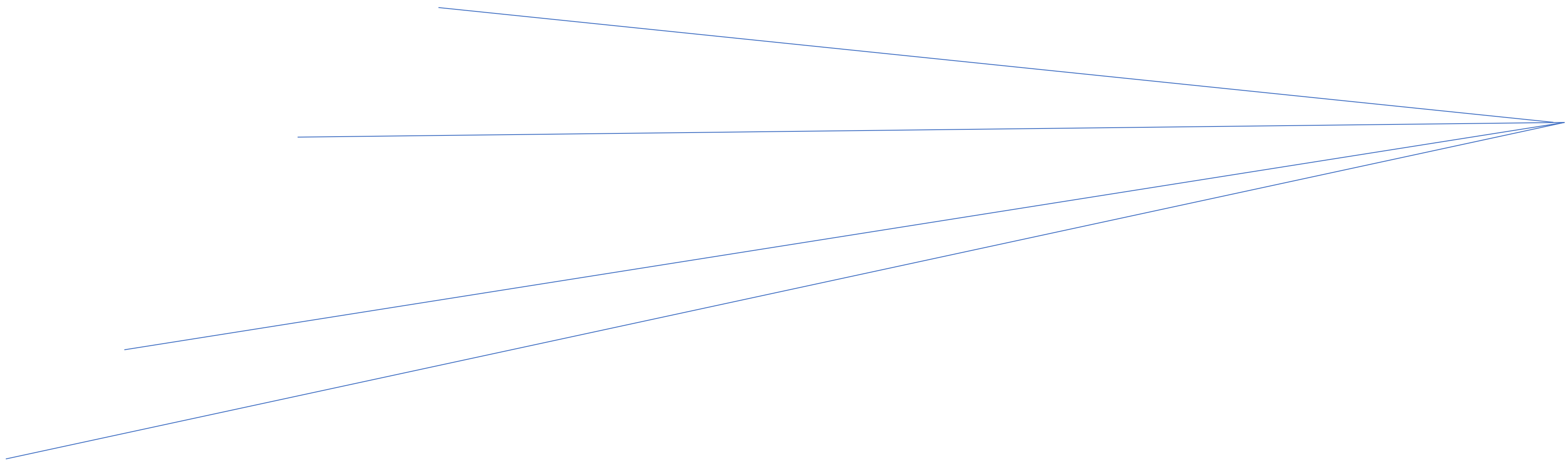
Cross ratios: invariant under projective mappings

1D cross ratio of a 4-tuple of colinear points

$$CR_{Y,Z,\mathrm{X}_1,\mathrm{X}_2} = \frac{c-a}{c-b} \Big/ \frac{d-a}{d-b}$$



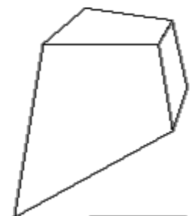
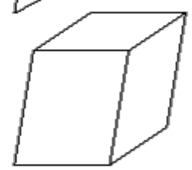
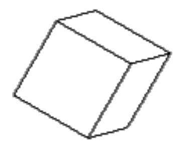
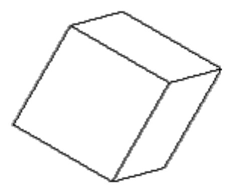
2D cross ratio of a 4-tuple of coplanar, concurrent lines

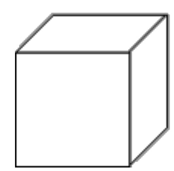


3D cross ratio of a 4-tuple of coaxial planes:

Hierarchy of projective transformations

Hierarchy of transformations

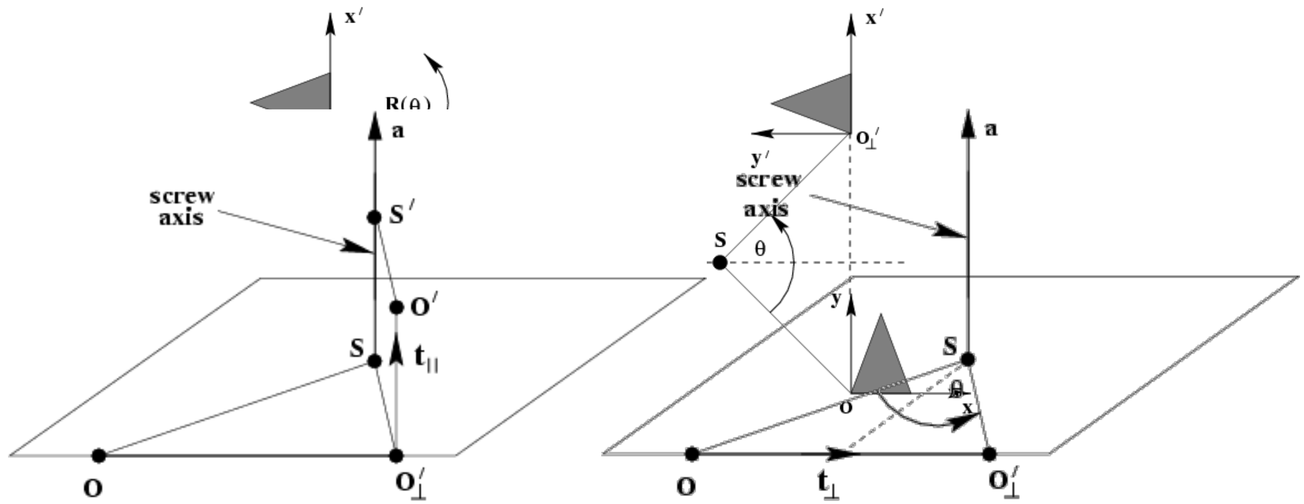
Projective 15dof	$\begin{bmatrix} A & t \\ v^T & v \end{bmatrix}$		Intersection and tangency
Affine 12dof	$\begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix}$		Parallellism of planes, Volume ratios, centroids, **The plane at infinity π_∞**
Similarity 7dof	$\begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix}$		**The absolute conic Ω_∞**
Euclidean 6dof	$\begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$		Volume



Caveat

A 3D rototranslation **is not** a pure rotation: screw decomposition

Any rotor-translation is equivalent to a rotation about a screw axis and a translation along the screw axis.



screw axis // rotation axis

$$t = t_{//} + t_{\perp}$$

Isometries (or Euclidean mappings)

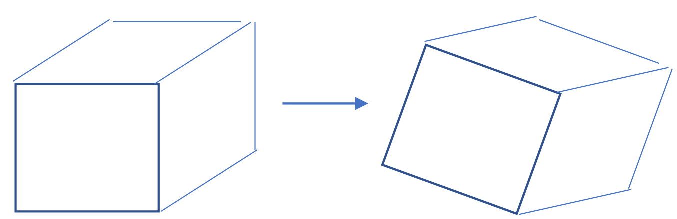
$$H_I = \begin{bmatrix} R_\perp & & & \boldsymbol{t} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$R_\perp$ is a 3x3 orthogonal matrix: ${R_\perp}^{-1} = {R_\perp}^{T}$

det ${R_\perp}^{-1} = 1$ planar rigid displacement (-1 for reflection)

6 dofs: translation $\boldsymbol{t}$ + Euler angles $\vartheta, \varphi,$ □

Invariants: lenghts, distances, areas → shape and size → relative positions



Similarities

$$H_S = \begin{bmatrix} s\,R_\perp & \boldsymbol{t} \\ 0 \quad 0 & 1 \end{bmatrix}$$

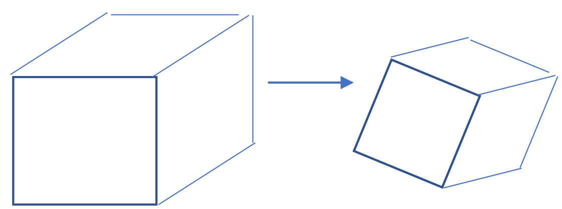
$R_\perp$ is a 3x3 orthogonal matrix: $R_\perp{}^{-1} = R_\perp{}^{T}$

7 dofs: rigid displacement + *scale*

Invariants: ratio of lenghts, angles → shape (not size)

<u>the absolute conic</u> Ω_∞

and the absolute dual quadric $\boldsymbol{Q}^*{}_\infty$



Affinities (or affine mappings)

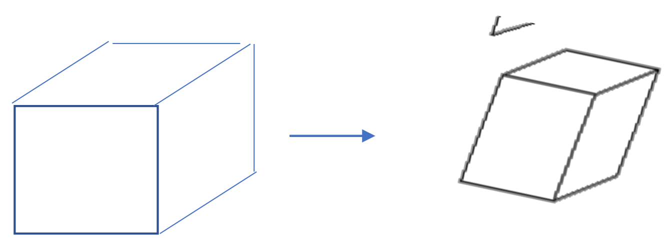
$$H_A = \begin{bmatrix} & A & & \boldsymbol{t} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A is any 3x3 invertible matrix

12 dofs: A + $\boldsymbol{t}$

Invariants: parallelism, ratio of parallel lenghts, ratio of areas

<u>the plane at the infinity</u> $\boldsymbol{\pi}_\infty$



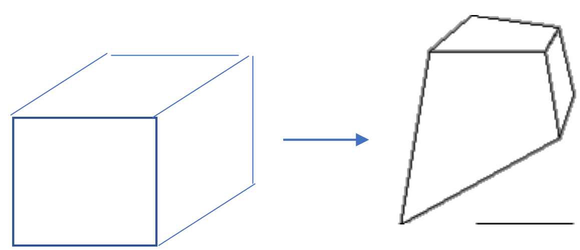
Projectivities (or projective mappings)

$$H = \begin{bmatrix} A & \boldsymbol{t} \\ \boldsymbol{v}^T & 1 \end{bmatrix}$$

A is any 3x3 invertible matrix

15 dofs: A + $\boldsymbol{v}$ + $\boldsymbol{t}$

Invariants: colinearity, incidence, order of contact (crossing, tangency, inflections), <u>the 1D cross ratio, the 2D cross ratio, the 3D cross ratio</u>



3D reconstruction problem formulation

3D reconstruction problem formulation

Unknown original scene = set of points in the 3D space

→

An unknown 3D projective mapping is applied to them

→

Suppose that the transformed **3D points** can be observed

→

From the observed points (different from the original)

recover a model of the original scene

HOW? Images are 2D, not 3D: **several views** (see later)

3D Shape reconstruction:

will be studied in Multi-view Geometry

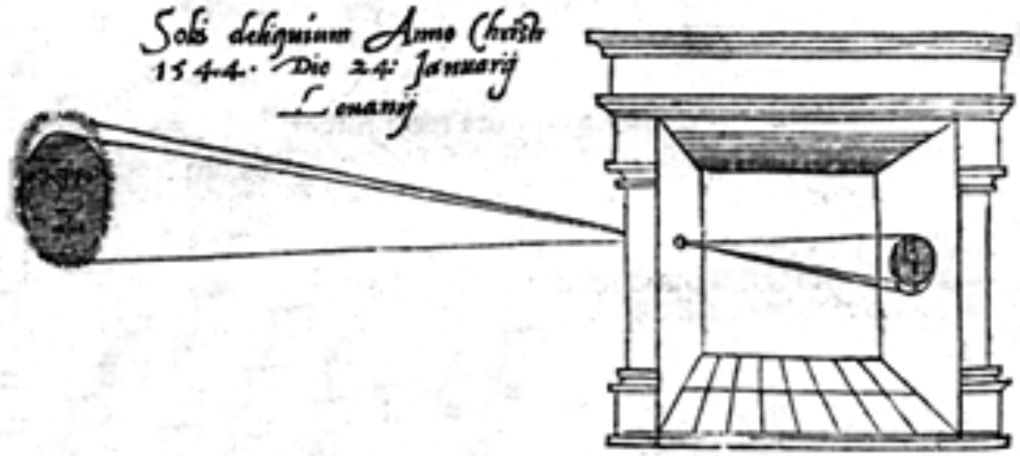
- Introduction and the Camera Optical System
- Planar (2D) Projective Geometry
- Spatial (3D) Projective Geometry
- **Camera Geometry (3D → 2D Projection) and single-view Geometry**

Camera Geometry

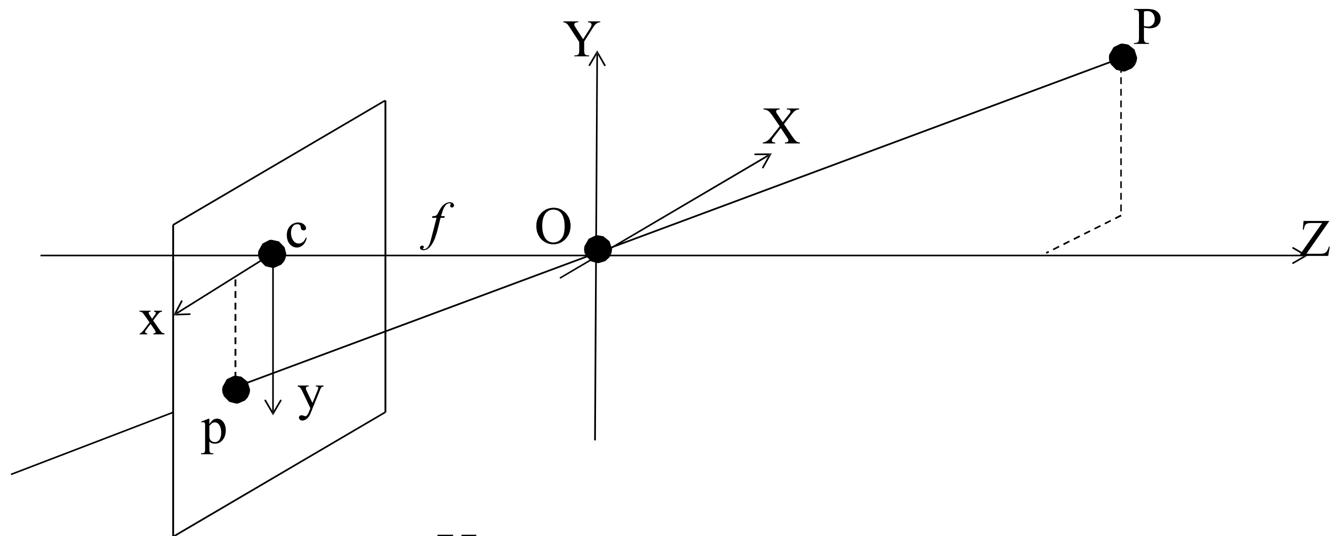
camera projective model

Pinhole camera

illum in tabula per radios Solis, quàm in cœlo contin-
git: hoc eſt, ſi in cœlo ſuperior pars deliquiũ patiatur, in
radiis apparebit inferior deficere, vt ratio exigit optica.



Sic nos exactè Anno .1544. Louanii eclipſim Solis
obſeruauimus, inuenimuſq; deficere paulò plus q̃ dex-



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

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

perspective projection

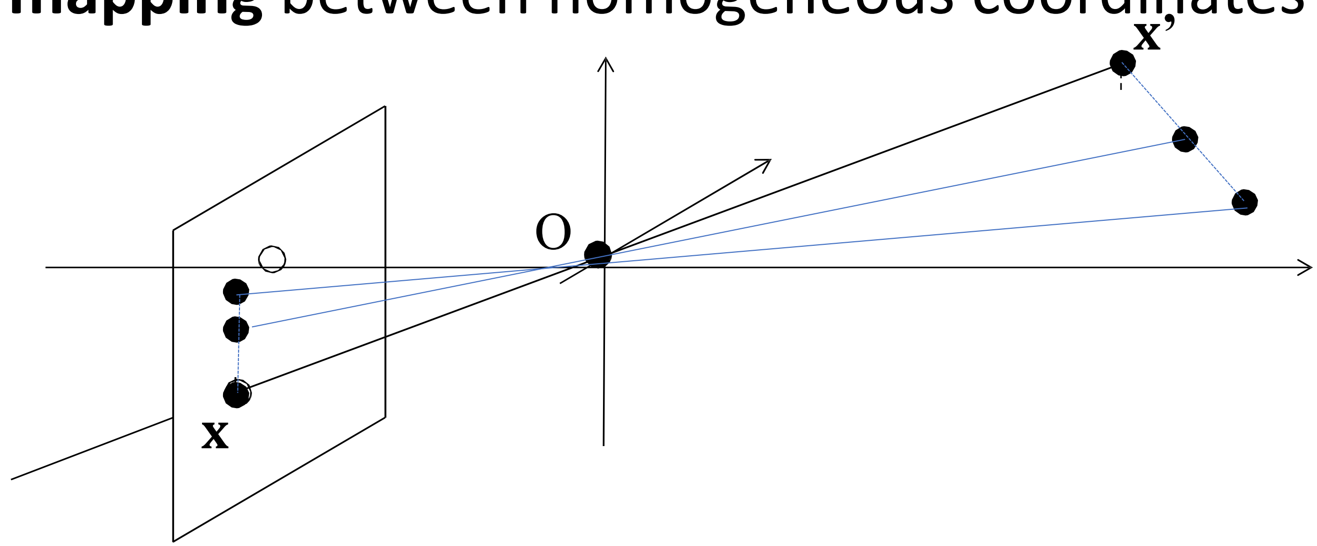
-nonlinear
-not shape-preserving
-not length-ratio preserving

Scene-to-image projection

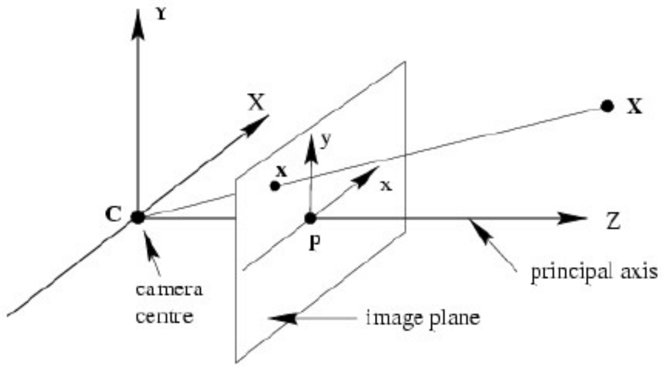
Colinear points are projected onto colinear image points

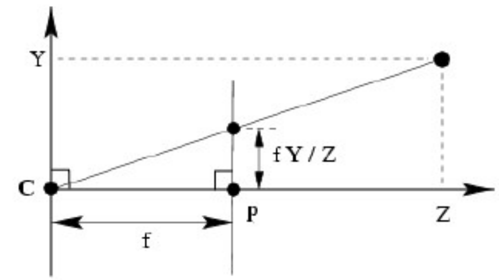
→ colinearity is preserved

→ **linear mapping** between homogeneous coordinates



CAMERA GEOMETRY





colinearity is preserved → linear relation among homogeneous coords

camera projection matrix

$$\mathbf{X} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \mathbf{u} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \mathbf{P}_{3x4} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \mathbf{P}_{3x4}\mathbf{X} = \left| \mathbf{M}_{3x3} \quad \mathbf{m}_{3x1} \right| \mathbf{X}$$

3D space

image

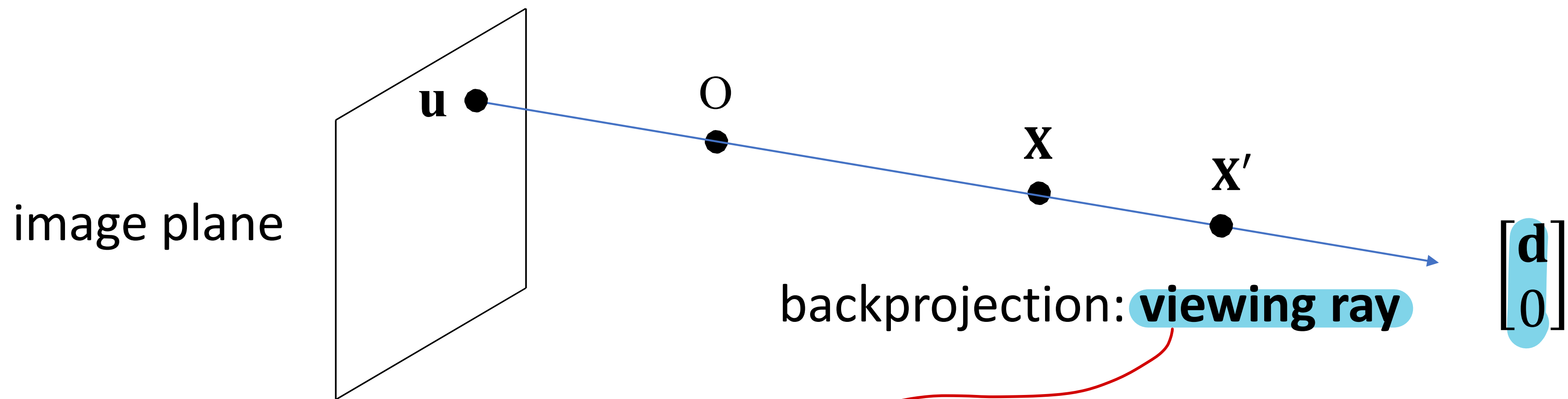
invertible

- **SCENE**
- CAMERA

14/11/2023

10 min

The viewing ray associated to an image point $\mathbf{u}$



The backprojection of image point $\mathbf{u}$ for the camera $\mathrm{P} = [\mathbf{M} \quad \mathbf{m}]$, is a straight line

- through $\underbrace{\mathrm{O} = \ RNS(\mathrm{P})}^{\text{easy to find}}$ *Proof*: any $\mathbf{X}' = \mathrm{O} + \lambda\mathbf{X}$ projects to $\mathrm{P}\mathbf{X}' = \mathrm{PO} + \mathrm{P}\mathbf{X} = \mathrm{P}\mathbf{X}$ (with $\mathrm{PO} = 0$)
- whose direction is $\mathbf{d} = \mathbf{M}^{-1}\mathbf{u}$ *Proof*: $\mathbf{u} = [\mathbf{M} \quad \mathbf{m}]\begin{bmatrix}\mathbf{d}\\0\end{bmatrix} = \mathbf{Md} \rightarrow \mathbf{d} = \mathbf{M}^{-1}\mathbf{u}$

Where is O ? from $\mathrm{PO} = \underbrace{[\mathbf{M} \quad \mathbf{m}]}^{\text{PROJ MATRIX}}\begin{bmatrix}\mathbf{o}\\1\end{bmatrix} = 0 \ \rightarrow \ \mathbf{o} = \ -\mathbf{M}^{-1}\mathbf{m}$ cartesian coordinates

- SCENE
- **CAMERA**

we started to understand camera behavior...

an we found the projection matrix P that relate each x in space to u in image plane

ARRIVE HERE 5/11

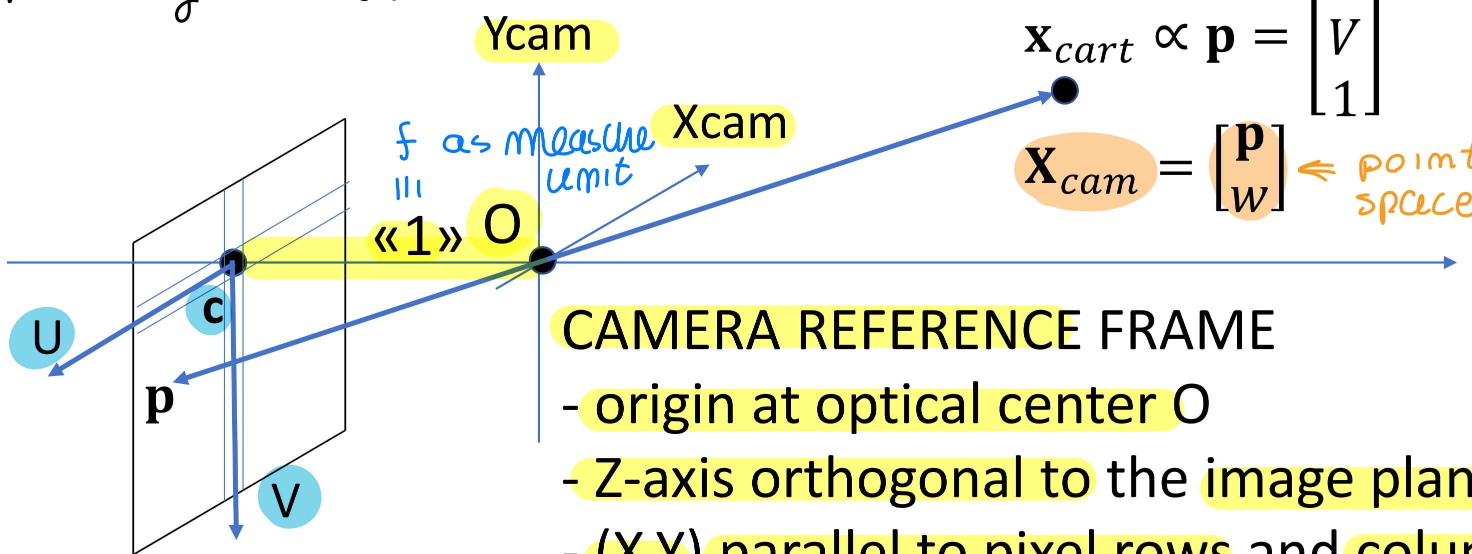
19/11/2024

See from 2023 lecture (Mute ~~?~~) in 2024)

and many part of this matrix helps to derive important features for example the <u>viewing ray</u>, (set of points projecting on image point)

homog. **geometric** coordinates of image point: $\mathbf{p} = [U, V, 1]^T$

looking at interior of camera:



$$\mathbf{x}_{cart} \propto \mathbf{p} = \begin{bmatrix} U \\ V \\ 1 \end{bmatrix}$$

$$\mathbf{X}_{cam} = \begin{bmatrix} \mathbf{p} \\ w \end{bmatrix}$$ ← point in 3D space, cam reference

CAMERA REFERENCE FRAME

- origin at optical center O
- Z-axis orthogonal to the image plane
- (X,Y) parallel to pixel rows and columns
- measure unit along X,Y and Z = focal distance f

- IMAGE **GEOMETRIC** COORDINATE SYSTEM
  - origin at the principal point C ≡ orthogonal projection on img plane of camera center
  - (U, V) parallel to pixel rows and columns, but opposite to (X, Y)
    ↑ opposite direction
  - measure unit along U and V = focal distance f
    (same measure unit of CAMERA Ref)

homog. **geometric** coordinates of image point: $\mathbf{p} = [U, V, 1]^T$

cartesian coord are proportional to point p... they are multiple of [U V 1]

← since X // U, Y // V

CARTESIAN COORD

$$\mathbf{x}_{cart} \propto \mathbf{p} = \begin{bmatrix} U \\ V \\ 1 \end{bmatrix}$$

$$\mathbf{X}_{cam} = \begin{bmatrix} \mathbf{p} \\ w \end{bmatrix}$$

in 3D cartesian coord

HOMOGENEUS coord, adding w

being parallel X,Y with U,V axis, the coordinates are same component in camera and image

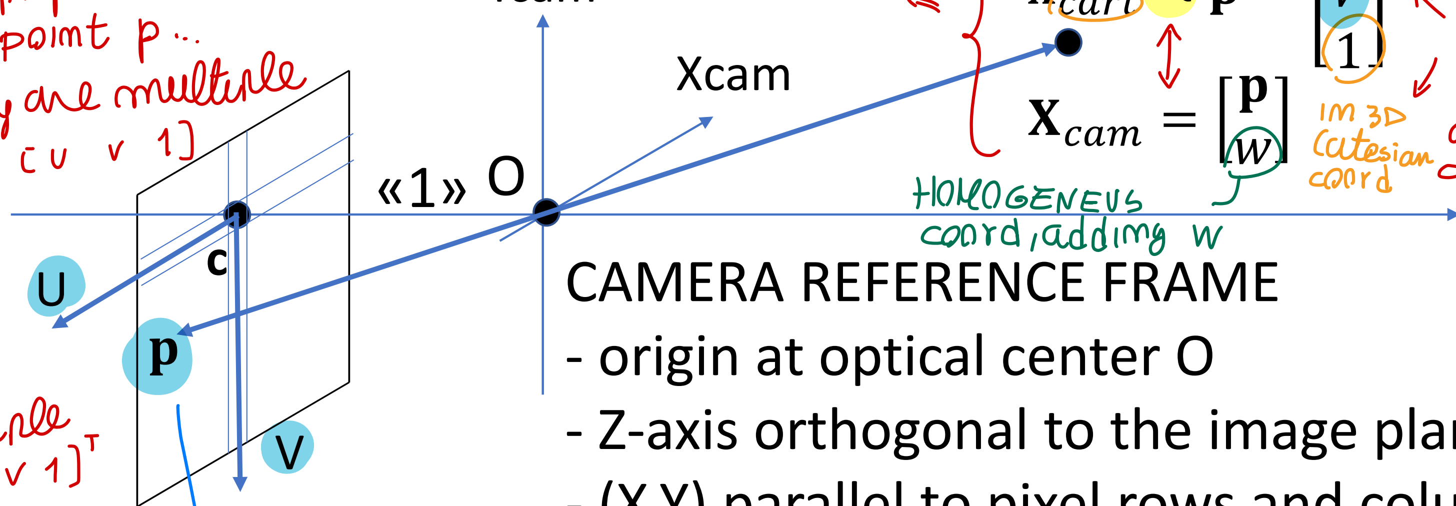


image point

are a multiple of $[U\ V\ 1]^T$ ↓ $x_{cart} \propto p$

CAMERA REFERENCE FRAME

- origin at optical center O
- Z-axis orthogonal to the image plane
- (X,Y) parallel to pixel rows and columns
- measure unit along X,Y and Z = focal distance f

IMAGE **GEOMETRIC** COORDINATE SYSTEM → comfortable system for computations, NOT for practical application

- origin at the principal point C
- (U, V) parallel to pixel rows and columns, but opposite to (X, Y)
- measure unit along U and V = focal distance f

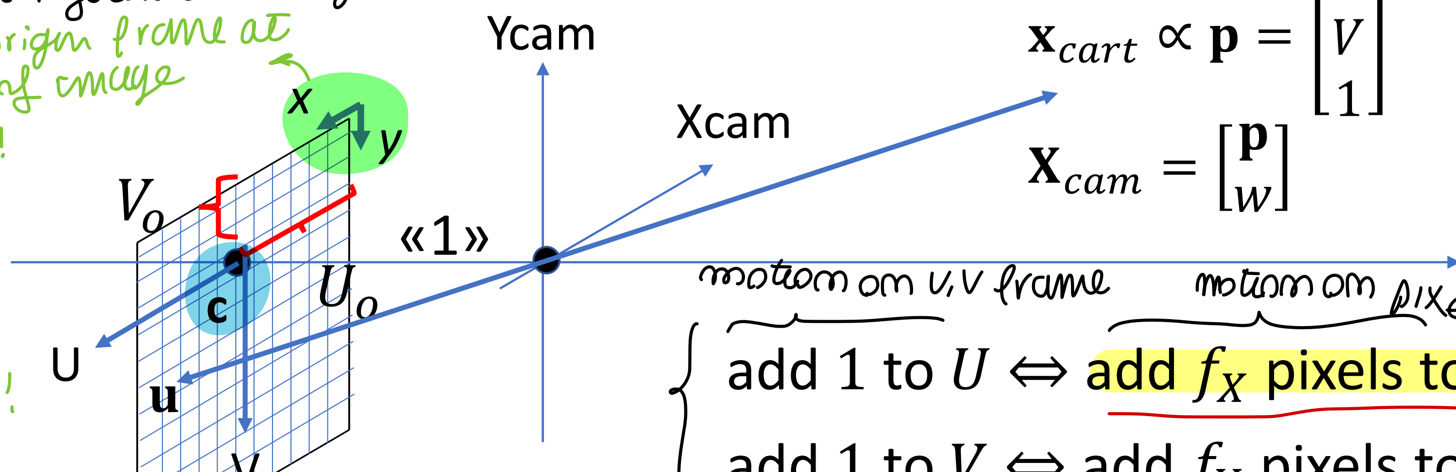
homogeneous **pixel** coordinates of image point: $\mathbf{u} = [x, y, 1]^T$

↓ consider reference change

Pixel origin frame at vertex of image plane!

↓

coordinates in pixels are bigger for same using pixelunits!



$$\mathbf{x}_{cart} \propto \mathbf{p} = \begin{bmatrix} U \\ V \\ 1 \end{bmatrix}$$

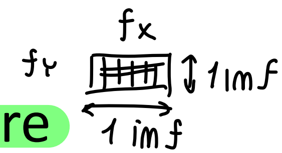
$$\mathbf{X}_{cam} = \begin{bmatrix} \mathbf{p} \\ w \end{bmatrix}$$

$$\begin{cases} \overbrace{\text{add 1 to } U}^{\text{motion on } U,V \text{ frame}} \Leftrightarrow \overbrace{\text{add } f_X \text{ pixels to } x}^{\text{motion on pixel frame}} \\ \text{add 1 to } V \Leftrightarrow \text{add } f_Y \text{ pixels to } y \end{cases} \text{ rescale}$$

pixel coordinates of C : (U_o, V_o) ← translate

⇓

pixels: not (always) square



$$\mathbf{u} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & U_o \\ 0 & f_y & V_o \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} U \\ V \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & 0 & U_o \\ 0 & f_y & V_o \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \mathbf{p} = [\mathbf{K} \quad \mathbf{0}]\mathbf{X}_{cam} = \mathbf{K}[\mathbf{I} \quad \mathbf{0}]\mathbf{X}_{cam}$$

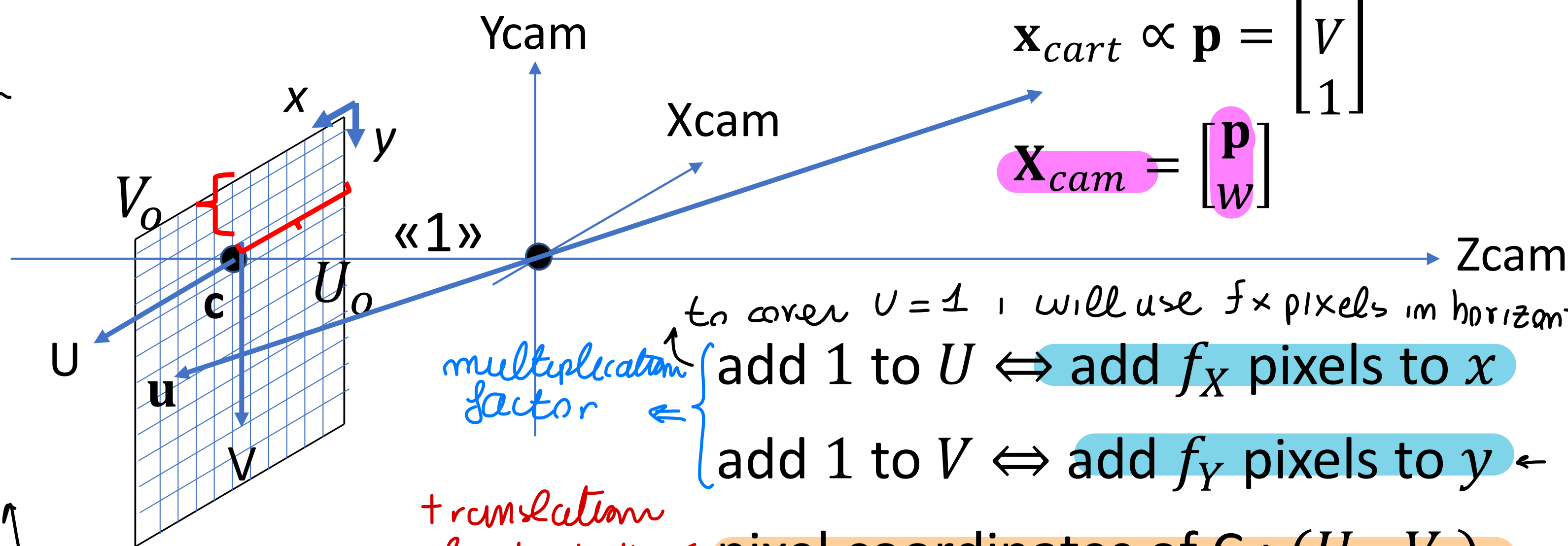
K : called **calibration matrix**

homogeneous **pixel** coordinates of image point: $\mathbf{u} = [x, y, 1]^T$

$\mathbf{x}_{cart} \propto \mathbf{p} = \begin{bmatrix} U \\ V \\ 1 \end{bmatrix}$

$\mathbf{X}_{cam} = \begin{bmatrix} \mathbf{p} \\ w \end{bmatrix}$

the estimation of this numbers concern camera calibration problem!



to cover $U = 1$, will use f_x pixels in horizontal row

multiplication factor

add 1 to U ⇔ add f_X pixels to x

add 1 to V ⇔ add f_Y pixels to y

vertical pixels f_Y to cover $V = 1$

translation factor between pixel and plane frame

pixel coordinates of C : (U_o, V_o)

pixels: not (always) square

(multiply w by 0)

camera reference coordinate, in 3D space, the original homogeneous coord.

$$\mathbf{u} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & U_o \\ 0 & f_y & V_o \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} U \\ V \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & 0 & U_o \\ 0 & f_y & V_o \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \mathbf{p} = [\mathbf{K} \quad \mathbf{0}]\mathbf{X}_{cam} = \mathbf{K}[\mathbf{I} \quad \mathbf{0}]\mathbf{X}_{cam}$$

image plane coordinates

collect camera features!

f_x / f_Y := aspect ratio of pixel (its shape)

etc...

K: called **calibration matrix**

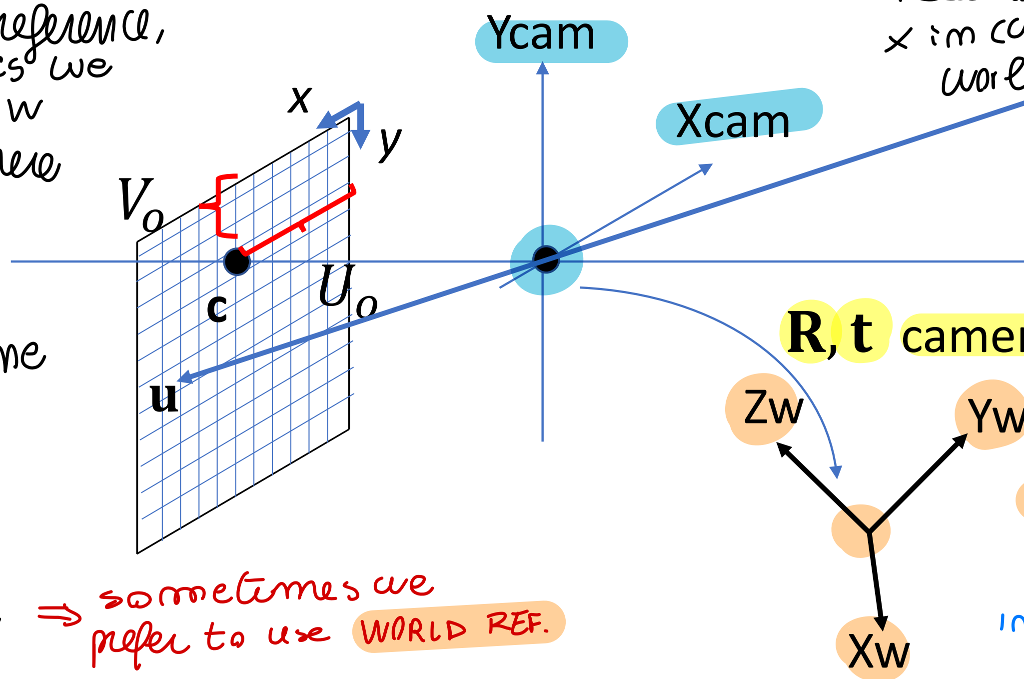
In general, world reference is $\neq$ camera reference

we don't always use the camera reference, sometimes we don't know exactly where is the camera wrt respect world frame
↓
You are NOT sure about x, y, z cam axis usually!

⇒ sometimes we prefer to use WORLD REF.

relationship of x in cam and world reference

⇐ represent how to go from world to camera



R, t camera → world: i.e., (world wrt camera)

≠ camera usually

world reference

$$\mathbf{X}_{cam} = \underbrace{\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}}_{\text{ISOMETRY}} \mathbf{X}_{world}$$

in camera reference

in world reference

rototranslation

$$\mathbf{u} = \mathbf{K}[\mathbf{I} \quad \mathbf{0}]\mathbf{X}_{cam} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]\ \mathbf{X}_{world} = \boxed{[\mathbf{KR} \quad \mathbf{Kt}]}\ \mathbf{X}_{world}$$

last row is multiplied by 0

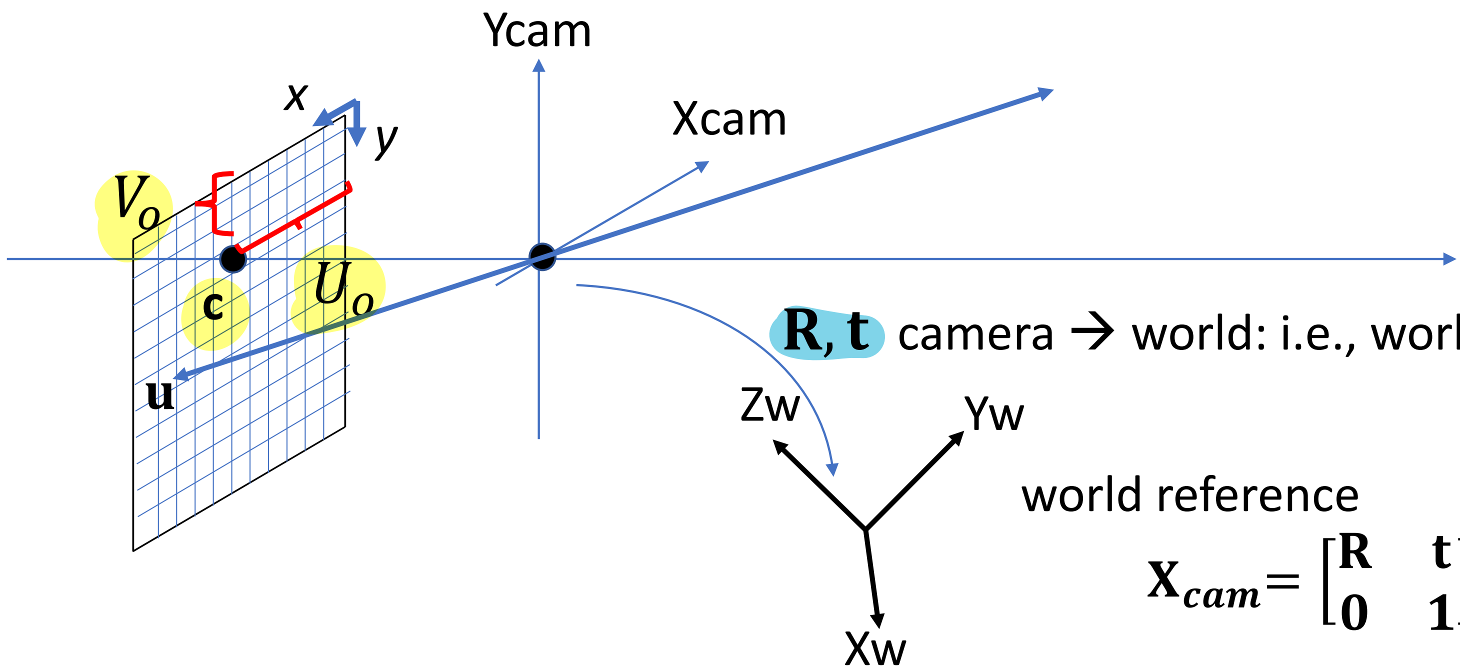
using the relationship found before...

remember $\mathbf{u}$ = $\quad = \quad \mathbf{P} \quad \mathbf{X}_{world} = \boxed{[\mathbf{M} \quad \mathbf{m}]}\ \mathbf{X}_{world}$

camera to world rotation

→ $\boxed{\mathbf{M} = \mathbf{KR}}$ and $\boxed{\mathbf{m} = \mathbf{Kt}}$

In general, world reference is ≠ camera reference



$$\mathbf{X}_{cam} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \mathbf{X}_{world}$$

$$\mathbf{u} = \mathbf{K}[\mathbf{I} \quad \mathbf{0}]\mathbf{X}_{cam} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]\ \mathbf{X}_{world} = [\mathbf{KR} \quad \mathbf{Kt}]\ \mathbf{X}_{world}$$

remember $\mathbf{u} =$ $\quad = \mathbf{P} \quad \mathbf{X}_{world} = [\mathbf{M} \quad \mathbf{m}]\ \mathbf{X}_{world}$

EXTRINSIC

INTRINSIC * ROTATION $\qquad$ INTRINSIC * TRANSLATION

EXTRINSIC

interpretation of projection matrix!

$\rightarrow$ $\boxed{\mathbf{M} = \mathbf{KR}}$ and $\boxed{\mathbf{m} = \mathbf{Kt}}$

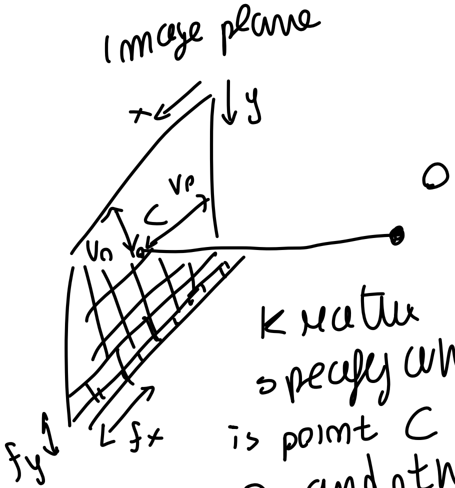
Calibration matrix

Calibration matrix

$$\begin{cases} M = K\,R \\ m = K\,t \end{cases}$$

INTRINSIC ≡ how camera is made

EXTRINSIC ≡ where is the camera w.r.t. world ← moving camera around this change!



K matrix specify where is point C respect O, and other stuff like f_x/f_y

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & U_o \\ 0 & f_y & V_o \\ 0 & 0 & 1 \end{bmatrix}$$

Intrinsic parameters: don't vary under camera displacement

depends on how the camera is made!

account for ⇓

- relative position of camera center and image plane
- pixel aspect ratio $a = \frac{f_x}{f_y}$ (for ***natural camera***, square pixels, is $a = 1$)

≃ special class of cameras

Camera projection matrix

$$\boldsymbol{P} = [\mathbf{KR} \quad \mathbf{Kt}] = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]$$

Intrinsic parameters + **Extrinsic parameters:**

- relative pose (= position & orientation) of camera reference and world reference

Camera calibration

↳ process of
estimating
matrix K of
intrinsic parameter ⇒

when fx=fy → square pixel ⇒ you define just f unique focal distance

fy

fx

this numbers define the focal distance by fx, fy

IN new camera models, this term is always 0

by calibration you have much more informations about your camera

Intrinsic camera **calibration**:

estimation of matrix $\mathbf{K}$

- focal distance f_X
- focal distance f_Y
- principal point (U_o, V_o)
- skew factor (in old cameras)

aspect ratio $a = f_X / f_Y$:

ratio between pixel width and height

intrinsic camera parameters don't vary under camera displacement

In old cameras

$K = \begin{bmatrix} f_X & & 0 \\ & f_Y & \\ & & \ddots \end{bmatrix}$

there was a SKEW FACTOR accounting video signal by sampling process (no digital)

⇓

take account of sampling period of pixels...

fx, fy NOT perfectly perpendicular!

Extrinsic camera **calibration**:
estimation of matrix $\mathbf{R}$ and vector $\boldsymbol{t}$

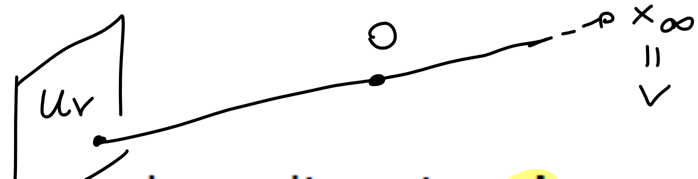
from camera to world relation

- camera→world rotation $\mathbf{R}$
- camera→world translation $\boldsymbol{t}$

extrinsic camera parameters vary under camera displacement

→ an important theorem is valid! →

vanishing points



V image of the point at the ∞ along direction **d**

to recover direction of the viewing ray on original world coordinate of uv point vanishing.

image of point at infinity (VANISHING POINT)

$$u_V = \Big[\underbrace{\overset{KR}{M} \,\Big|\, m}_{P}\Big] \cdot \begin{bmatrix} d \\ \hdashline 0 \end{bmatrix} = \underbrace{M \cdot d}_{[KR]d}$$

P projection matrix

V, point at infinity

the direction of viewing ray d considered

$d = M^{-1} u_V$ gives the direction of viewing ray associated to the point u_V

Remember: the direction of the backprojection of image point $\mathbf{V}$ (viewing ray associated to $\mathbf{V}$) is $\mathbf{M}^{-1}\mathbf{u}_\mathbf{V} = \mathbf{M}^{-1}\,\mathbf{M}\mathbf{d} = \mathbf{d}$

→

Vanishing Point Theorem:

The viewing ray associated to the vanishing point $\mathbf{V}$ of a direction $\mathbf{d}$ is parallel to $\mathbf{d}$

→ APPLICATIONS...

KNOWING M can be usefull for navigation on desired direct

Navigation based on vanishing point of desired direction

to automomous navigate along street... you can find vanishing point of desired direction

mobile robot navigation

when corridor navig. you move parallel to vanishing lines... concurrent on image up to vanishing point

vehicle driving

you can command to move along it!

parallel lines, you drive safely along it!

→ it is interesting to derive the direction of vanishing point in WORLD Reference

Find $\mathbf{V}$ and, knowing $\mathbf{M}$, follow the direction $\mathbf{d} = \mathbf{M}^{-1}\mathbf{V}$ of the corridor

$\mathbf{M} = \mathbf{KR}$ can be estimated through extrinsic + intrinsic calibration

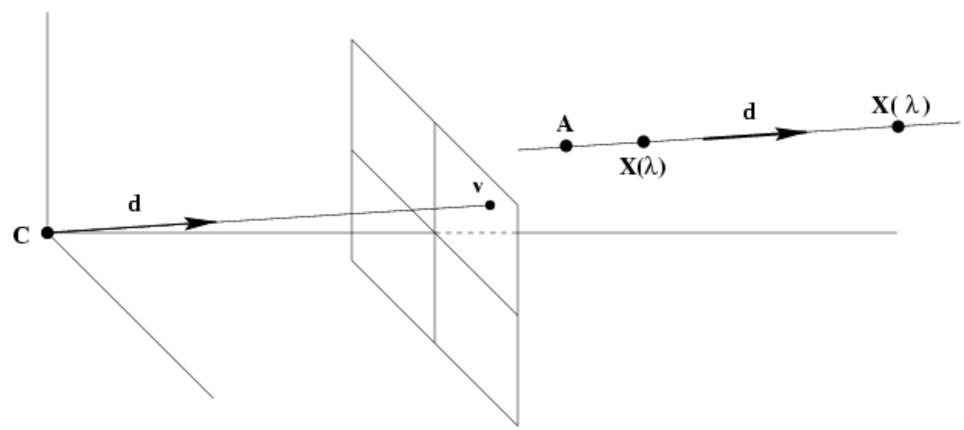
You can command the ROBOT to navigate along that direction!

Vanishing points (d measured wrt camera reference)

$$x(\lambda) = PX(\lambda) = PA + \lambda PD = a + \lambda Kd$$

$$v = \lim_{\lambda \to \infty} x(\lambda) = \lim_{\lambda \to \infty} (a + \lambda Kd) = Kd$$

$$v = PX_\infty = Kd$$



Vanishing points (d measured wrt world reference) $v = PX_\infty = KRd_w$

when you know INTRINSIC parameters (K) by this knowledge:

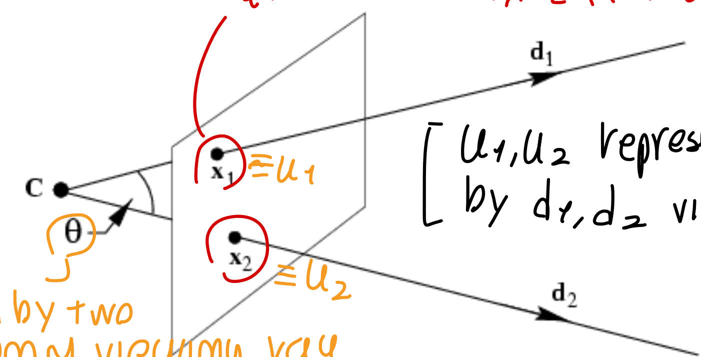
What does INTRINSIC calibration give?

⇓ KNOWING K, from u_1, u_2 you can derive θ

from this x_1, x_2 (OR u_1, u_2)

$$u = K[I|0]\begin{bmatrix} d \\ 0 \end{bmatrix}$$

$$d = K^{-1}u$$



$\equiv u_1$

$\equiv u_2$

[u_1, u_2 represented by d_1, d_2 viewing ray]

angle formed by two direction of viewing ray

angle between d_1, d_2 can be expressed by $\cos\theta$ as:

⇓

$$\cos\theta = \frac{d_1{}^T d_2}{\sqrt{(d_1{}^T d_1)(d_2{}^T d_2)}} = \frac{u_1{}^T (K^{-T} K^{-1}) u_2}{\sqrt{(u_1{}^T (K^{-T} K^{-1}) u_1)(u_2{}^T (K^{-T} K^{-1}) u_2)}}$$

These angles don't depend on the absolute position of the camera

from K and u_1, u_2 you can derive θ, even without extrinsic parameters!

R, t NOT needed! INTRINSIC are enough!

→ Relative position of viewing rays associated to different image points

So, given K, you can consider any couple of image points
u_i, u_j and compute the
relative position of viewing ray of this points.

that matrix $(K^{-T} K^{-1}) = (K K^{T})^{-1}$

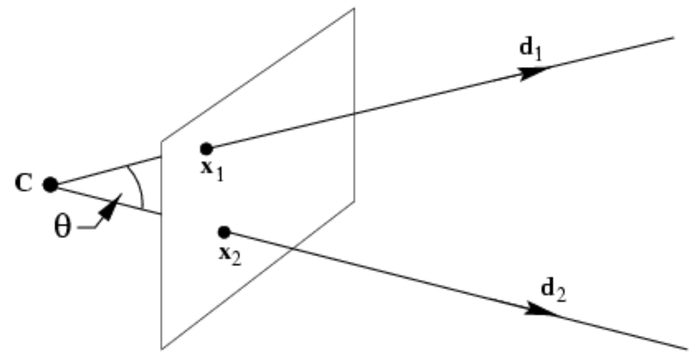
can be defined as a special matrix, since it is often used! $\Longrightarrow$

The ω matrix (sometimes called the IAC)

Define $\omega \triangleq (\mathbf{K}\mathbf{K}^T)^{-1}$ (ω is a symmetric matrix)

[3×3]

using ω matrix, the formula of cos θ gets simpler..



Property: The directions d_1 and d_2 of the viewing rays of two image points u_1 and u_2 form an angle given by

$$\cos\theta = \frac{{u_1}^T \omega\, u_2}{\sqrt{({u_1}^T \omega\, u_1)({u_2}^T \omega u_2)}}$$

Useful properties

very usefull to solve calibration
and localization / rectification
problems...

abstract
properties
usefull in the
future...

The circular points in the space

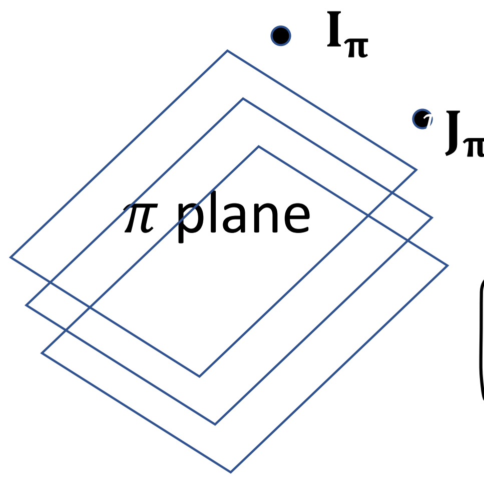
The circular points in the space

pair of points associated to plane π, because in space we have lot of planes π_i and so many I_{π_i}, J_{π_i}

$\Downarrow$

for each π_i has its own pair in 3D geometry

since I, J are points at ∞ of each plane... and all parallel planes share common ℓ_∞, since $I, J \in \ell_\infty$ common through $\pi //$ ... same I, J



$$\left.\begin{array}{l} I = [1 \quad i \quad 0] \\ J = [1 \quad -i \quad 0] \end{array}\right\}$$ plane (2D) coordinate homog

what are I_π, J_π as we chang π value!

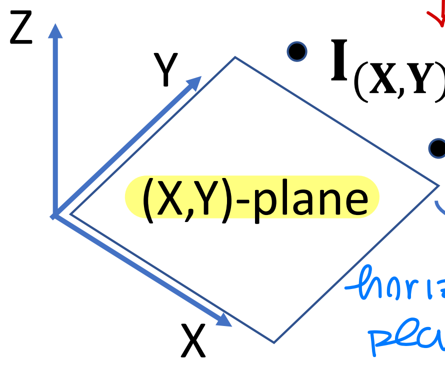
- Each plane π has its own pair of circular points $\mathbf{I}_\pi$, $\mathbf{J}_\pi$

specific of plane, each family share same I, J!

- Parallel planes share the same circular points

The circular points $I = \begin{bmatrix} 1 \\ i \\ 0 \end{bmatrix}$ and $J = \begin{bmatrix} 1 \\ -i \\ 0 \end{bmatrix}$ in the space

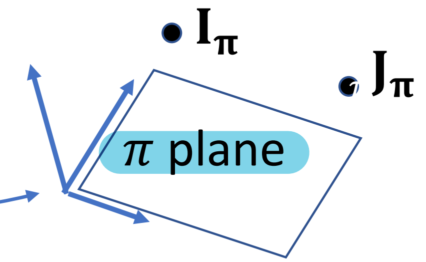
How to study 3D coordinates of this points...



- $I_{(X,Y)} = [1, i, 0, 0]^T$
- $J_{(X,Y)} = [1, -i, 0, 0]^T$

point at ∞

3D coordinates of circular pts $I_{(X,Y)}$ $J_{(X,Y)}$



In space fix Z = 0

horizontal plane Z = 0

$$\mathbf{T}_\pi = \begin{array}{cc} \text{ROTATION} & \text{TRANSLATION} \\ \left[\begin{array}{c|c} \mathbf{R}_{\pi W} & \boldsymbol{o}_\pi \\ \hline 0 \quad 0 \quad 0 & 1 \end{array}\right] & \end{array}$$

T_π

Generic plane π obtained by rototranslating the (X,Y)-plane with isometry $\mathbf{T}_\pi$

In new plane we move also $I_{X,Y}$ $J_{X,Y}$ of X Y-plane!

by ROTOTRANSLATING:

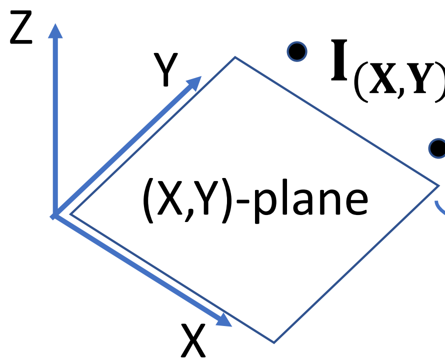
→ 3D coordinates of $\mathbf{I}_\pi, \mathbf{J}_\pi$: $\mathbf{I}_\pi = \mathbf{T}_\pi \mathbf{I}_{(X,Y)} = \begin{bmatrix} \mathbf{R}_{\pi W} \boldsymbol{I} \\ \mathbf{0} \end{bmatrix}$ $\quad \mathbf{J}_\pi = \mathbf{T}_\pi \mathbf{J}_{(X,Y)} = \begin{bmatrix} \mathbf{R}_{\pi W} \boldsymbol{J} \\ \mathbf{0} \end{bmatrix}$

of generic plane π depends only on ROTATION, NOT on translation, since point at ∞!

→ 2D coordinates of circular points of π within π_∞: $\boldsymbol{I}_\pi = \mathbf{R}_{\pi W} \boldsymbol{I}, \ \boldsymbol{J}_\pi = \mathbf{R}_{\pi W} \boldsymbol{J}$

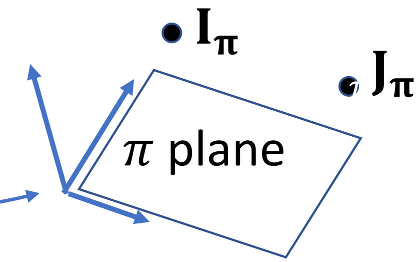
If we take π as plane at infinity ↗ π_∞ as 2D geometry of plane @∞

The circular points $I = \begin{bmatrix} \mathbf{1} \\ \boldsymbol{i} \\ \mathbf{0} \end{bmatrix}$ and $J = \begin{bmatrix} \mathbf{1} \\ -\boldsymbol{i} \\ \mathbf{0} \end{bmatrix}$ in the space



- $\mathbf{I}_{(\mathbf{X},\mathbf{Y})} = [\mathbf{1}, \boldsymbol{i}, \mathbf{0}, \mathbf{0}]^T$
- $\mathbf{J}_{(\mathbf{X},\mathbf{Y})} = [\mathbf{1}, -\boldsymbol{i}, \mathbf{0}, \mathbf{0}]^T$

3D coordinates of circular pts $\mathbf{I}_{(\mathbf{X},\mathbf{Y})}$ $\mathbf{J}_{(\mathbf{X},\mathbf{Y})}$



$$\mathbf{T}_\pi = \begin{bmatrix} & \mathbf{R}_{\pi w} & & \boldsymbol{o}_\pi \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Generic plane π obtained by rototranslating the (X,Y)-plane with isometry $\mathbf{T}_\pi$

→ 3D coordinates of $\mathbf{I}_{\boldsymbol{\pi}}, \mathbf{J}_{\boldsymbol{\pi}}$: $\mathbf{I}_{\boldsymbol{\pi}} = \mathbf{T}_\pi\, \mathbf{I}_{(\mathbf{X},\mathbf{Y})} = \begin{bmatrix} \mathbf{R}_{\pi w} \boldsymbol{I} \\ \mathbf{0} \end{bmatrix}$ $\quad \mathbf{J}_{\boldsymbol{\pi}} = \mathbf{T}_\pi\, \mathbf{J}_{(\mathbf{X},\mathbf{Y})} = \begin{bmatrix} \mathbf{R}_{\pi w} \boldsymbol{J} \\ \mathbf{0} \end{bmatrix}$

from [d 0] homog in 3D → in 2D [d] on plane

Just remove last 0

→ 2D coordinates of circular points of π within π_∞: $\boldsymbol{I}_{\boldsymbol{\pi}} = \mathbf{R}_{\pi w} \boldsymbol{I}$, $\boldsymbol{J}_{\boldsymbol{\pi}} = \mathbf{R}_{\pi w} \boldsymbol{J}$

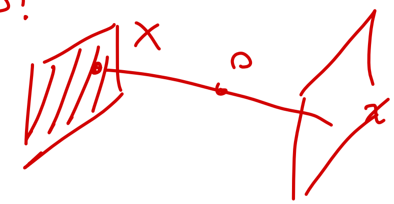
↳ only 3 coord homogeneous

Restricting attention to planar scene on plane π and its image

as 2D to 2D!

$x = H X$

HOMOGRAPHY (3x3)



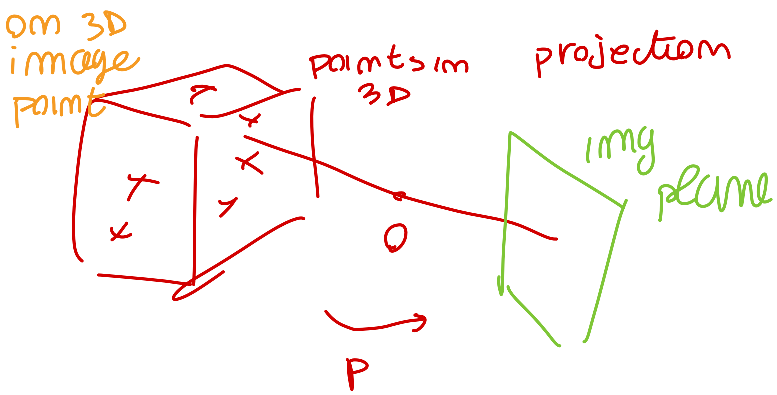
Useful properties

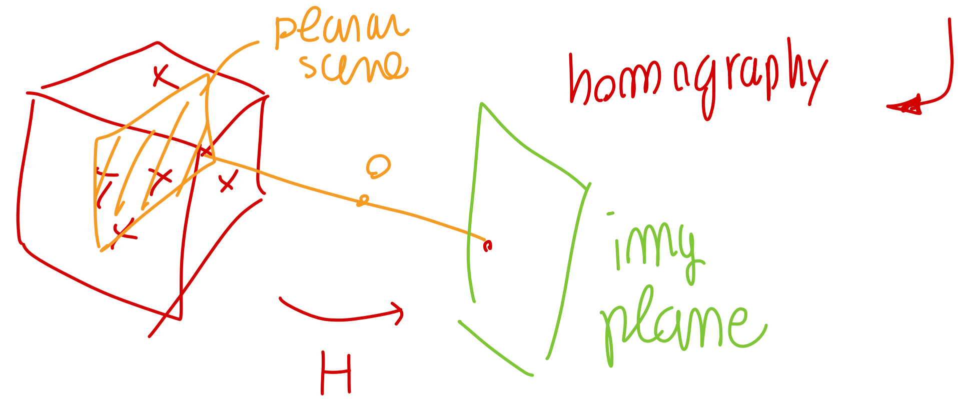
very important! we can discover something about it...

this image of planar scene is related to plane itself by an

The image of a planar scene:

Homography between a plane π and its image

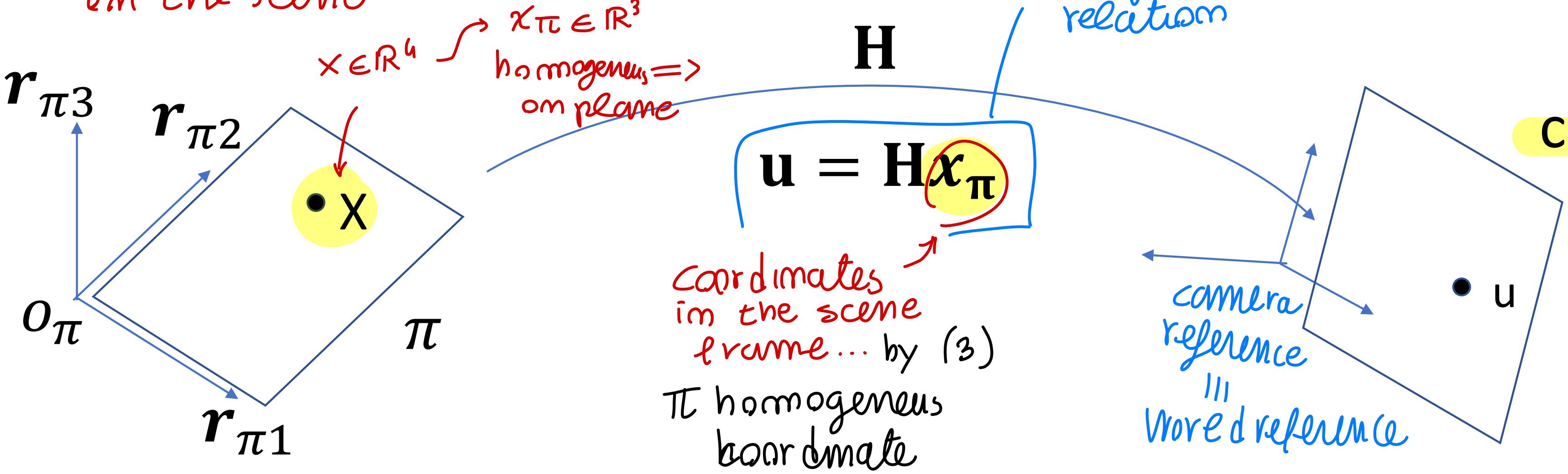




Now that we describe the CAMERA! we want to match Homography 2D to 2D with camera projection model ↴ IMAGE OF PLANAR SCENE how to compute H now that K camera known...

Homography between a plane π and its image

plane π somewhere on the scene



$$\boxed{\mathbf{u} = \mathbf{H}\boldsymbol{x}_\pi}$$

coordinates in the scene frame... by (3)

π homogeneous coordinate

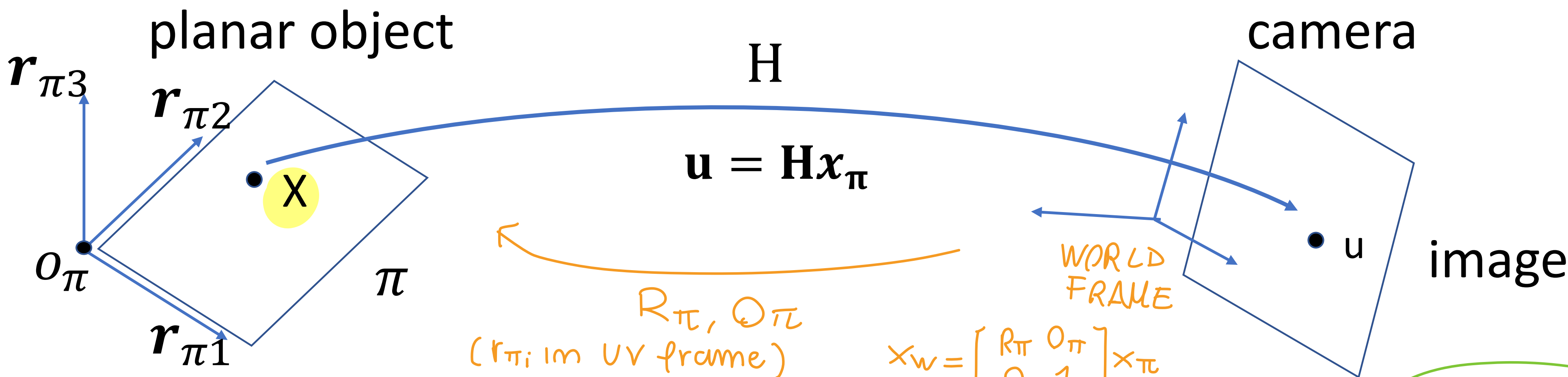
plane π reference: relative pose plane wrt camera → rototranslation $\mathbf{R}_\pi, \boldsymbol{o}_\pi$ ← relation from camera to plane π

same origin!

world reference ≡ camera reference

$$\mathrm{P} = [\mathbf{K} \quad \mathbf{0}]$$

$(t = 0 \quad R = I_3)$



plane reference

world reference ≡ camera reference → $\mathrm{P} = [\mathbf{K} \quad \mathbf{0}]$

First we need to express X in the world frame...

$$\mathbf{X}_{\boldsymbol{\pi}} = \begin{bmatrix} x \\ y \\ 0 \\ w \end{bmatrix}$$

z = 0 in this plane π

$$\mathbf{X}_{\mathbf{w}} = \begin{bmatrix} \boldsymbol{r}_{\boldsymbol{\pi}1} & \boldsymbol{r}_{\boldsymbol{\pi}2} & \cancel{\boldsymbol{r}_{\boldsymbol{\pi}3}} & \mathbf{o}_{\boldsymbol{\pi}} \\ 0 & 0 & \cancel{0} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \\ w \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{r}_{\pi 1} & \boldsymbol{r}_{\pi 2} & \mathbf{o}_{\pi} \\ 0 & 0 & 1 \end{bmatrix}}_{\text{3×3 homogeneous transformation MATRIX}} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

multiplyed by 0

$\boldsymbol{x}_{\boldsymbol{\pi}}$

taking proj. on image plane

$$\Rightarrow \mathbf{u} = \mathbf{P}\mathbf{X}_{\mathbf{w}} = [\mathbf{K} \quad \mathbf{0}] \begin{bmatrix} \boldsymbol{r}_{\pi 1} & \boldsymbol{r}_{\pi 2} & \mathbf{o}_{\pi} \\ \cancel{0} & \cancel{0} & \cancel{1} \end{bmatrix} \boldsymbol{x}_{\boldsymbol{\pi}} = \boxed{\mathbf{K}[\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad \mathbf{o}_{\pi}]}\, \boldsymbol{x}_{\boldsymbol{\pi}}$$

multiplyed by 0

↓ HOMOGRAPHY

function of rotations on $r_{\pi 1}, r_{\pi 2}$ and transl.

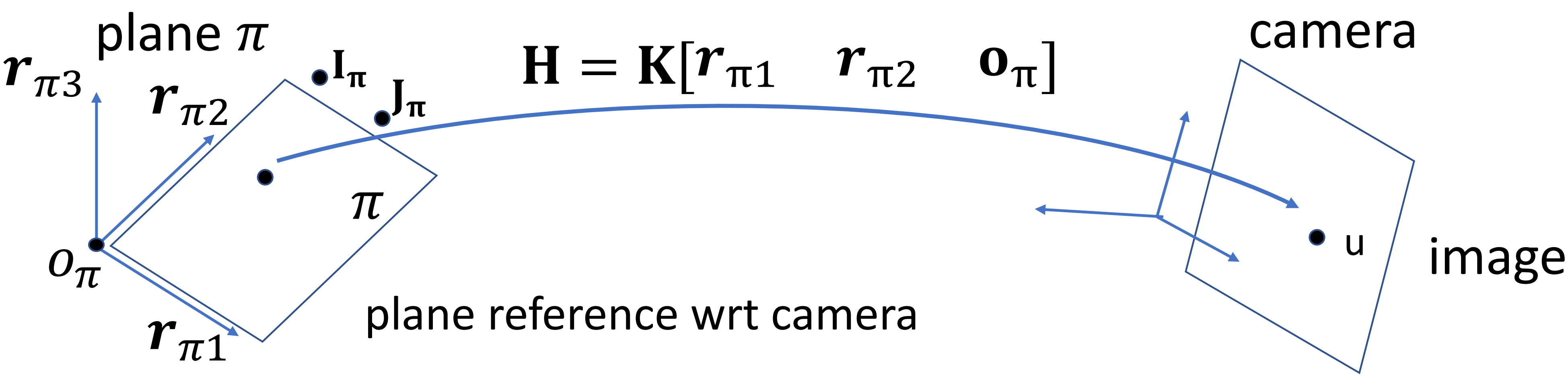
plane-to-image homography $\mathbf{H} = \mathbf{K}[\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad \mathbf{o}_{\pi}]$

The image of a planar scene:
Homography between a plane π and its image

Example:

- Image of the circular points of plane π

$$\mathbf{R}_{\pi} = [\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad \boldsymbol{r}_{\pi 3}]$$: rotation of plane π wrt camera



$$(\boldsymbol{I}'_{\pi}, \boldsymbol{J}'_{\pi}) = \mathbf{K}[\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad \mathbf{o}_{\pi}]\left(\begin{bmatrix}\mathbf{1}\\ \boldsymbol{i}\\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{1}\\ -\boldsymbol{i}\\ \mathbf{0}\end{bmatrix}\right) = \mathbf{K}[\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad \boldsymbol{r}_{\pi 3}]\left(\begin{bmatrix}\mathbf{1}\\ \boldsymbol{i}\\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{1}\\ -\boldsymbol{i}\\ \mathbf{0}\end{bmatrix}\right)$$

$$\rightarrow$$

$$\boxed{\boldsymbol{I}'_{\pi} = \mathbf{K}\mathbf{R}_{\pi}\,\boldsymbol{I}, \boldsymbol{J}'_{\pi} = \mathbf{K}\mathbf{R}_{\pi}\,\boldsymbol{J}}$$

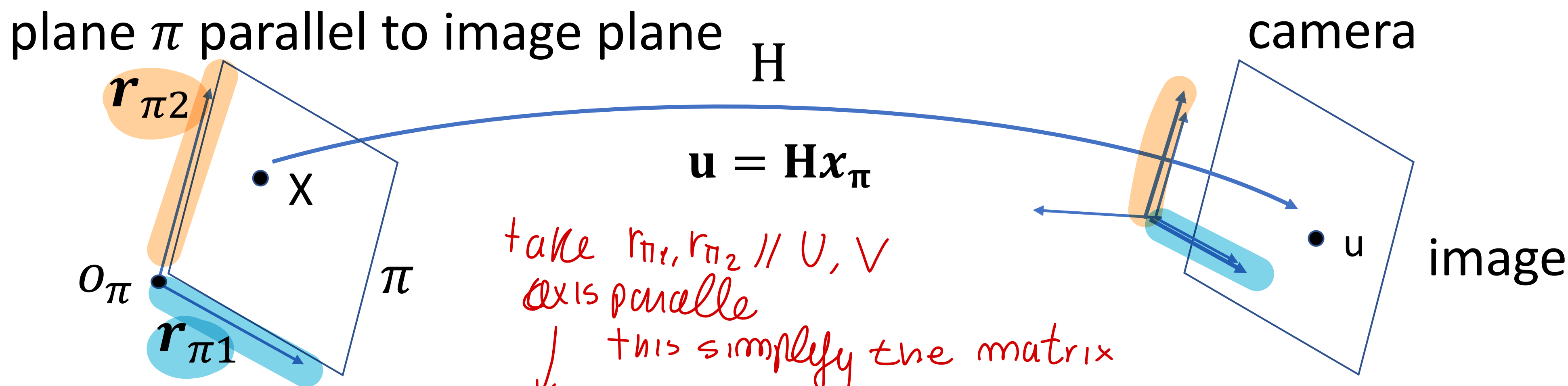
some particular cases...

for ex. when π // image plane

The image of a planar scene: Homography between a plane π and its image

Particular case:

- plane π parallel to the image plane



take $r_{\pi1}, r_{\pi2}$ // U, V
axis parallel
this simplify the matrix

$$\boldsymbol{r}_{\boldsymbol{\pi 1}} = \begin{bmatrix}1\\0\\0\end{bmatrix}, \ \boldsymbol{r}_{\boldsymbol{\pi 2}} = \begin{bmatrix}0\\1\\0\end{bmatrix} \rightarrow [\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad o_\pi] = \begin{bmatrix}1 & 0 & X_o\\0 & 1 & Y_o\\0 & 0 & 1\end{bmatrix}$$

$$\mathbf{u} = \mathbf{K}[\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad \mathbf{o}_\pi]\boldsymbol{x}_{\boldsymbol{\pi}} = \mathbf{K}\begin{bmatrix}1 & 0 & X_o\\0 & 1 & Y_o\\0 & 0 & 1\end{bmatrix}\boldsymbol{x}_{\boldsymbol{\pi}} = \begin{bmatrix}f_x & 0 & U_o\\0 & f_y & V_o\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & X_o\\0 & 1 & Y_o\\0 & 0 & 1\end{bmatrix}\boldsymbol{x}_{\boldsymbol{\pi}}$$

plane-to-image homography $\mathbf{H}$ = 2D-affine * 2D-isometry = 2-D affinity looking at K as homography in 2D MATRIX

$$K = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Affine transformation

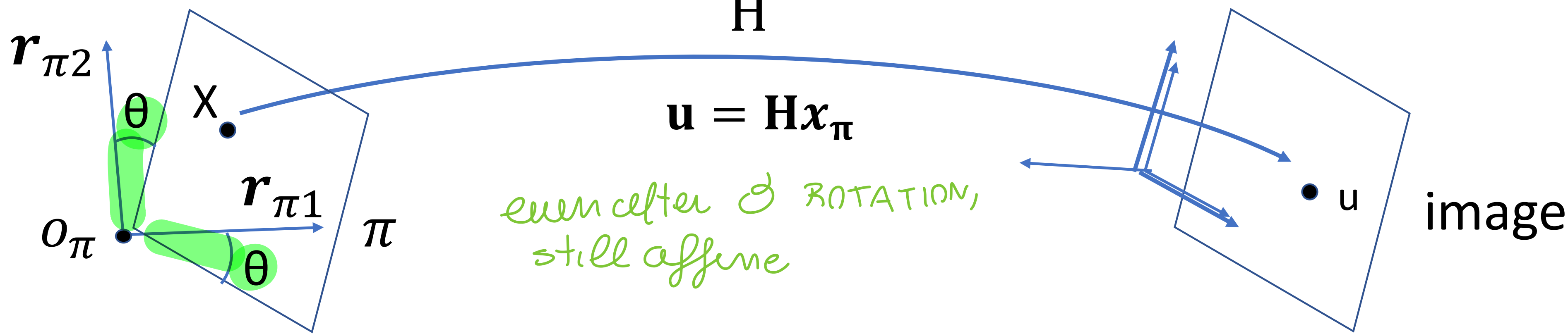
$$* \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} := \underbrace{\text{Affinity}}$$

if plane // img plane

you get affine Transform

even rotating the axis r_π wrt u, v you have still Affinity! $\Longrightarrow$

plane π : rotated reference



$$\boldsymbol{r}_{\boldsymbol{\pi}1} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix},\ \boldsymbol{r}_{\boldsymbol{\pi}2} = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{bmatrix} \rightarrow [\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad o_\pi] = \begin{bmatrix} & R & & X_o \\ & & & Y_o \\ 0 & & 0 & 1 \end{bmatrix}$$

$$\mathbf{u} = \mathbf{K}[\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad \mathbf{o}_\pi]\boldsymbol{x}_{\boldsymbol{\pi}} = \mathbf{K}\begin{bmatrix} & R & & X_o \\ & & & Y_o \\ 0 & & 0 & 1 \end{bmatrix}\boldsymbol{x}_{\boldsymbol{\pi}} = \begin{bmatrix} f_x & 0 & U_o \\ 0 & f_y & V_o \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} & R & & X_o \\ & & & Y_o \\ 0 & & 0 & 1 \end{bmatrix}\boldsymbol{x}_{\boldsymbol{\pi}}$$

plane-to-image homography $\mathbf{H}$ = 2D-affine * 2D-isometry = 2-D affinity with two special cases...

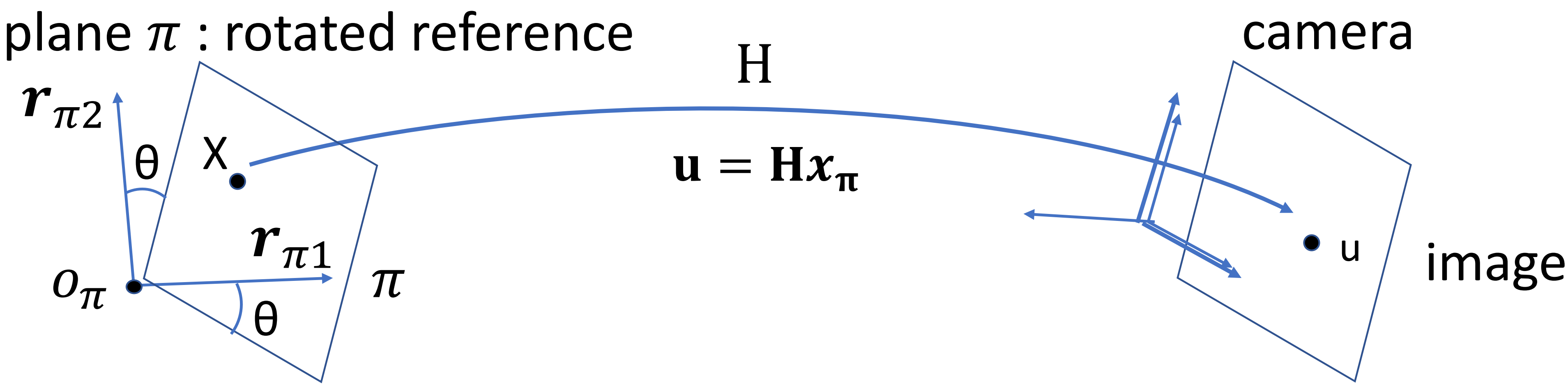
The image of a planar scene: Homography between a plane π and its image

Particular case:

- plane π parallel to the image plane

AND

- natural camera, i.e., square pixels $f_x = f_y = f$



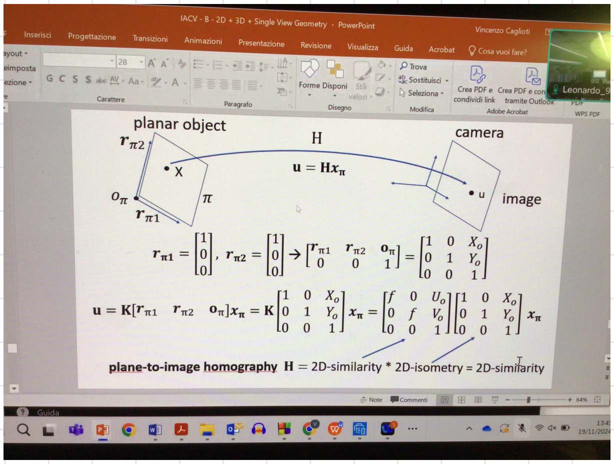
$$\boldsymbol{r}_{\boldsymbol{\pi}\mathbf{1}} = \begin{bmatrix}\cos\theta \\ \sin\theta \\ 0\end{bmatrix}, \ \boldsymbol{r}_{\boldsymbol{\pi}\mathbf{2}} = \begin{bmatrix}-\sin\theta \\ \cos\theta \\ 0\end{bmatrix} \rightarrow [\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad o_\pi] = \begin{bmatrix} & R & X_o \\ & & Y_o \\ 0 & 0 & 1\end{bmatrix}$$

natural camera

$$\mathbf{u} = \mathbf{K}[\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad \mathbf{o}_\pi]\boldsymbol{x}_{\boldsymbol{\pi}} = \mathbf{K}\begin{bmatrix} & R & X_o \\ & & Y_o \\ 0 & 0 & 1\end{bmatrix}\boldsymbol{x}_{\boldsymbol{\pi}} = \begin{bmatrix} f & 0 & U_o \\ 0 & f & V_o \\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix} & R & X_o \\ & & Y_o \\ 0 & 0 & 1\end{bmatrix}\boldsymbol{x}_{\boldsymbol{\pi}}$$

plane-to-image homography $\mathbf{H}$ = 2D-~~affine~~ * 2D-isometry = 2D-isometry

similarity



the shape of what I get when

1) π // plane

&

2) $f_x = f_y = f$

$\Downarrow$

similarity!

$\uparrow$

shape preserved!

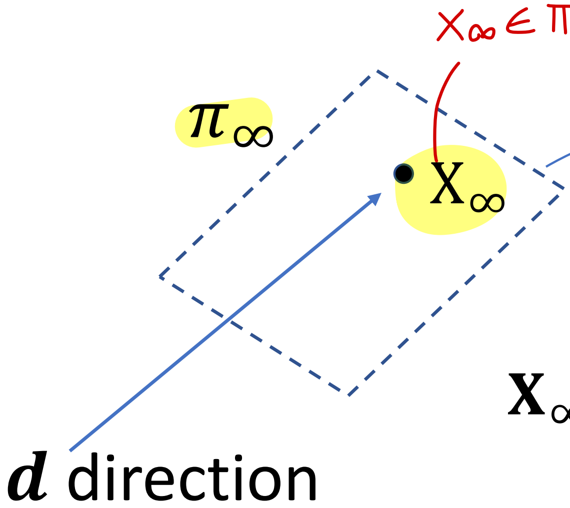
only case when it is preserved image

The image of a planar scene:
Homography between a plane π and its image

Particular case:

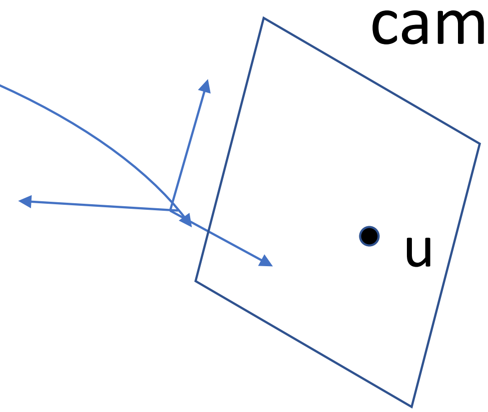
- plane π at the infinity: $\boxed{\pi = \pi_\infty} \Longrightarrow$

Homography $\mathbf{H}$ between the **plane at the infinity $\boldsymbol{\pi}_\infty$** and its image



$\mathbf{H}$

$$\mathbf{u} = \mathbf{H}\boldsymbol{x}_\infty$$



$$\mathbf{X}_\infty = \begin{bmatrix} x \\ y \\ w \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_\infty \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{d} \\ 0 \end{bmatrix}$$

← always a 0 here being at ∞

$$\mathbf{P} = [\mathbf{M} \quad \boldsymbol{m}] = [\mathbf{KR} \quad \mathbf{K}\boldsymbol{t}]$$

camera → world

$$\mathrm{u} = \mathbf{P}\mathbf{X}_\infty = [\mathbf{M} \quad \cancel{\boldsymbol{m}}]\begin{bmatrix} \boldsymbol{x}_\infty \\ 0 \end{bmatrix} = \mathbf{M}\boldsymbol{x}_\infty = \mathbf{K}\boxed{\mathbf{R}}\boldsymbol{x}_\infty$$

classical ROTATION MATRIX

$r_i :=$ from camera to world frame, so axis r_i in W frame

$\boldsymbol{\pi}_\infty$ to image homography: $\mathbf{H} = \mathbf{M} = \mathbf{KR} = \mathbf{K}[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boldsymbol{r}_3]$

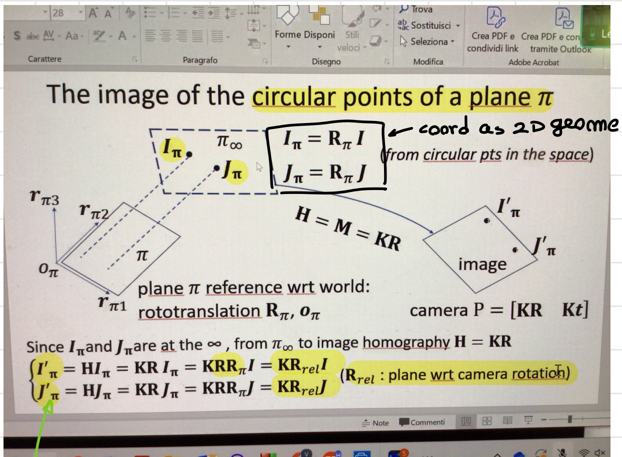
USEFUL PROPERTIES

image of the circular points of a plane π

Now that we can compute homography H between plane π and image...

↓

previously we focus on computing I_π, J_π of any plane!



← coord as 2D geometry

Now, compute I'_π, J'_π img of circular points of π

R_π := rotation matrix from world to π, also circular points I, J moves after world to plane π

from: how to compute images of points on π_∞, as we have seen... we can apply that rule!

$H = KR$

$$\underbrace{R\,R_\pi}_{\text{from camera to plane}} := R_{rel} \quad \text{plane } \pi \text{ respect camera frame}$$

R: from camera to world

R_π: from world to plane

Relative rotation of plane π with respect to camera (indipendent of WORLD ref, we care of where is plane wrt camera)

$\Longrightarrow$

what are

Example: The image of the circular points of a plane π parallel to the image plane

Rree

Since π is parallel to the image plane, the relative rotation ~~$\mathbf{R}_\pi$~~ is the Identity matrix →

I_3 : being π // image plane (NO ROTATION)

$$\boldsymbol{I}'_{\boldsymbol{\pi}} = \mathbf{K}\underbrace{\mathbf{R}_{\cancel{\pi}}}_{\text{Ree}}\boldsymbol{I} = \mathbf{K}\boldsymbol{I} = \begin{bmatrix} f_x & 0 & U_o \\ 0 & f_y & V_o \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \mathbf{1} \\ \boldsymbol{i} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} f_x \\ \boldsymbol{i}f_y \\ \mathbf{0} \end{bmatrix}$$

similarly, $\boldsymbol{J}'_{\boldsymbol{\pi}} = \begin{bmatrix} f_x \\ -\boldsymbol{i}f_y \\ \mathbf{0} \end{bmatrix}$

MAIN PROPERTY

remember ω $\triangleq$ $(\mathbf{KK}^T)^{-1}$ (ω is a symmetric matrix)

From $\boldsymbol{\omega}$ to calibration matrix $\mathbf{K}$: Cholesky factorisation of inverse $\boldsymbol{\omega}^{-1} = \mathbf{KK}^{\mathrm{T}}$

original [1 i 0] image of circular point

remember **Property 1**:

for any plane π, $\boldsymbol{I'}_{\boldsymbol{\pi}} = \mathbf{KR}_{\pi} \boldsymbol{I}$ and $\boldsymbol{J'}_{\boldsymbol{\pi}} = \mathbf{KR}_{\pi} \boldsymbol{J}$

Property 2: images of circular points I', J' of π

for any plane π, $\boldsymbol{I'}_{\boldsymbol{\pi}}{}^{T} \omega\, \boldsymbol{I'}_{\boldsymbol{\pi}} = \mathbf{0}$ and $\boldsymbol{J'}_{\boldsymbol{\pi}}{}^{T} \omega\, \boldsymbol{J'}_{\boldsymbol{\pi}} = \mathbf{0}$

Proof sketch: ($R\pi := R_{rel}$) from camera to plane π

when $A^T B = 0$ orthogonal, holds $I^T I = 0$

for any plane π, $\boldsymbol{I'}_{\boldsymbol{\pi}} = \mathbf{KR}_{\pi} \boldsymbol{I} \rightarrow \boldsymbol{I} = (\mathbf{KR}_{\pi})^{-1} \boldsymbol{I'}_{\boldsymbol{\pi}}$ // self orthogonality

from self-orthogonality $\boldsymbol{I}^T \boldsymbol{I} = [\mathbf{1} \quad \boldsymbol{i} \quad \mathbf{0}][\mathbf{1} \quad \boldsymbol{i} \quad \mathbf{0}]^{T} = \mathbf{0}$ and thus

$\mathbf{0} = \boldsymbol{I}^T \boldsymbol{I} = \boldsymbol{I'}_{\boldsymbol{\pi}}{}^{T} (\mathbf{K}\cancel{\mathbf{R}_{\pi}})^{-T} (\mathbf{K}\cancel{\mathbf{R}_{\pi}})^{-1} \boldsymbol{I'}_{\boldsymbol{\pi}} = \boldsymbol{I'}_{\boldsymbol{\pi}}{}^{T} (\mathbf{KK}^{T})^{-1} \boldsymbol{I'}_{\boldsymbol{\pi}} = \boldsymbol{I'}_{\boldsymbol{\pi}}{}^{T} \omega\, \boldsymbol{I'}_{\boldsymbol{\pi}} = \mathbf{0}$

it desappear independently of Rπ !

being $I_\pi'^{T}\,\omega\,I_\pi' = 0$ $\quad \omega = (KK^T)^{-1}$

the image of any plane's circular point belongs to conic ω

symm ~ CONIC MATRIX ?

IF this ω is a "conic"

the image of any circular points belongs to this conics ω

(it is possible to show how ω is the image of absolute conic)

"IAC"

any image of the circular conic belongs to IAC....

$\Downarrow$

usefull property for CAMERA CALIBRATION ! $\Longrightarrow$

this problem solves many aspect of computer vision....

relationship between ω and K...

↓

if we can estimate ω, is this usefull to find K?

⇓

if we obtain ω... to get K, matrix algebra operation → cholesky factorization

↓

of ω^{-1}

taking $\underbrace{\omega^{-1} = K K^T}$ even if K unkn. by CHOLESKY of ω^{-1} } this give output K matrix!

this is satisfyed by ∞ solution K possible...

$K' = K U$, U orthogon

also $K' K'^T = (K \underbrace{U) (U^T}_{I} K^T) = K K^T$

only one solution is upper triangular, so

$$K = \begin{bmatrix} f_x & 0 & U_0 \\ 0 & f_y & V_0 \\ 0 & 0 & 1 \end{bmatrix}$$

among ∞ sol, only one is upper triangular!

⇓

univoquly find K from ω using Cholesky

Frequently is easyer to estimate ω rather than K...

↑

we derive ω and than K from it often

From projection matrix P to calibration matrix $\mathbf{K}$

20/11

remember

$$\mathrm{P} = [\mathbf{M} \quad \boldsymbol{m}] = [\mathbf{KR} \quad \mathbf{K}\boldsymbol{t}] = K[R \quad t]$$

Rot. trans

Rot/trans of WORLD wrt camera (so from camera to world)

$$\text{from } \omega \triangleq \left(\mathbf{K}\mathbf{K}^{T}\right)^{-1}$$

$$\mathbf{M}\mathbf{M}^{\mathrm{T}} = \mathbf{KR}(\mathbf{KR})^{\mathrm{T}} = \mathbf{K}\mathbf{K}^{\mathrm{T}} = \boldsymbol{\omega}^{-1}$$

R, t

camera

↓

world

From $\mathrm{P} = [\mathbf{M} \quad \boldsymbol{m}]$:

- take $\omega^{-1} = \mathbf{M}\mathbf{M}^{\mathrm{T}}$
- $\mathbf{K}$ = Choleski factorization of $\omega^{-1} = \mathbf{M}\mathbf{M}^{\mathrm{T}} = \mathbf{K}\mathbf{K}^{\mathrm{T}}$

Camera calibration through the ω matrix

we can derive constraints over ω from informations on the scene + image!

USEFUL FACTS

vanishing points = image of points at the ∞: $\mathbf{v_d} = \mathrm{P}\mathrm{X}_\infty = \mathbf{K}\,[\mathrm{R} \quad \mathbf{t}]\begin{bmatrix}\mathbf{d}\\0\end{bmatrix} = \mathbf{KR\,d}$

→ Homography from π_∞ to image plane: $\mathbf{H} = \mathbf{KR}$ → inverse: $\mathbf{d} = (\mathbf{KR})^{-1}\mathbf{v_d}$

From $\boldsymbol{\omega}$ to calibration matrix $\mathbf{K}$: Cholesky factorisation of inverse $\boldsymbol{\omega}^{-1} = \mathbf{K}\mathbf{K}^{\mathrm{T}}$

IF we know this directions for θ known angle... we can derive constraint on ω

sources of informations to estimate ω..

Constraints on $\boldsymbol{\omega}$ from:

various scenarios

extract vanishing points from image

- known angles btw directions, observing: their vanishing points
- (includes) self-orthogonality, observing: images of circular points
- known planar shapes, observing: their images

use this knowledge to constraint ω

circular point is self-orthogonal (equation valid ∀ img of circular points)

→ I', J' helps for $I' \omega I'^{T} = 0$ constr

Constraints on the ω matrix

$\omega = (KK^T)^{-1}$ independent on R

$K^{-T}R^{-T}R^{-1}K^{-1} = K^{-T}K^{-1} = (KK^T)^{-1}$

- **known angles btw directions $\mathbf{d}_i$**, and **their vanishing points $\mathbf{v}_i = \mathbf{KRd}_i$**

from direction relation

$v_i = (KR)d_i$ is the vanishing point v_i and direction d_i relationship

$$\cos\theta = \frac{\mathbf{d_1}^{\mathrm{T}}\mathbf{d_2}}{\sqrt{(\mathbf{d_1}^{\mathrm{T}}\mathbf{d_1})(\mathbf{d_2}^{\mathrm{T}}\mathbf{d_2})}} = \frac{\mathbf{v_1}^{\mathrm{T}}\underbrace{(\mathbf{KR})^{-\mathrm{T}}(\mathbf{KR})^{-1}}_{=\omega}\mathbf{v_2}}{\sqrt{(\mathbf{v_1}^{\mathrm{T}}..\mathbf{v_1})(\mathbf{v_2}^{\mathrm{T}}..\mathbf{v_2})}} = \frac{\mathbf{v_1}^{\mathrm{T}}\omega\mathbf{v_2}}{\sqrt{(\mathbf{v_1}^{\mathrm{T}}\omega\mathbf{v_1})(\mathbf{v_2}^{\mathrm{T}}\omega\mathbf{v_2})}}$$

KNOWN

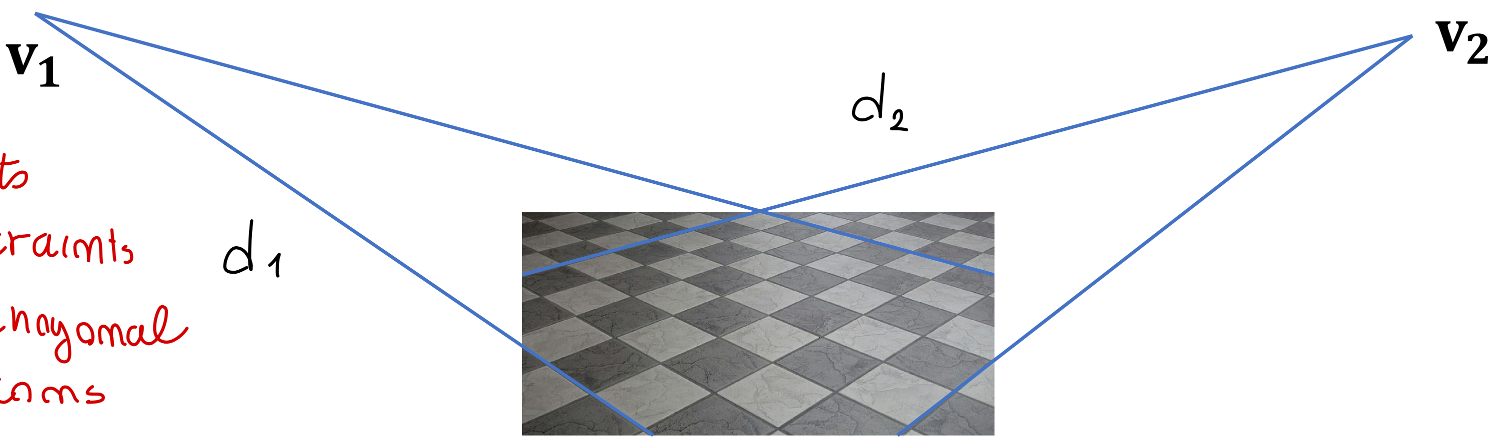
KNOWN vanish points.

set constraint on ω

$(\cos\theta = 0)$

if directions are orthogonal $\theta = 90°$ → **linear constraint:** $\mathbf{v_1}^{\mathrm{T}}\omega\mathbf{v_2} = \mathbf{0}$

it is prefferable to find constraints from orthogonal directions

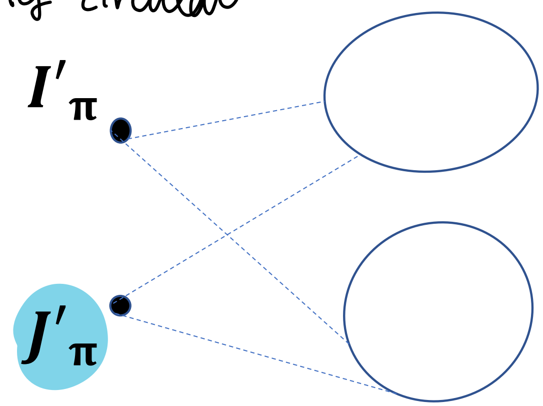


Constraints on the ω matrix

another constraint on ω...

- **images of circular points**
(e.g., intersection of imaged circumferences)

if we can extract I', J' images of circular points, by ex. intersect two circumferences image



from Property 2: $$\boldsymbol{I'_\pi}^T \omega \boldsymbol{I'_\pi} = \mathbf{0}$$

(always holds!)

complex domain

Amounts to 2 independent eqns:

- Real part = 0

and

- Imaginary part = 0

to formulate this equation that involves complex numbers =D

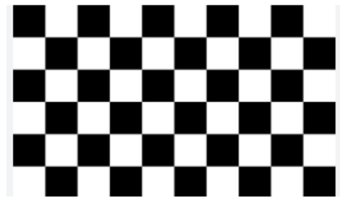
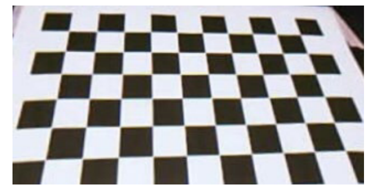
the two equations from Re, Im on I'_π are same as J'_π eqs... same 2 CONSTRAINTS!

($\boldsymbol{J'_\pi}^T \omega \boldsymbol{J'_\pi} = \mathbf{0}$ leads to equivalent equations)

Constraints on the ω matrix

planar scene - image, related by Homography

from camera to plane reference relation 3×3

- known planar shapes and  $\xrightarrow{\mathbf{H}}$  their images

to extract constraint on ω, we start from

$\Downarrow$ expressed as function of K and the pose of object wrt camera

plane-to-image homography $\mathbf{H} = \mathbf{K}[\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad \mathbf{o}_{\pi}]$

$$\rightarrow [\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad \mathbf{o}_{\pi}] = \mathbf{K}^{-1}\mathbf{H} = \left[\mathbf{K}^{-1}\boldsymbol{h}_1 \quad \mathbf{K}^{-1}\boldsymbol{h}_2 \quad \mathbf{K}^{-1}\boldsymbol{h}_3\right]$$

KNOWN PLANAR SHAPE on camera, each frame has different H_i homography between known planar shape and images

vectors $\mathbf{K}^{-1}\boldsymbol{h}_1$, $\mathbf{K}^{-1}\boldsymbol{h}_2$ (i.e. $\boldsymbol{r}_{\pi 1}, \boldsymbol{r}_{\pi 2}$) are orthogonal and have the same module

$\rightarrow$

$$\mathbf{0} = \boldsymbol{h}_1^T\mathbf{K}^{-T}\mathbf{K}^{-1}\boldsymbol{h}_2 = \boldsymbol{h}_1^T\,\omega\,\boldsymbol{h}_2$$

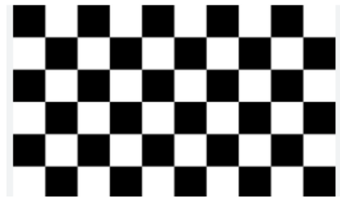
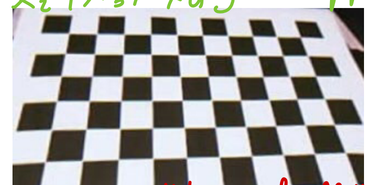
and

$$\mathbf{0} = \boldsymbol{h}_1^T\mathbf{K}^{-T}\mathbf{K}^{-1}\boldsymbol{h}_1 - \boldsymbol{h}_2^T\mathbf{K}^{-T}\mathbf{K}^{-1}\boldsymbol{h}_2 = \boldsymbol{h}_1^T\,\omega\,\boldsymbol{h}_1 - \boldsymbol{h}_2^T\,\omega\,\boldsymbol{h}_2$$

Constraints on the $\boldsymbol{\omega}$ matrix

for instance H derived by x_1, x_2, x_3, x_4 / x_1', x_2', x_3', x_4'

because image mapping fitting to derive H
$x' = Hx$ ...

- **known planar shapes and**  $\xrightarrow{\mathbf{H}}$  **their images**

when H KNOWN if shapes...

from peculiar shape KNOWLEDGE, H can be derived..

plane-to-image homography $\mathbf{H} = \mathbf{K}[\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad \mathbf{o}_{\pi}]$

$$\rightarrow [\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad \mathbf{o}_{\pi}] = \mathbf{K}^{-1}\mathbf{H} = \left[\mathbf{K}^{-1}\boldsymbol{h}_1 \quad \mathbf{K}^{-1}\boldsymbol{h}_2 \quad \mathbf{K}^{-1}\boldsymbol{h}_3\right]$$

unknown

we assume to KNOW H (homography)

vectors $\mathbf{K}^{-1}\boldsymbol{h}_1, \mathbf{K}^{-1}\boldsymbol{h}_2$ (i.e. $\boldsymbol{r}_{\pi 1}, \boldsymbol{r}_{\pi 2}$) are orthogonal and have the same module

$$\rightarrow$$

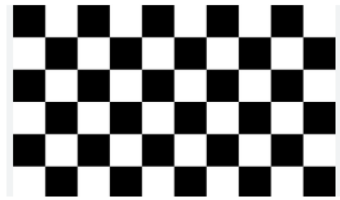
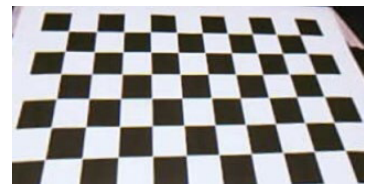
$$\mathbf{0} = \boldsymbol{h}_1^T \mathbf{K}^{-T}\mathbf{K}^{-1}\boldsymbol{h}_2 = \boldsymbol{h}_1^T\, \omega\, \boldsymbol{h}_2$$

and

$$\mathbf{0} = \boldsymbol{h}_1^T \mathbf{K}^{-T}\mathbf{K}^{-1}\boldsymbol{h}_1 - \boldsymbol{h}_2^T \mathbf{K}^{-T}\mathbf{K}^{-1}\boldsymbol{h}_2 = \boldsymbol{h}_1^T\, \omega\, \boldsymbol{h}_1 - \boldsymbol{h}_2^T\, \omega\, \boldsymbol{h}_2$$

Constraints on the ω matrix

unit vectors (same length) orthogonal (being columns of orth matrix)

- **known planar shapes and**  $\xrightarrow{\mathbf{H}}$  **their images**

plane-to-image homography $\mathbf{H} = \mathbf{K}[\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad \mathbf{o}_\pi]$

$\begin{cases} r_{\pi 1} = K^{-1} h_1 \\ r_{\pi 2} = K^{-1} h_2 \end{cases} \Leftarrow$

$\rightarrow [\boldsymbol{r}_{\pi 1} \mid \boldsymbol{r}_{\pi 2} \mid \mathbf{o}_\pi] = \mathbf{K}^{-1}\mathbf{H} = [\mathbf{K}^{-1}\boldsymbol{h}_1 \mid \mathbf{K}^{-1}\boldsymbol{h}_2 \mid \mathbf{K}^{-1}\boldsymbol{h}_3]$

$K^{-1}H \qquad H = [h_1 \; h_2 \; h_3]$ naming

vectors $\mathbf{K}^{-1}\boldsymbol{h}_1$, $\mathbf{K}^{-1}\boldsymbol{h}_2$ (i.e. $\boldsymbol{r}_{\pi 1}, \boldsymbol{r}_{\pi 2}$) are orthogonal and have the same module

$\rightarrow$ orthogonal IFF $(K^{-1}h_1)^T(K^{-1}h_2) = 0$

ORTHOGONAL $\Downarrow$

$$0 = \boldsymbol{h}_1^T \mathbf{K}^{-T}\mathbf{K}^{-1}\boldsymbol{h}_2 = \boldsymbol{h}_1^T\, \omega\, \boldsymbol{h}_2$$

this is equivalent to $\mathrm{Re}, \mathrm{Im}(I_\pi'^T \omega I_\pi' = 0)$

only in Re numbers equations!

and

$\Uparrow$ SAME MODULE (they same has $r_{\pi 1} = r_{\pi 2}$)

$$0 = \boldsymbol{h}_1^T \mathbf{K}^{-T}\mathbf{K}^{-1}\boldsymbol{h}_1 - \boldsymbol{h}_2^T \mathbf{K}^{-T}\mathbf{K}^{-1}\boldsymbol{h}_2 = \boldsymbol{h}_1^T\, \omega\, \boldsymbol{h}_1 - \boldsymbol{h}_2^T\, \omega\, \boldsymbol{h}_2$$

by this equations, calibration device $\Longrightarrow$

Camera calibration from images of known planar shapes (Zhang method)

in between images we derive homography

For each homography $\mathbf{H} = [\boldsymbol{h}_1 \quad \boldsymbol{h}_2 \quad \boldsymbol{h}_3]$

5 dof in ω ... we need 3 images $(3\times 2=6)>5$ to collect all data

$\omega = \omega^T$ symm matrix by definition

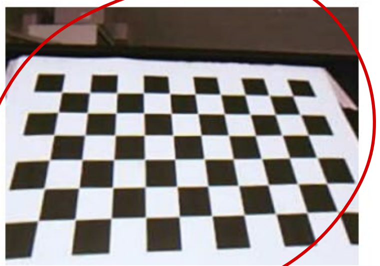
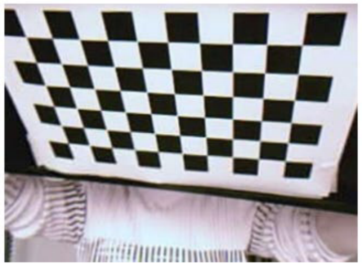
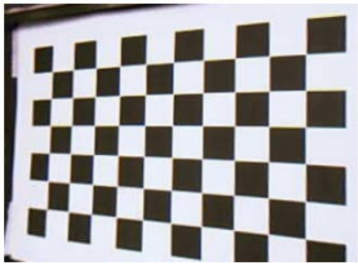
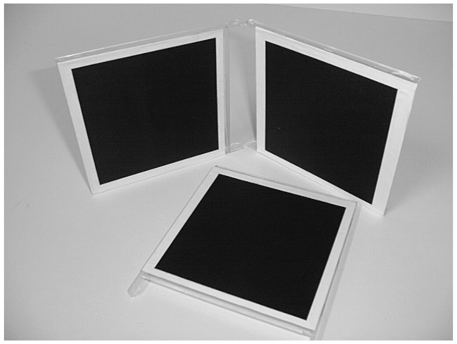
$$\boldsymbol{h_1}^T \omega \boldsymbol{h_2} = 0$$

$$\boldsymbol{h_1}^T \omega \boldsymbol{h_1} - \boldsymbol{h_2}^T \omega \boldsymbol{h_2} = 0$$

~~6 dof~~ in ω to est. 5 elements due to homog.

2 homogeneous equations in ω → at least 3 homographies needed

we know the object in image. Its shape is KNOWN!

e.g.    or 

from Cholesky factorisation of ω^{-1} → $\mathbf{K}$

Camera calibration from images of known planar shapes (Zhang method)

MATLAB provide toolbox ⇒

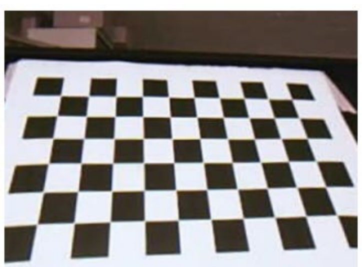
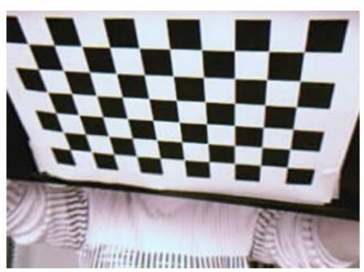
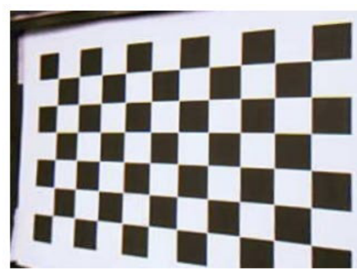
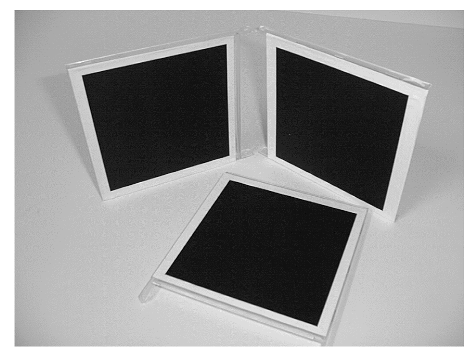
we get 2 equations for each homography!
⇓
6 equations, 3 images to estimate the whole ω matrix

For each homography $\mathbf{H} = [\boldsymbol{h}_1 \quad \boldsymbol{h}_2 \quad \boldsymbol{h}_3]$

$$\boldsymbol{h_1}^T \omega \boldsymbol{h_2} = 0$$

$$\boldsymbol{h_1}^T \omega \boldsymbol{h_1} - \boldsymbol{h_2}^T \omega \boldsymbol{h_2} = 0$$

2 homogeneous equations in ω → at least 3 homographies needed

e.g.    or 

from Cholesky factorisation of ω^{-1} → $\mathbf{K}$

even if our model is simplyfied, when real camera has FOV broad, we have destorsion where straight lines become curved...

Matlab camera calibration toolbox

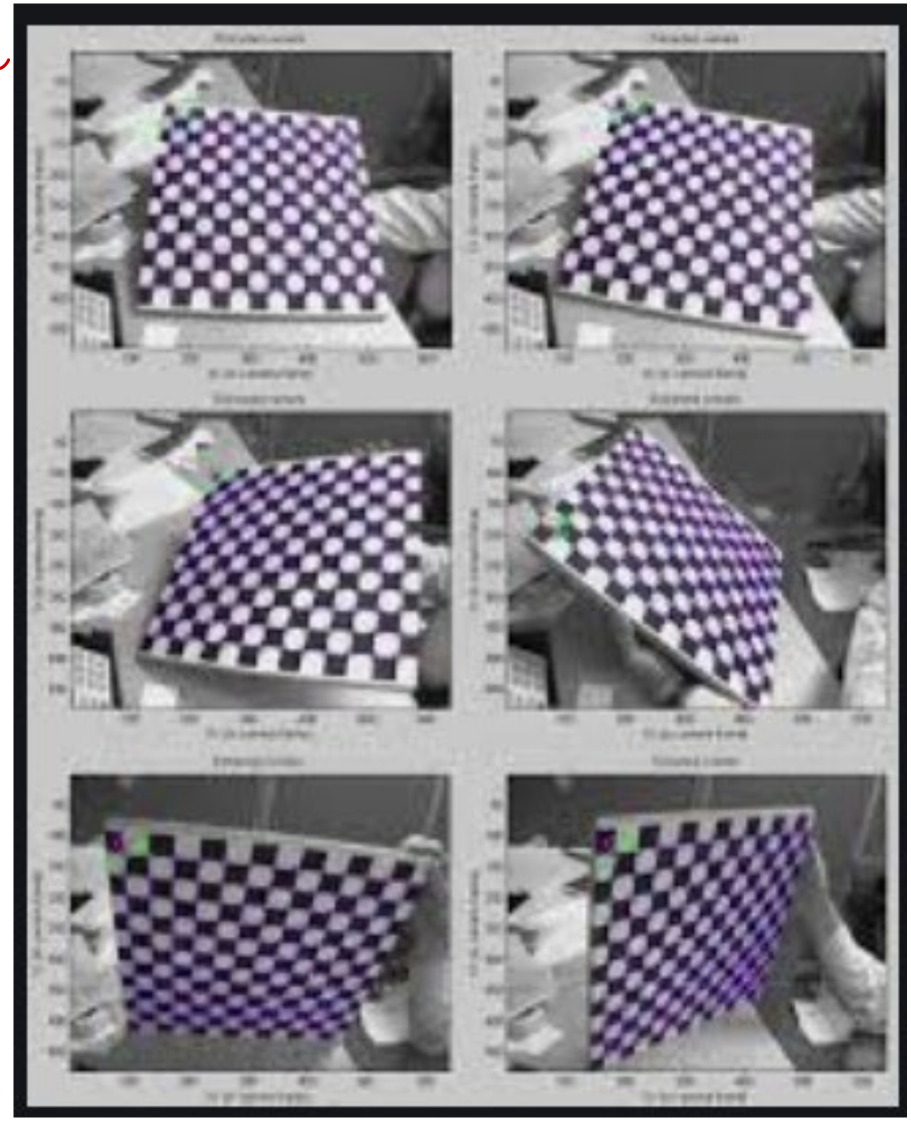
- implements Zhang method
- planar target (easily printable)
- several images (~ 20) to cope with noise

help to better perform!

- also estimates distortion param.
- provides accurate calibration

you get accurate camera

try to show up to ~20 images



Matlab Calibration Toolbox also contains estimation of **distortion** parameters

$$x = x_o + (x_o - c_x)(K_1 r^2 + K_2 r^4 + \ldots)$$
$$y = y_o + (y_o - c_y)(K_1 r^2 + K_2 r^4 + \ldots)$$

$$r = (x_o - c_x)^2 + (y_o - c_y)^2 \quad .$$

Scenario n. 1

you can exploit angle θ expression and v_1, v_2, ω in various way...

Known: vanishing points and angle θ between directions

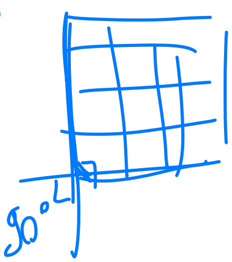
$$\cos\theta = \frac{\mathbf{d_1}^T\mathbf{d_2}}{\sqrt{(\mathbf{d_1}^T\mathbf{d_1})(\mathbf{d_2}^T\mathbf{d_2})}} = \frac{\mathbf{v_1}^T\omega\mathbf{v_2}}{\sqrt{(\mathbf{v_1}^T\omega\mathbf{v_1})(\mathbf{v_2}^T\omega\mathbf{v_2})}}$$

extract from image

angle KNOWN whatever planar scene

→

ex:



constraint on the ω matrix

(linear if directions are orthogonal)

Scenario n. 2

Alternative scenario?

Known: vanishing points and ω matrix

$$\cos\theta = \frac{\mathbf{d_1}^{\mathrm{T}}\mathbf{d_2}}{\sqrt{(\mathbf{d_1}^{\mathrm{T}}\mathbf{d_1})(\mathbf{d_2}^{\mathrm{T}}\mathbf{d_2})}} = \frac{\mathbf{v_1}^{\mathrm{T}}\omega\mathbf{v_2}}{\sqrt{(\mathbf{v_1}^{\mathrm{T}}\omega\mathbf{v_1})(\mathbf{v_2}^{\mathrm{T}}\omega\mathbf{v_2})}}$$

→

compute angle θ between directions

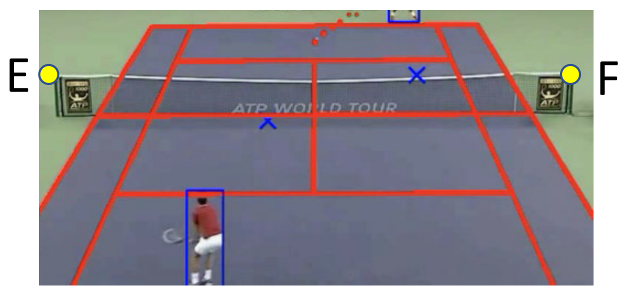
→ reconstruction of the shape

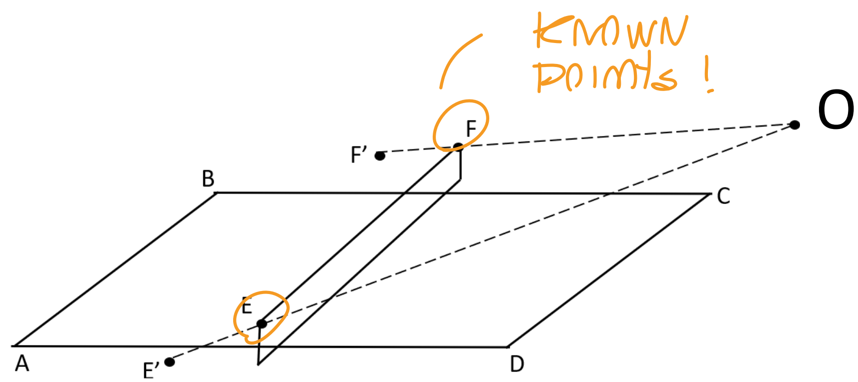
allow you to reconstruct angles! ~ shape planar

EXAMPLE

to estimate both K, R, t (also relative pose)

Extrinsic + intrinsic camera calibration from a single image of a known planar shape and known camera center position



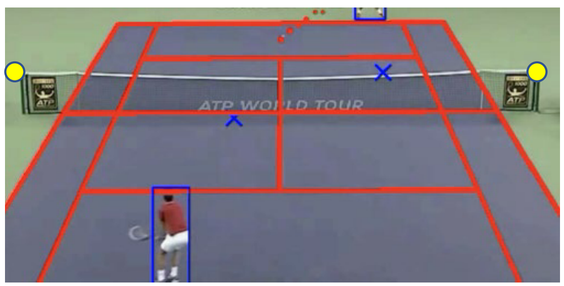


when planar shape is known, official measures fully known on scene + you know where is camera position wrt field... How to estimate it?

Extrinsic + intrinsic camera calibration from a single image of a known planar shape and known camera center position

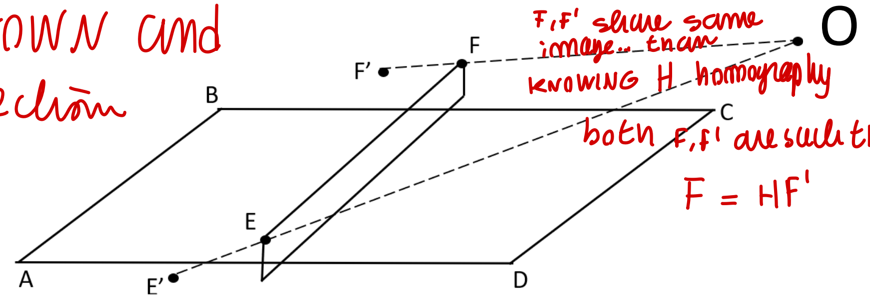
KNOWING those points, you can derive view point ---

KNOWING a straight line between O (camera center)

E  F

F KNOWN and projection F' on field plane

↓



F,F' share same image.. than KNOWING H homography both F,F' are such that $F = HF'$

≃ img of point behind Net! ⇒ extract 2D coordinate...

FROM shape you know homography H points - image

KNOWING img - plane homography

$x \equiv F$ of image before

$X = Hx'$ you want to see x' untie the court plane..

to project it, KNOWING the image x

$x = Hx' \Rightarrow$ on plane

$x' = \underbrace{(H^{-1})}_{\text{KNOWN!}} x$

$\Downarrow$

you can discover where this points are on the plane

So to find the position of eye O respect field, you use

EE', FF' ← 2 lines de, df crossing!

↑

you should know some depths...

and you can find position of camera crossing this KNOWLEDGE

IF you have a CALIBRATED camera...

LOCALIZATION problem

from camera calibrated + image of known planar object

assumes $\omega(K)$ is KNOWN ⇐

↓

you can localize the camera wrt object

while on previous ex.

E, F where sufficient points to find O
even without knowing K!

↓

still can determ E', F' by intersecting space lines!

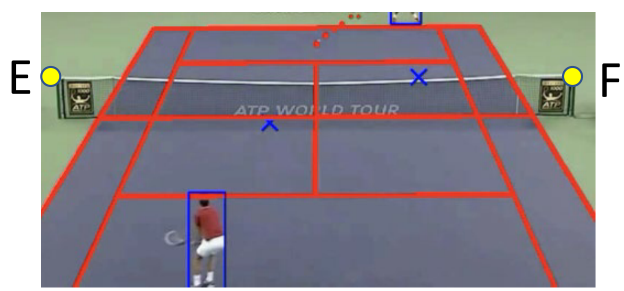
↳ once you identify those points...

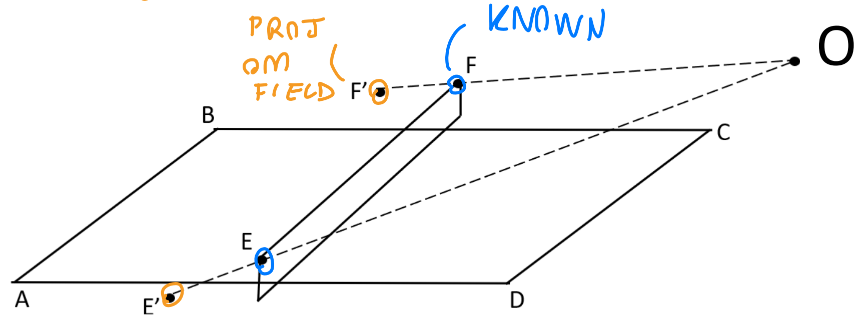
CALIBRATION ⟹

estimate both K and R,t (scene-camera relative position) ⇒ sometimes is possible from a single image, when you know planar scene

Extrinsic + intrinsic camera calibration from a single image of a known planar shape and known camera center position

K & R,t estimate ... + you know planar scene
+ you know camera position wrt field → you can estimate it





you know points on the tennis court! ⇒ than you derive information

by image of net ... you know the shape of court, you know the homography between F and F', E and E'
than you can inverse homography ...

KNOWING image → plane homography..

$\underset{\text{image}}{x} = H \underset{\text{plane}}{x'}$ ⇒ you want to know the projection on the court plane of point!

KNOWN

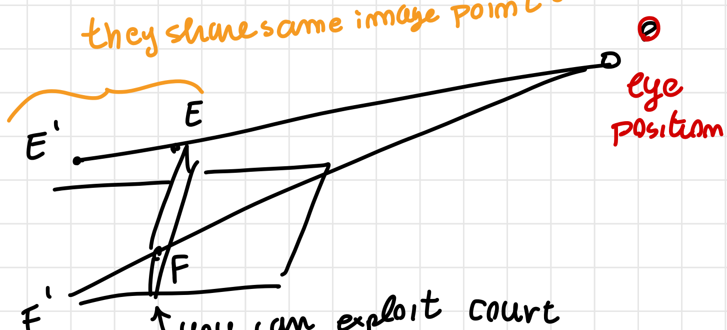
$F = HF'$ on the plane..

when...

↳ $F' = H^{-1}F$

you can discover where are those points

than this helps you to find camera position by letting two lines FF', EE' crossing ←



↳ and H estimated by 4 plane points!

KNOWING homography H, then court points can be found applying inverse homography to their image

↑ you can exploit court information crossing the lines!

you need points on the plane for it...

↓

you can anyway derive it IF camera is calibrated

FROM K (calibrated camera) + image of KNOWN planar object } → you can localize camera wrt object

ASSUMING KNOWLEDGE of K

↑ while in example above you have uncalibrated camera K(?), you still derive eye position intersecting space lines

⇒ Calibration from known plane π and camera center O

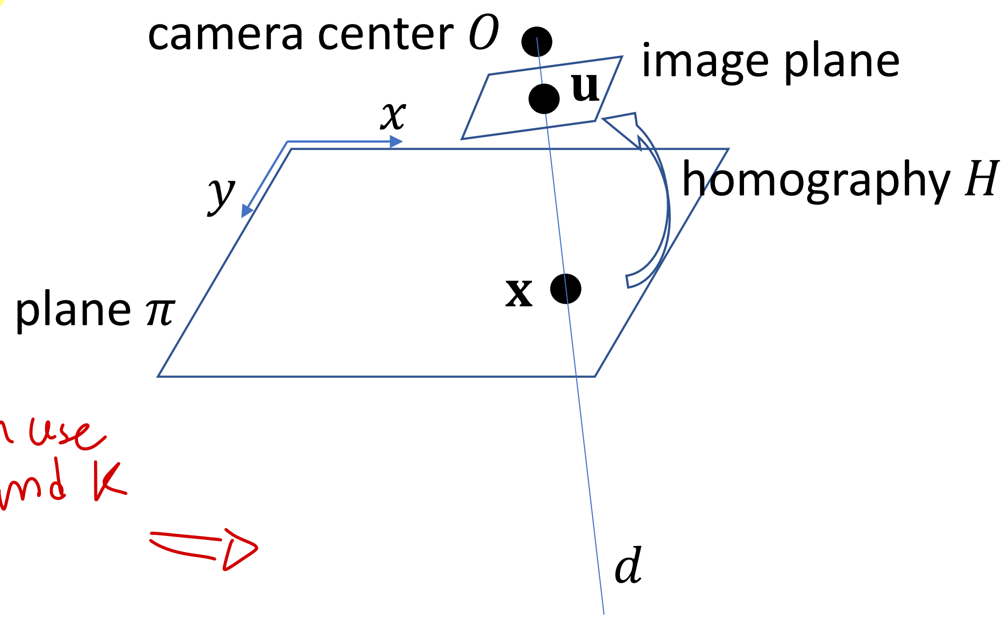
(shape of field) (derived)

Let us refer the coordinates to a reference attached to a known plane π : a generic point on this plane has homogeneous coordinates

$\mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$, while $O = \begin{bmatrix} \mathrm{x}_o \\ \mathrm{y}_o \\ \mathrm{z}_o \end{bmatrix}$ are the known cartesian coordinates of the camera center,

⇓

we can find relationship between any points on court and its image ⇒ you can use it to find K ⇒



Calibration from known plane π and camera center O

1) Estimate homography H from known points on the plane π and their images
(by using set of equations on points)

2) Call M_o the matrix relating any point $\mathbf{x}$ on π to the direction $\mathbf{d}$ of a ray from O to $\mathbf{x}$:

written in terms of image

homogeneous coord of point

$$\mathbf{d} = \begin{bmatrix} x - \mathrm{x}_o \\ y - \mathrm{y}_o \\ z - \mathrm{z}_o \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\mathrm{x}_o \\ 0 & 1 & -\mathrm{y}_o \\ 0 & 0 & -\mathrm{z}_o \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \rightarrow M_o^{-1} = \begin{bmatrix} 1 & 0 & -\mathrm{x}_o \\ 0 & 1 & -\mathrm{y}_o \\ 0 & 0 & -\mathrm{z}_o \end{bmatrix}$$

on the plane

z=0, point on plane

3) Compute the matrix $\mathbf{M}$ relating any image point $\mathbf{u}$ to the direction $\mathbf{d}$ of its viewing ray from $\mathbf{u} = H\,\mathbf{x}$, is $\mathbf{d} = M_o^{-1}\mathbf{x} = M_o^{-1}H^{-1}\mathbf{u} \rightarrow \mathbf{M}^{-1} = M_o^{-1}H^{-1}$

4) Q-R decompose matrix $\mathbf{M}^{-1}$ as $\mathbf{M}^{-1} = \mathbf{R}^{-1}\mathbf{K}^{-1}$, where $\mathbf{K}$ is the camera intrinsic calibration matrix and $\mathbf{R}^{-1}$ is the rotation matrix from the world (reference attached to π) to the camera

Calibration from known plane π and camera center O

1) Estimate homography H from known points on the plane π and their images

2) Call M_o the matrix relating any point $\mathbf{x}$ on π to the direction $\mathbf{d}$ of a ray from O to $\mathbf{x}$:

homography between any x on the plane, and direction of the line joining view point with that x

$$\mathbf{d} = \begin{bmatrix} x - \mathrm{x}_o \\ y - \mathrm{y}_o \\ -\mathrm{z}_o \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\mathrm{x}_o \\ 0 & 1 & -\mathrm{y}_o \\ 0 & 0 & -\mathrm{z}_o \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \rightarrow {M_o}^{-1} = \begin{bmatrix} 1 & 0 & -\mathrm{x}_o \\ 0 & 1 & -\mathrm{y}_o \\ 0 & 0 & -\mathrm{z}_o \end{bmatrix}$$

P = [M m] submatrix of Projection Matrix

3) Compute the matrix $\mathbf{M}$ relating any image point $\mathbf{u}$ to the direction $\mathbf{d}$ of its viewing ray from $\mathbf{u} = H\,\mathbf{x}$, is $\mathbf{d} = {M_o}^{-1}\mathbf{x} = {M_o}^{-1}H^{-1}\mathbf{u} \rightarrow \mathbf{M}^{-1} = {M_o}^{-1}H^{-1}$ ← $u = Md$

point in 3D on plane ↖ relation from direction to image $d = [\cdot]\cdot u$

4) Q-R decompose matrix $\mathbf{M}^{-1}$ as $\mathbf{M}^{-1} = \mathbf{R}^{-1}\mathbf{K}^{-1}$, where $\mathbf{K}$ is the camera intrinsic calibration matrix and $\mathbf{R}^{-1}$ is the rotation matrix from the world (reference attached to π) to the camera

⇑ to extract K, R from matrix $M = KR$ we can extract K, R by Q-R decomposition (MATLAB!)

Q-R decomposition: decompose matrix

$\Downarrow$

$$M^{-1} = R^{-1} K^{-1}$$

$\downarrow$

you can use it to extract K, R from M! $\leftarrow$ Q-R decompose on $A \in \mathbb{R}^{3\times3}$:= orthogonal matrix $\times$ triangular matrix

$\downarrow -1$

A^{-1} = upper triangular $\times$ orthogonal matrix

Camera calibration from images of an **unknown** planar scene

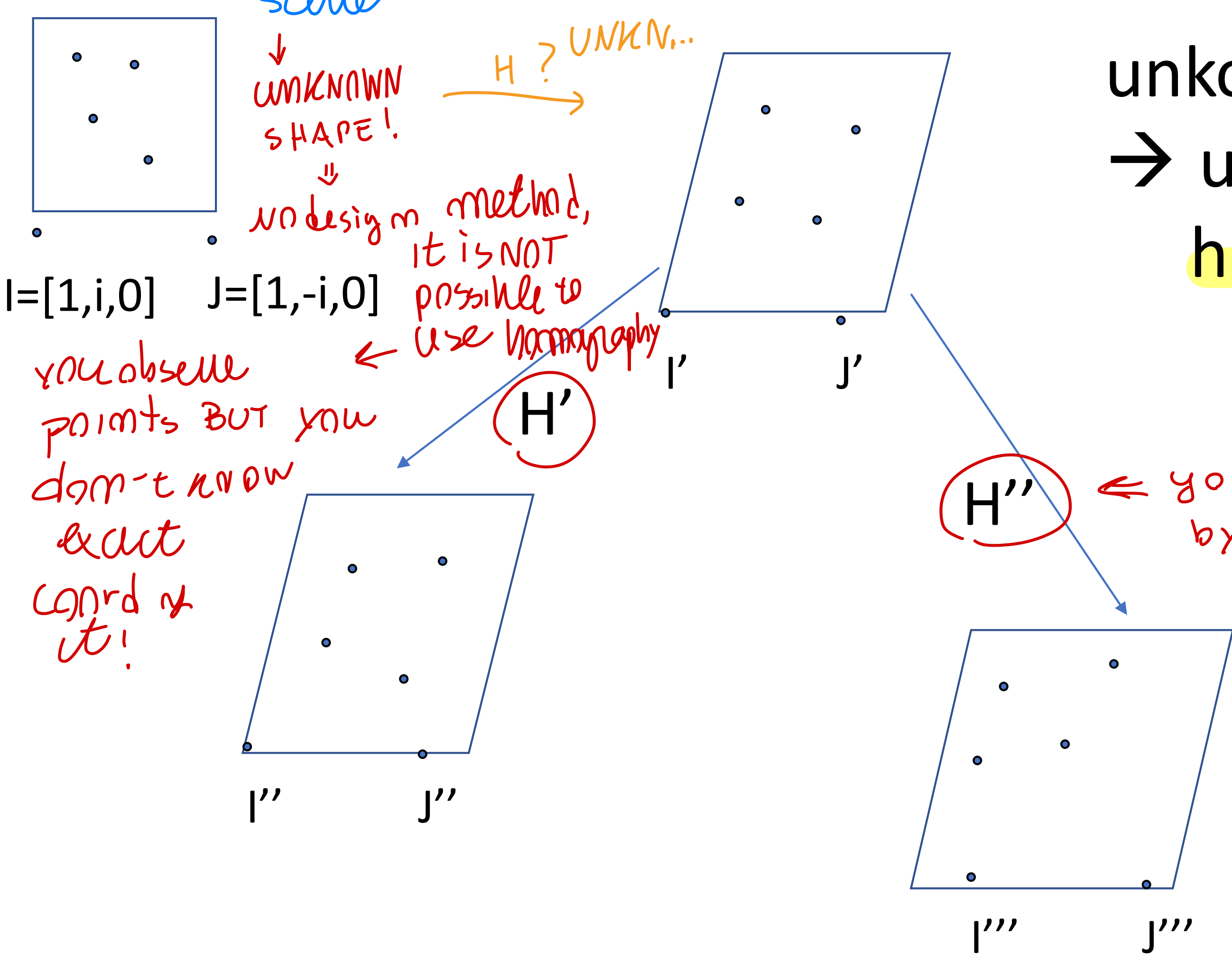
MORE difficult case...

when you have lot of images BUT the scene observed has UNKNOWN shape

it is still possible to calibrate camera!

Real planar scene

you observe those images,



instead to use scene to image homography

unkown planar scene

→ use image-to-image homographies

relating images of UNKNOWN planar scene!

← you can compute this by points correspondence!

you can rely on images of circular points

this involves new unknowns!

$$I'' = H'I'$$

$$I'^T \omega I' = 0$$

$$\left(I''^T \omega I'' = 0\right)$$

you use relationship between images

unknowns is the same, NOT increased!

$$I'^T H'^T \omega H' I' = 0$$

since planar scene is unkn. you are NOT able to reconstruct I', J'...

⇓ therefore I', J' unknown... you can correlate it over more images!

to avoid new unkn. you can use homography

images of circular points

Unknowns: I' and J' and ω → at least 5 images

(each nonlinear eqn leads to 2 constraints: Re and Im part)

we need 5 images overall, 10 equations to solve all unknowns

(while in Zhang method 3 images were enough... here 5 at least)

Camera calibration from images of an **unknown** planar scene

- take images of a planar scene with camera with constant $\mathbf{K}$
- estimate image-image homographies
- formulate equations

$$I'^T \omega I' = 0$$

$$I'^T H'^T \omega H' I' = 0$$

- solve them for ω and I'
- then take the inverse matrix ω^{-1}
- find $\mathbf{K}$ by Cholesky factorisation of $\omega^{-1} = \mathbf{K}\mathbf{K}^{\mathrm{T}}$

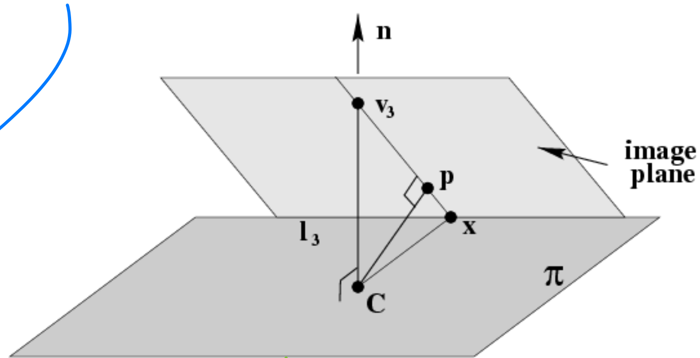
USUALLY LESS ACCURATE THAN ZHANG METHOD

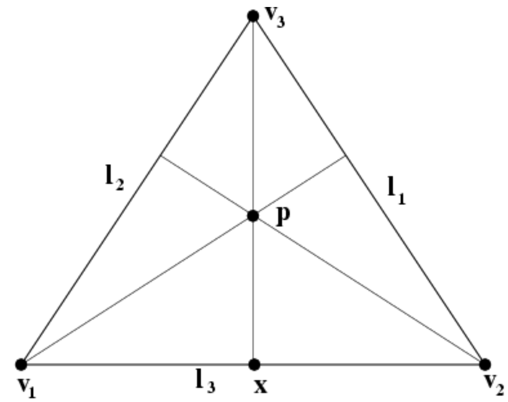
Calibration of natural cameras ($f_X = f_Y = f$) from vanishing points of mutually orthogonal directions

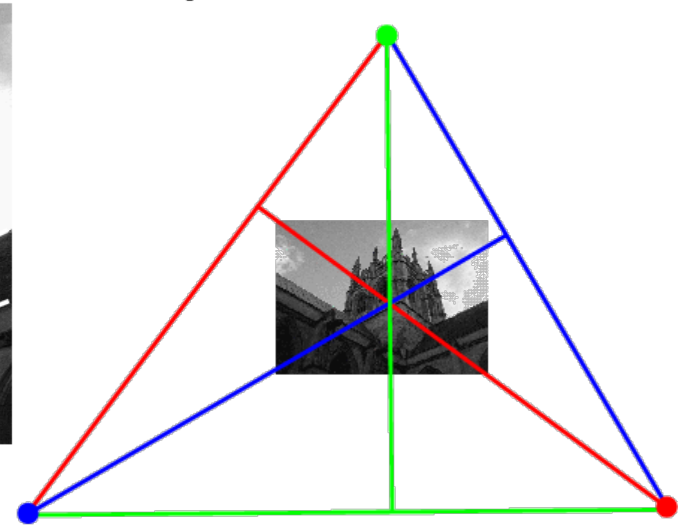
Calibration of NATURAL CAMERAS from vanishing points of orthogonal directions

in this case PROBLEM is simplifyed! we have less unknowns!

You observe 3 vanishing points ⇒ just 3 constr are enough







Orthogonality relation

$$\cos\theta = \frac{\mathrm{v}_1{}^{\mathrm{T}}\omega \mathrm{v}_2}{\sqrt{(\mathrm{v}_1{}^{\mathrm{T}}\omega \mathrm{v}_1)(\mathrm{v}_2{}^{\mathrm{T}}\omega \mathrm{v}_2)}} = 0$$

$$\mathbf{K} = \begin{bmatrix} f & 0 & U_o \\ 0 & f & V_o \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \omega = (\mathbf{K}\mathbf{K}^T)^{-1} = \begin{bmatrix} 1 & 0 & -U_0 \\ * & 1 & -V_0 \\ * & * & f^2 + U_0^2 + V_0^2 \end{bmatrix}$$

3 unknowns

$$\Downarrow$$

you need 3 mutually orthogonal vanishing points $\Longrightarrow$

$$\mathrm{v}_1{}^{\mathrm{T}}\omega \mathrm{v}_2 = 0$$
$$\mathrm{v}_2{}^{\mathrm{T}}\omega \mathrm{v}_3 = 0$$
$$\mathrm{v}_3{}^{\mathrm{T}}\omega \mathrm{v}_1 = 0$$

3 equations

linear in 3 unkns.

3 vanishing points mutually orthogonal

[usefull in HOMEWORK 2024]

Camera calibration from

very frequent case...

- a planar face of known (reconstructed) shape and
- a vanishing point of the normal to the plane

① first step is to rectify a planar object...
then...

② you also have V of direction perpendicular to planar shape

you reconstruct shape of horizontal face + you reconstruct vanishing point of vertical direction ex. in HOMEWORK.

you rectify and you see
horiz. vanishing points
↓
allow reconstruction
+ you have VERTICAL vanishing

1) reconstruct horizontal planar face

2) use additional v vanish point
vetical to complete camera calibration

↳ Ex...

calibration after reconstruction of a planar face

PLANAR SHAPE + vertical vanishing point at crossing of lines

Shape of the face: known after metric rectificaton

you rectify
upper face

How much additional information is needed to calibrate the camera?

- Vanishing point of the direction normal to the reconstructed face

up to know
you reconstruct
without using
vertical V, to
find ω matrix
(4 unkn)

Vanishing point of vertical direction

Camera calibration (assume skew = 0)

- Only four unknowns:

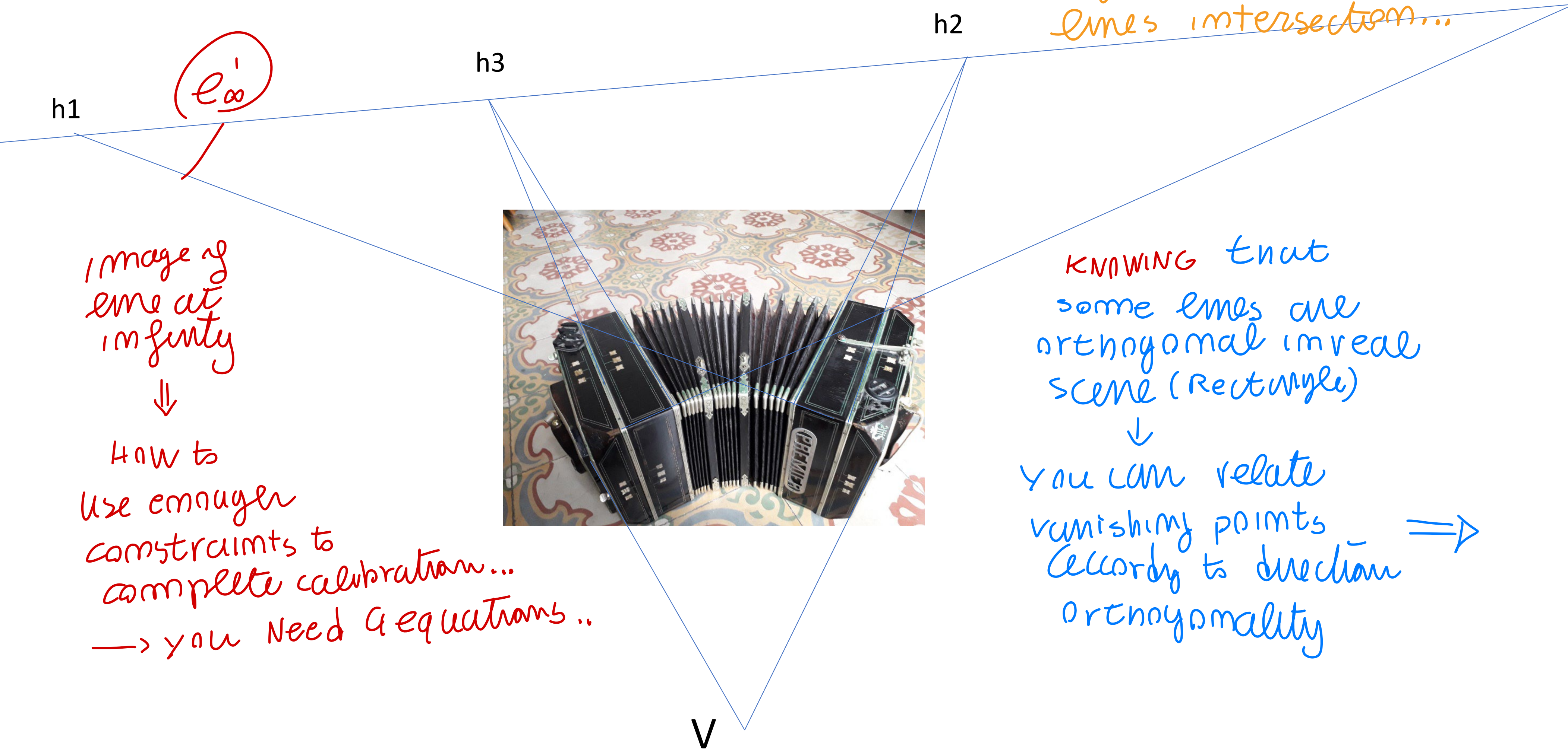
↓ general shape of ω when $f_x \neq f_Y$

aspect ratio

(you wanna find it :

$$\omega = (KK^T)^{-1} = \begin{vmatrix} a^2 & 0 & -u_0 a^2 \\ * & 1 & -v_0 \\ * & * & f_Y^2 + a^2 u_0^2 + v_0^2 \end{vmatrix}$$

→ only $\underbrace{\text{four equations}}_{f_Y,\ f_X,\ u_0,\ v_0}$ are needed to find ω (4 unknowns) → during reconstruction you find horizon lines using images of parallel lines intersection...



ℓ'_∞

image of line at infinity

⇓

How to use enough constraints to complete calibration...

→ you Need 4 equations..

KNOWING that some lines are orthogonal in real scene (Rectangle)

↓

You can relate vanishing points according to direction orthogonality ⟹

Calibration: rectified planar face plus vanishing point of the direction normal to the face

$\Downarrow$ orthogonality

$\cos\theta = \dots = 0$

$\hookleftarrow$ constraint

2 eq horiz

2 vertical

vanishing point

horthogonality $\leftarrow$

$$
\begin{aligned}
h_1^T \omega h_2 &= 0 \\
h_3^T \omega h_4 &= 0 \\
v^T \omega h_1 &= 0 \\
v^T \omega h_2 &= 0
\end{aligned}
$$

$\leftarrow$ vertical vanishing point orthogonal to horizontal ones...

3° and 4° equations are linearly independent,

but there are no further ones (why?)

$v^T \omega h_3 = 0$ is NO more meaningful...

- → solve for ω
- → find $\mathbf{K}$ (by Cholesky factorisation of $\omega^{-1} = \mathbf{K}\mathbf{K}^{\mathrm{T}}$)

$v^T \omega h_3 = 0$ is useless, in fact

$$\begin{cases} v^T \omega h_1 = 0 \\ v^T \omega h_2 = 0 \end{cases}$$

$\uparrow$

$v^T \omega h_3 = 0$ is useless,

being h_1, h_2, h_3 co-linear (all in vanishing line)

$\Downarrow$

$$h_3 = \alpha h_1 + \beta h_2$$

$$v^T \omega \alpha h_1 + v^T \omega \beta h_2 = v^T \omega h_3 = 0 \,!$$

$\uparrow$

it is linear combination of first two! USELESS!

any additional constr. involving new vanish. points belonging to ℓ_∞, is JUST combination of previous one! $\longleftarrow$

Calibration from rectified face plus orthogonal vanishing point

→ that previous method depends on the chosen vanishing point... Result change a bit using different points on the equations ! ⇒

direct method:
independent of the chosen pairs of mutually orthogonal vanishing points

↓ find ω in same way ?

H_R^{-1} := from scene ∼ similar to orig. to image

from

(H_R is the inverse of homography from image to scene)

- the reconstructing homography H_R from given img to rectified img
- the image of the line at the infinity l'_∞
- the vanishing point $\mathbf{v}$ along the direction orthogonal to the face

without chosing individual points...

Calibration from rectified face plus orthogonal vanishing point

as inverse of reconstructy homography (was img to model)

- From plane-to-image homography $\mathbf{H} = [h_1 \quad h_2 \quad h_3]$ inverse of the rectifying homography $\mathbf{H}_R$: $\mathbf{H} = \mathbf{H}_R^{-1}$ →

known, because already reconstructed planar scene!

inverse of reconstr. homography

$r_{\pi 1} \perp r_{\pi 2}$ ⇐ orthogonal vectors of partial rot. matrix are orthog.

same modules of $r_{\pi 1}, r_{\pi 2}$

$$h_1{}^{T} \omega h_2 = 0$$
$$h_1{}^{T} \omega h_1 - h_2{}^{T} \omega h_2 = 0$$

(same as Zhang method)

- from $\mathbf{v}_1^T \omega \mathbf{v} = 0$ and $\mathbf{v}_2^T \omega \mathbf{v} = 0$, and $\boldsymbol{l}'_\infty = \mathbf{v}_1 \times \mathbf{v}_2$ →

ADDITIONALLY: V_1, V_2 vanish points

$$\boldsymbol{l}'_\infty = \omega \mathbf{v} \text{ (2 eqns)}$$

additional equations

Calibration from rectified face plus orthogonal vanishing point

- From plane-to-image homography $\mathbf{H} = [h_1 \quad h_2 \quad h_3]$ inverse of the rectifying homography $\mathbf{H}_R$: $\mathbf{H} = \mathbf{H}_R^{-1}$ →

Vector orthogonal to both is given by VECTOR (CROSS) - PRODUCT

but $v_1 \times v_2 := \ell'_\infty$ is ℓ'_∞ by definition ⇒ therefore $\omega v = \ell'_\infty$ (in modulo equal)

⇑

$\omega v \perp v_1, v_2$

$$\boxed{\begin{gathered} {h_1}^T \omega h_2 = 0 \\ {h_1}^T \omega h_1 - {h_2}^T \omega h_2 = 0 \end{gathered}}$$

(same as Zhang method)

$v_2 \perp (\omega v)$

- from $\underbrace{\mathbf{v}_1^T \omega \mathbf{v}}_{v_1 \perp (\omega v)} = 0$ and $\mathbf{v}_2^T \omega \mathbf{v} = 0$, and $\boldsymbol{l}'_\infty = \mathbf{v}_1 \times \mathbf{v}_2$ →

$$\boxed{\boldsymbol{l}'_\infty = \omega\mathbf{v} \ \text{(2 eqns)}}$$

homogeneous, it needs 2 eqs.

FROM RECONSTRUCTED planar scene you can use also this equation

Calibration from rectified face plus orthogonal vanishing point: alternative to the last 2 eqns

- from $I' = H_R^{-1} I = H_R^{-1} \begin{bmatrix} 1 \\ i \\ 0 \end{bmatrix} = [h_1 \quad h_2 \quad h_3] \begin{bmatrix} 1 \\ i \\ 0 \end{bmatrix} = h_1 + i h_2,$

since $I'^T \omega\, I' = (h_1 + i h_2)^T \omega (h_1 + i h_2) = 0 \rightarrow$

$$\boxed{\begin{array}{c} h_1{}^T \omega h_2 = 0 \\ h_1{}^T \omega h_1 - h_2{}^T \omega h_2 = 0 \end{array}}$$

- for any $\mathbf{x} = [\alpha \quad \beta \quad 0]^{\mathbf{T}} \in \boldsymbol{l}_\infty,\ \mathbf{x}' = H_R^{-1}\mathbf{x} = \alpha h_1 + \beta h_2 \in \boldsymbol{l}'_\infty$ and since $\mathbf{v}$ is a vanishing point of direction $\perp$ to $\mathbf{x}$, is $\mathbf{v}^T \omega\, \mathbf{x} = 0\ \forall(\alpha, \beta)$

$\rightarrow$

$$\boxed{\mathbf{v}^T \omega h_1 = 0, \text{ and } \mathbf{v}^T \omega h_2 = 0}$$

so, once derived ω

↓

apply Cholesky decomposition to find K

$\Rightarrow$ cholesky $(\omega^{-1}) = K$

26/11 preview:

simple-view geometry ...
- how to reconstruct planar scene (rectify) using calibrated camera + images
- how to localize camera wrt planar scene

⇐ we saw many examples of camera calibration...

↳ Some problems important in 3D single view geometry
- RECTIFICATION
- LOCALIZATION

[Homework]

↓

solved based on CALIBRATION + theoretical aspect on Rectification

later ⇒ multi-view geometry : usefull for 3D reconstruction based on multiple image

we have K (calibrated camera)
+ img of scene by that camera + horizon line of plane ⟹ Reconstruct planar scene shape ? ⟹

Rectification of a single calibrated image from vanishing points

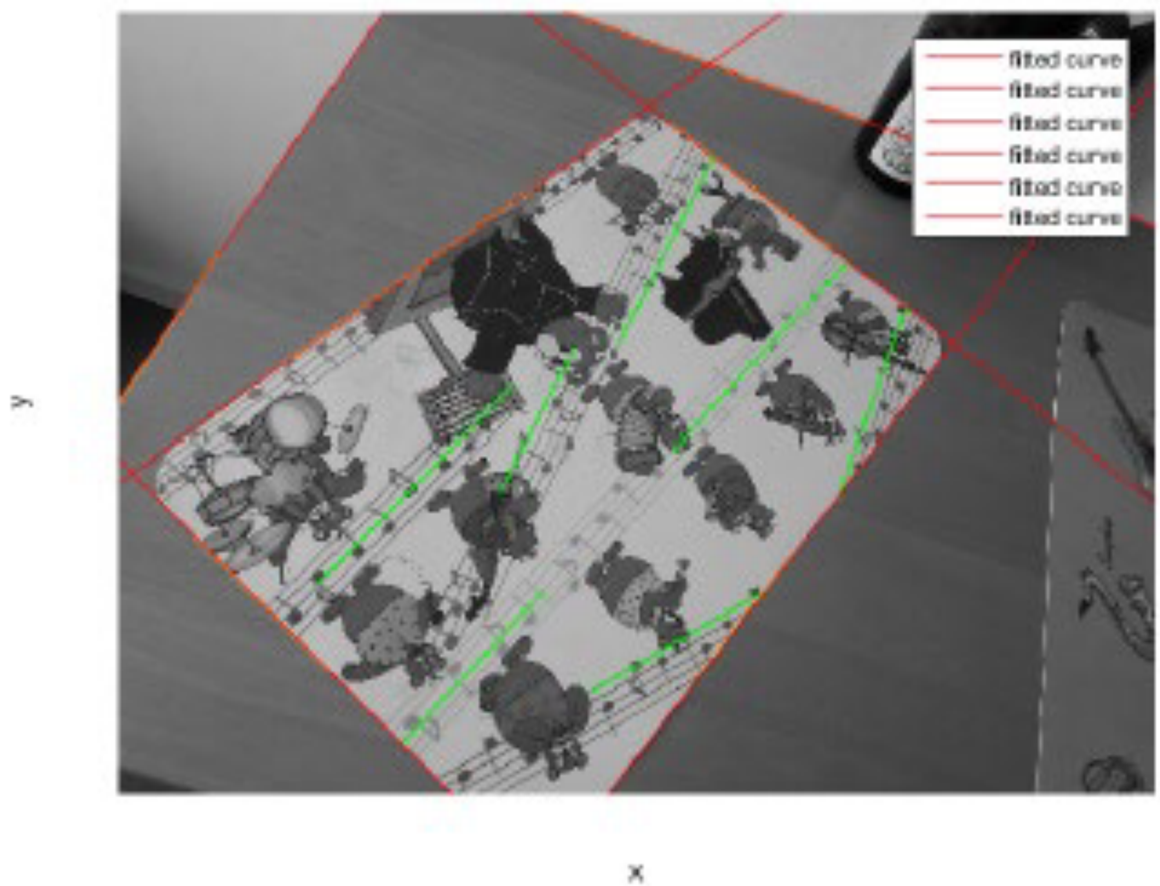
we suppose to know how to calibrate the camera

we have a calibrated camera
+ image of a scene containing planar object

we are able to extract vanishing points ⟹ 2 or more !

then we can easily construct the image of the line at ∞, as line passing through vanishing points this is like one question of the Homework

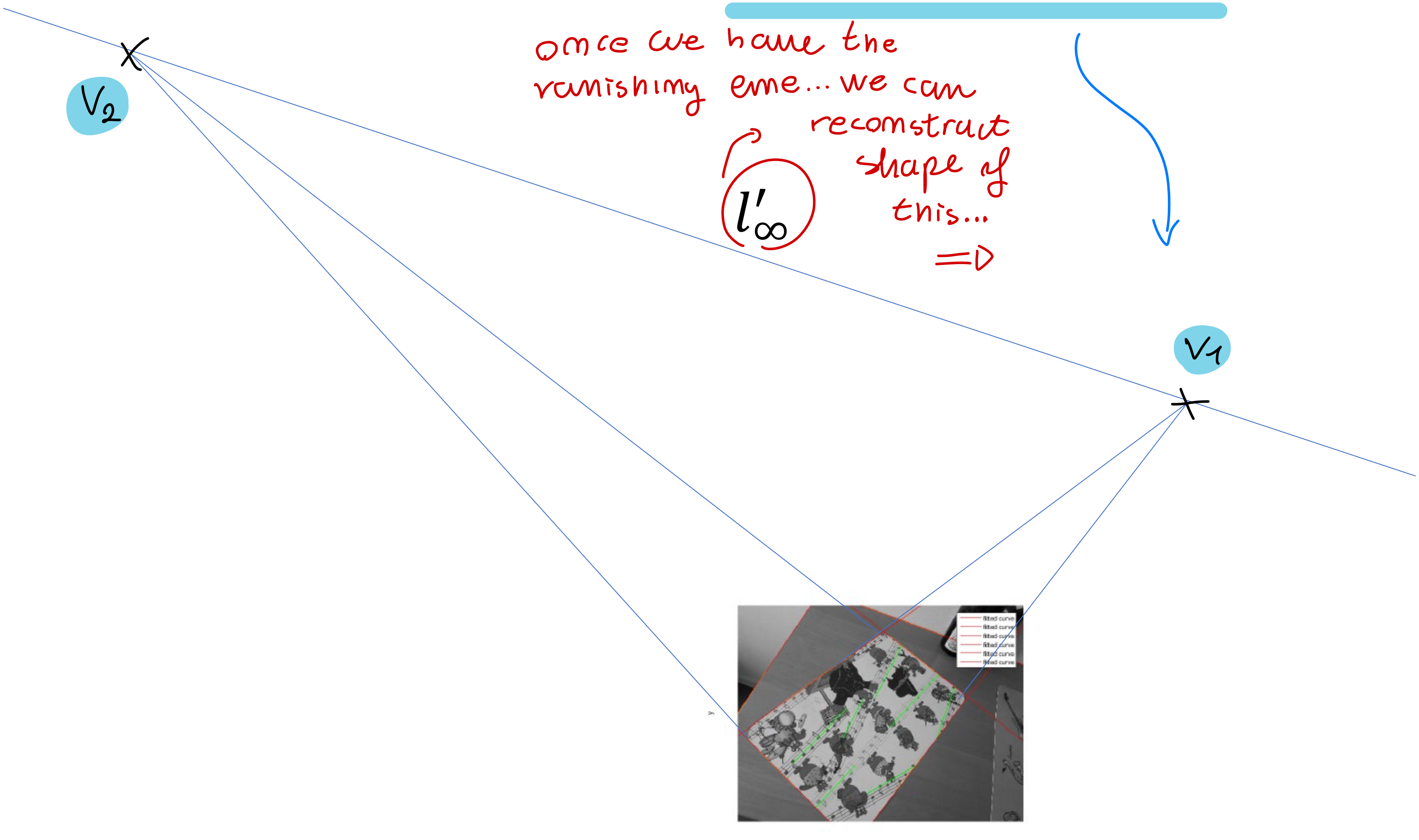
Images of pairs of parallel lines



Vanishing line l'_∞ from vanishing points

once we have the vanishing line... we can reconstruct shape of this... $\Rightarrow$

l'_∞



Unknown shape, known image: → plane-to-image homography $\mathbf{H}$ provides position of points on the plane

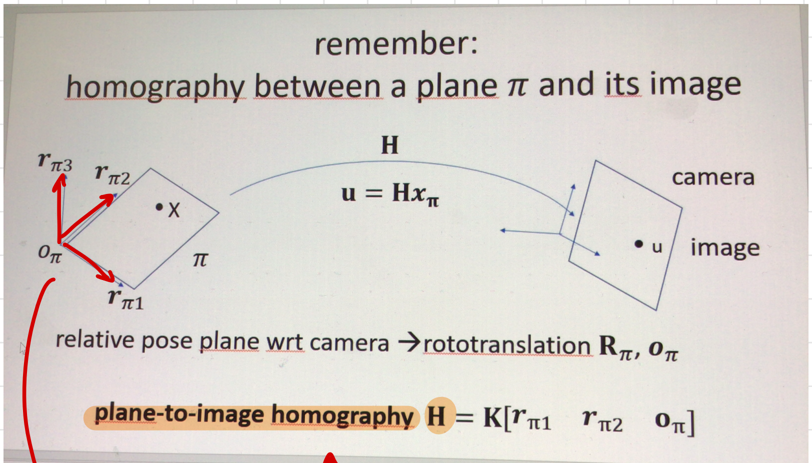
homography from plane to image... the solution of

$\mathbf{H}$?

Rectification is H^{-1} its inverse...

Solution (shape): image-to-plane homography $\mathbf{H}_R = \mathbf{H}^{-1}$ modulo reference frame on the plane and scale factor

exploiting the homography between plane π and image



reference frame attached to π

$x_\pi \in \mathbb{R}^3$ homogeneous coord on plane π

$u = H x_\pi$ image of point of plane π

we have to know relative rotation by $r_{\pi 1}$, $r_{\pi 2}$ and position o_π of plane origin

$\left\{ \text{homography between plane and image} \right\}$

$\Rightarrow$ from this relationship $\Rightarrow$

we need to derive that homography...

Reconstructing homography $\mathbf{H}_R = \mathbf{H}^{-1}$

from single-view image ⇒ we can reconstruct shape (NOT size!)
ambiguity over scale!

plane-to-image homography $\mathbf{H} = \mathbf{K}[\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad \mathbf{o}_{\pi}]$:

$\mathbf{H}$ is related to $\mathbf{K}$ and to the relative pose of plane π wrt camera

we can use information about ℓ'_∞ ⇓ → IF we KNOW K and ℓ'_∞, we can derive the rotation r_{π_i}!

free to choose a «comfortable» reference frame on the plane π:

the only constrained element is the direction $\mathbf{n}_\pi$ normal to plane π

$\mathbf{n}_\pi = \mathbf{K}^T l'_\infty$ (*for proof see next slide, with* $\mathbf{M} = \mathbf{K}$)

→

Choose any $\boldsymbol{r}_{\pi 1} \perp \boldsymbol{r}_{\pi 2}$ both orthogonal to $\mathbf{n}_\pi$ and normalize all of them;

then, take $\mathbf{o}_\pi = \mathbf{n}_\pi$ → in this way $[\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad \mathbf{o}_{\pi}] = \mathbf{R}_\pi$

→

$$\mathbf{H}_R = \mathbf{H}^{-1} = (\mathbf{K}\mathbf{R}_\pi)^{-1} = \mathbf{R}_\pi{}^T \mathbf{K}^{-1}$$

being e'_∞ as img of e_∞ [10 min 26/11] (?)

↓

wrt vanishing point, if we have x_∞ point at infinity

↓

of a plane π

⇓

we know that V1 vanishing point, has viewing ray parallel to direction of it

other vanishing points has viewing ray parallel..

↑

IF camera is calibrated we know that directions relative to camera reference

→ we can extract viewing rays directions,

We know 2 directions // to plane

↓

so plane orientation can be extracted

from π orientation coordinates of other points on the plane can be extracted given image

↓

from orientation, arbitrary distant new plane...
some ray, intersect

↓

and you can reconstruct shape by New image plane parallel to object plane → than affine relation composed by similarity

you can INVERT it!

← geometrically you remove K affine deformation,
rearrange
as square pixels!

- Algebraically you rely on formula $H = K[r_{\pi 1}, r_{\pi 2} \; O_\pi]$

+ you know ℓ'_∞ vanishing line

↓

you can find NORMAL DIRECTION to plane, given ℓ'_∞

$n_\pi = K^T \ell'_\infty$ // normal direction to the plane π

↓

IF you know n_π: it is z axis of reference $:= r_{\pi 3}$

you can compute n_π from K, ℓ'_∞

↓

than any two unit vector orthogonals mutually

$r_{\pi 1} \perp r_{\pi 2}$ and both $r_{\pi 1} \perp n_\pi$
$r_{\pi 2} \perp n_\pi$

↓

reference attached to plane coherent with $r_{\pi 1}, r_{\pi 2}$ as defined before...

$R_\pi = [r_{\pi 1}, \; r_{\pi 2} \; O_\pi]$

↓ than H_R can be extracted!

$= H^{-1} = (R_\pi^T K^{-1})$

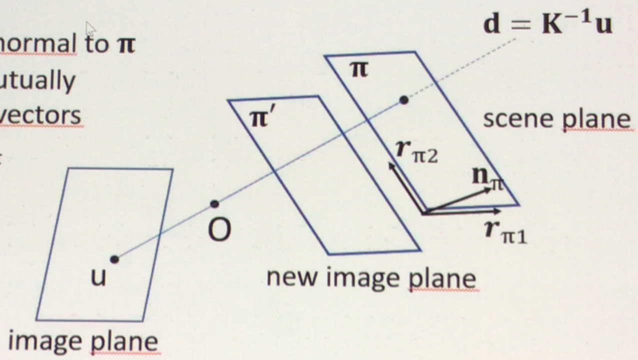
Rectification (reconstruction):

reprojection of the image onto a new image plane $\boldsymbol{\pi}'$ parallel to the scene plane $\boldsymbol{\pi}$

geometric proof

$\mathbf{n}_\pi$: unit vector normal to $\boldsymbol{\pi}$

$\boldsymbol{r}_{\pi 1}$ and $\boldsymbol{r}_{\pi 2}$: mutually orthogonal unit vectors orthogonal to $\mathbf{n}_\pi$



Rectification (reconstruction):

reprojection of the image onto a new image plane $\boldsymbol{\pi}'$ parallel to the scene plane $\boldsymbol{\pi}$

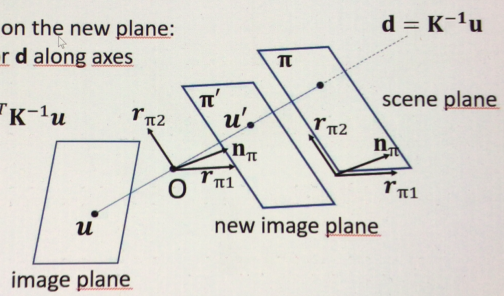
coord.s of the image on the new plane: components of vector **d** along axes

$$\boldsymbol{u}' = \begin{bmatrix} \boldsymbol{r}_{\pi 1}{}^T\mathbf{d} \\ \boldsymbol{r}_{\pi 2}{}^T\mathbf{d} \\ \boldsymbol{n}_\pi{}^T\mathbf{d} \end{bmatrix} = \mathbf{R}_\pi{}^T\mathbf{K}^{-1}\boldsymbol{u}$$

→

$$\mathbf{H}_R = \mathbf{R}_\pi{}^T\mathbf{K}^{-1}$$

same result geometrically



Reconstructing homography $\mathbf{H}_R = \mathbf{H}^{-1}$

plane-to-image homography $\mathbf{H} = \mathbf{K}[\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad \mathbf{o}_{\pi}]$:

$\mathbf{H}$ is related to $\mathbf{K}$ and to the relative pose of plane π wrt camera

→

free to choose a «comfortable» reference frame on the plane π:

the only constrained element is the direction $\mathbf{n}_{\pi}$ normal to plane π

$\mathbf{n}_{\pi} = \mathbf{K}^T l'_{\infty}$ (*for proof see next slide, with* $\mathbf{M} = \mathbf{K}$)

construct new image plane parallel to previous, you take O_π freely

→

Choose any $\boldsymbol{r}_{\pi 1} \perp \boldsymbol{r}_{\pi 2}$ both orthogonal to $\mathbf{n}_{\pi}$ and normalize all of them;

only

then, take $\mathbf{o}_{\pi} = \mathbf{n}_{\pi}$ → in this way $[\boldsymbol{r}_{\pi 1} \quad \boldsymbol{r}_{\pi 2} \quad \mathbf{o}_{\pi}] = \mathbf{R}_{\pi}$

geometric requirements is have new image with img plane parallel

→

$$\mathbf{H}_R = \mathbf{H}^{-1} = (\mathbf{K}\mathbf{R}_{\pi})^{-1} = \mathbf{R}_{\pi}{}^T \mathbf{K}^{-1}$$

DIGRESSION: Back-projection of an image line

↓ reason why the normal to a plane π, given K, l'_∞ is this $M^T l$ express...

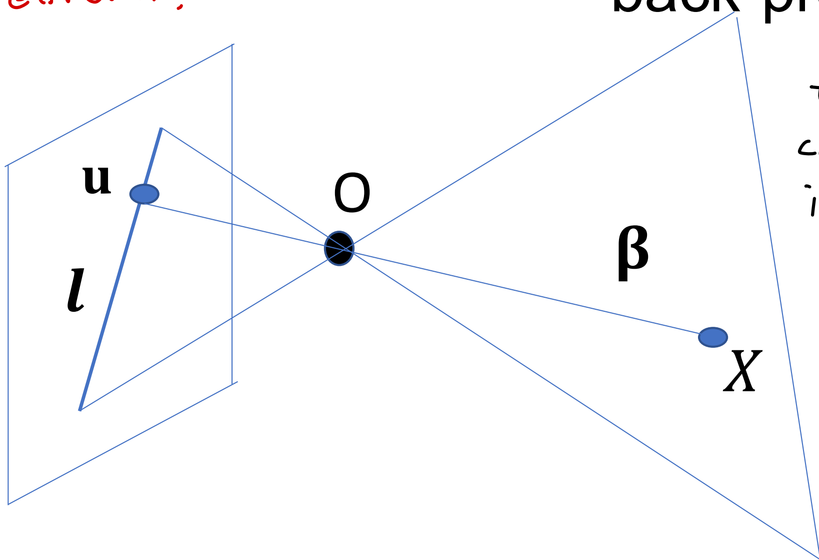
set of space points X, whose image projection $\mathbf{u} = \mathbf{P}X$ is on image line $\boldsymbol{l}$

this rely on vanishing point theorem!

↓

you can describe orientation of the plane..

back-projection



translate constraint in u as X constraint ⟹

point in 3D

$$l^{\mathrm{T}}\mathbf{u} = \underline{l^{\mathrm{T}}\mathbf{P}}X = 0$$

linear homg eq in 3D

$$\underline{\boldsymbol{\beta}^T}\mathrm{X} = 0$$

identify plane

back-projection of $\boldsymbol{l}$: plane $\boldsymbol{\beta} = \mathbf{P}^{\mathrm{T}}\boldsymbol{l}$

through O, in fact, since O $= RNS(\mathbf{P})$

$$\boldsymbol{\beta}^T \mathrm{O} = \boldsymbol{l}^{\mathrm{T}}\mathbf{P}\,\mathrm{O} = \boldsymbol{l}^{\mathrm{T}}0 = 0$$

In our case WORLD = CAMERA reference $M = KR = K$ $(R = I)$

$\simeq K$

back-projection of $\boldsymbol{l}$: plane $\boldsymbol{\pi} = \mathbf{P}^{\mathrm{T}}\boldsymbol{l} = \begin{bmatrix}\mathbf{M}^T \\ \boldsymbol{m}^T\end{bmatrix}\boldsymbol{l} \rightarrow$ normal: $\mathbf{n}_\beta = \mathbf{M}^T\boldsymbol{l}$

when camera = world

↳ so we are able to rectify image of taken by calibrated camera + vanishing line l'_∞ known

2D shape reconstruction = image rectification

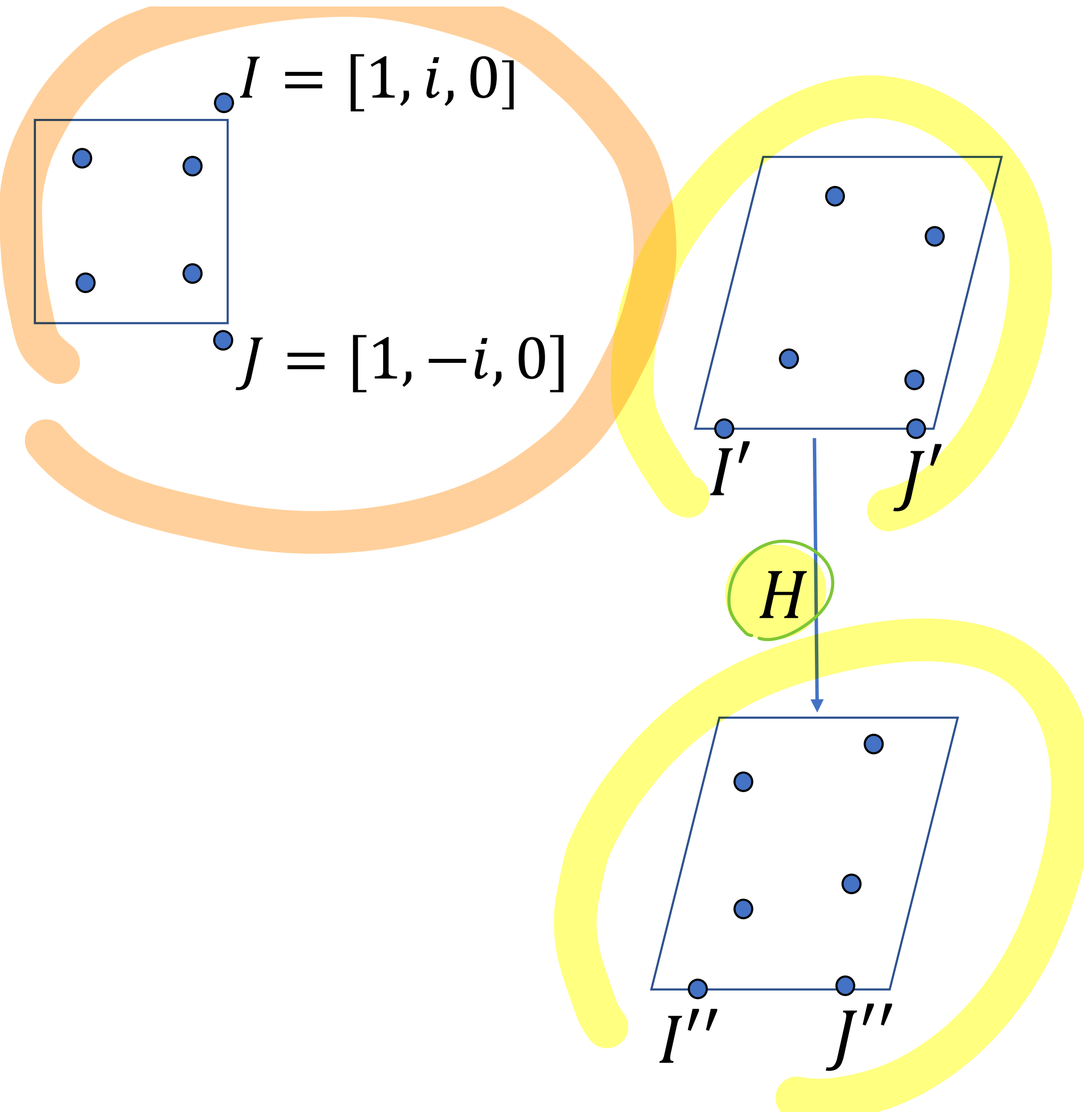
result

you can take MORE IMAGES of same scene!

Reconstruction of an **unknown** planar scene from calibrated images (i.e. images taken by calibrated cameras)

but NOT able to extract vanishing lines! you don't know ℓ'_∞!
previous technique NOT holds

I took images from different viewpoint, by calibrated camera
$\Rightarrow$ rectify image?



reconstruction of an **unkown** planar **scene,** from two **calibrated** images

→ use image-to-image homographies

(we extract 4 points (at least) and we can relate it.

homography between images will be sufficient to reconstruct shape of unkn object

from img to img H:
$\Rightarrow$ REMEMBER: $I'^{T} \omega I' = 0$ for any imaged circular point I'

UNKNOWNS

$$I'^{T} \omega I' = 0$$ Re, Im (2 eqs)

$$I''^{T} \omega I'' = 0$$

$$I'' = H I'$$ KNOWN!

$$I'^{T} H^{T} \omega H I' = 0$$ Re, Im (2 eqs)

KNOWN, from 2 images

KNOWN

solve in 4 unknowns as
$I'_{x,y,w}$
homogeneous
coordinates in 2D
2 coordinate BUT
complex, so 2 × 2 (Re, Im)

Same camera hypothesis can easily be relaxed

$$\omega = \left(\mathbf{K}\mathbf{K}^{\mathrm{T}}\right)^{-1} = \mathbf{K}^{-\mathrm{T}}\mathbf{K}^{-1}$$ is known after calibration

sufficient to reconstruct

2 Re, 2 img..

just 4 unknowns I' complex coordinates → at least 2 images
(each eqn leads to 2 constraints: Re and Im part)

GEOMETRICALLY

conic 1

$$\begin{cases} I'^T \omega' I' = 0 \\ I'^T H'^T \omega'' H' I' = 0 \end{cases}$$

conic 2

two different cameras ω' and ω''

in this case, different cameras considered

corresponds to

Geometrically: intersection of two conics:

ω' and $H'^T \omega'' H'$

→

two resulting pairs of

imaged circular points

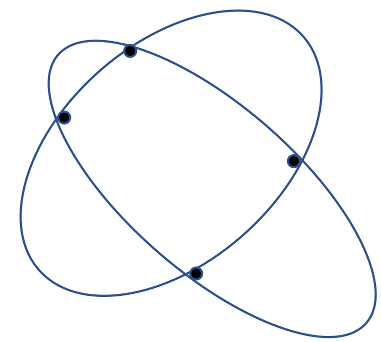
→

selection based on reprojection

or on an additional (third) image

you can use 3 images if needed when NOT enough, as 3 conics intersection

ω'



$H'^T \omega'' H'$

As usual: image rectification from the image

classic approach

$(\boldsymbol{I'}, \boldsymbol{J'})$ of the circular points $(\boldsymbol{I}, \boldsymbol{J})$

- Image of the circular points → image of the conic dual to the circular points

$$C_\infty^{*\prime} = \boldsymbol{I'}\boldsymbol{J'}^{\boldsymbol{T}} + \boldsymbol{J'}\boldsymbol{I'}^{\boldsymbol{T}}$$

- Singular value decomposition

$$\mathrm{svd}(C_\infty^{*\prime}) = U \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix} U^T = H_{SR}^{-1} C_\infty^{*} H_{SR}^{-T}$$

- Rectifying transformation (from svd output U)

using singular value

$$H_{SR} = \begin{bmatrix} 1/\sqrt{a} & 0 & 0 \\ 0 & 1/\sqrt{b} & 0 \\ 0 & 0 & 1 \end{bmatrix} U^T$$

- Euclidean reconstructed model $\boxed{M_S = \underline{H_{SR} \ * \mathrm{Image}}}$ shape reconstruction

How to localize planar shape if is KNOWN, and K is known... where is position/orientation of this planar object

⇒ **Localization** of a known planar shape from a calibrated image

relatively to camera

important in Robotics to understand where is the plane wrt camera

⇑

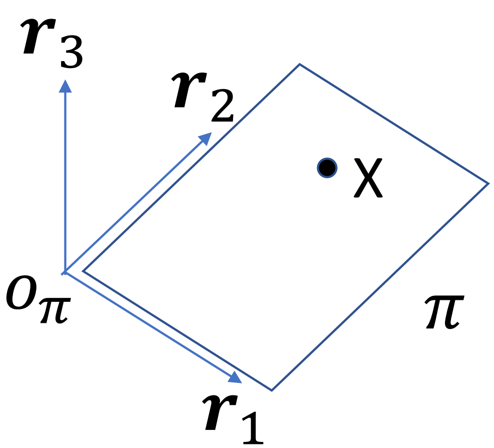
Just a matter of "where is the object wrt camera?"

problem similar to the one of "what is plane-image homography"?

Localization of a known planar shape from a calibrated image

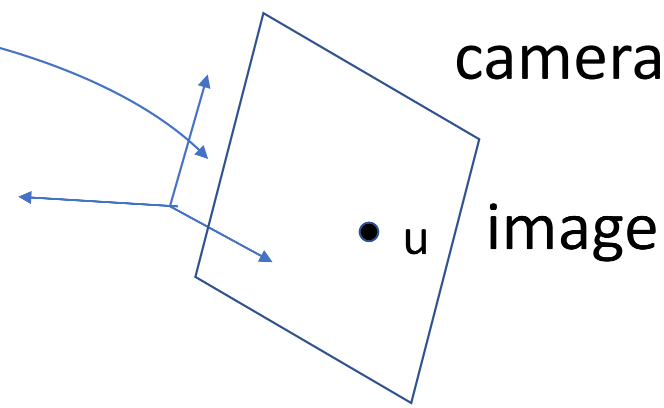
we derive position/orient of π wrt image

Fit homography H between the plane π and its image



H

$$\begin{cases} \mathbf{u}_j = \mathbf{H}\boldsymbol{x}_{\boldsymbol{\pi}j} \\ j = 1\,..\,4 \end{cases}$$



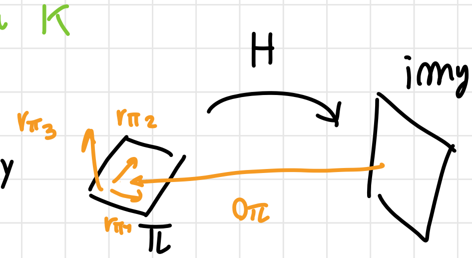
plane π reference: relative pose plane wrt camera → rototranslation $\mathbf{R}_\pi$, $\boldsymbol{o}_\pi$

world reference ≡ camera reference

$$\mathrm{P} = [\mathbf{K} \quad \mathbf{0}]$$

we have pose = position + orientation
of plane π wrt camera O
(relative to)
+ calibrated camera K

we found H as homography
plane to image



$$H = K\,[r_{\pi 1}\ r_{\pi 2}\ O_\pi]$$

$\downarrow$

in localization we have the same problem formula, ($\overbrace{\text{same}}$ phenomenon)
BUT NOW the unknown is different
$\downarrow$
$r_{\pi 1}, r_{\pi 2}, O_\pi$?

localization KNOWN:
- K: calibration ✓
- H: because we have both the image and shape (coord. of the points on plane π)
  NOT size but shape! ✓

we want pose!: $r_{\pi 1}, r_{\pi 2}, O_\pi$? $\Leftarrow$

shape ~ $|K| \chi_\pi$ KNOWN shape

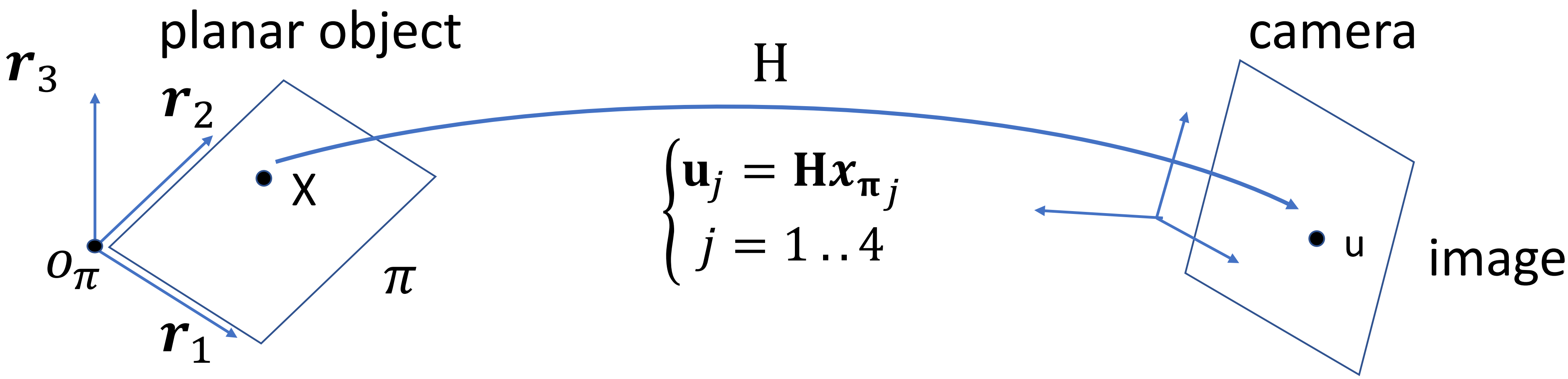
homography with points distance is KNOWN ... we KNOW this homography H

$$\underbrace{H = K\,[r_{\pi 1}\ r_{\pi 2}\ O_\pi]}$$

$\downarrow$

$$[r_{\pi 1}\ r_{\pi 2}\ O_\pi] = K^{-1} H$$

$$r_{\pi 3} = r_{\pi 1} \times r_{\pi 2}$$



plane reference

world reference ≡ camera reference → $\mathrm{P} = [\mathbf{K} \quad \mathbf{0}]$

$$\mathbf{X}_{\boldsymbol{\pi}} = \begin{bmatrix} x \\ y \\ 0 \\ w \end{bmatrix} \qquad \mathbf{X}_{\mathbf{w}} = \begin{bmatrix} \boldsymbol{r}_{\mathbf{1}} & \boldsymbol{r}_{\mathbf{2}} & \boldsymbol{r}_{\mathbf{3}} & \mathbf{o}_{\boldsymbol{\pi}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \\ w \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \mathbf{o}_\pi \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} x \\ y \\ w \end{bmatrix}}_{\boldsymbol{x}_\pi}$$

$$\mathbf{u} = \mathrm{P}\mathbf{X}_{\mathbf{w}} = [\mathbf{K} \quad \mathbf{0}] \begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \mathbf{o}_\pi \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}_{\boldsymbol{\pi}} = \mathbf{K}[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \mathbf{o}_\pi]\boldsymbol{x}_{\boldsymbol{\pi}}$$

homography $\mathbf{H} = \mathbf{K}[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \mathbf{o}_\pi]$ → obj. pose wrt camera $[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \mathbf{o}_\pi] = \mathbf{K}^{-1}\mathrm{H}$

If no accurate in specify distances between points in plane, you get some multiple of true translation!

⇓

you get correct unit vector of axis

BUT instead of correct translation you get an incomplete one

that formula carry out:

⇓

homography $\mathbf{H} = \mathbf{K}[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \mathbf{o}_\pi]$ → obj. pose wrt camera $[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \mathbf{o}_\pi] = \mathbf{K}^{-1}\mathrm{H}$

and

$$\boldsymbol{r}_3 = \boldsymbol{r}_1 \times \boldsymbol{r}_2$$

NOTE: if the coordinates of the points $\boldsymbol{x}_{\pi_j}$ on the plane are known modulo scale, then the translation $\mathbf{o}_\pi$ is only known modulo scale (i.e., its direction is known but its module is unknown)

→

- Shape and image → orientation and viewing direction
- Shape + size and image → position and orientation ← if you have correct size you also have position!

if only shape KNOWN, O_π can be derived, but not its module ...

↑

depends if size correct distances is KNOWN!

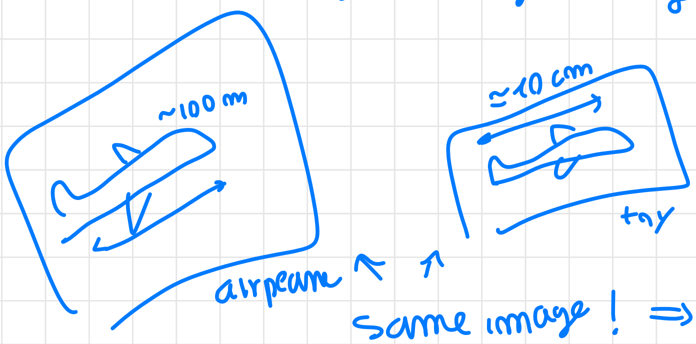
⇑

- in ROBOTICS applications often also size is KNOWN!
- when no size identification, you can rely on orientation + direction

← common fact in vision... when only images

↓

when looking to away objects,
we can have
same image even if small toy object



same image ! ⇒ large object far
~ small object closer

⇓

only image is NOT enough to disambiguate scale

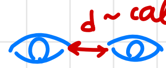
← unless you use multiple-view geometry !

⇓

2-view geometry
allow to solve also size
if some scale is fixed...
like the distance
between two camera...

← like 2 eye view of human

d ~ calibrated distance of eyes! ⇒ $\boxed{\text{IACV - C}}$



NEXT SLIDES !

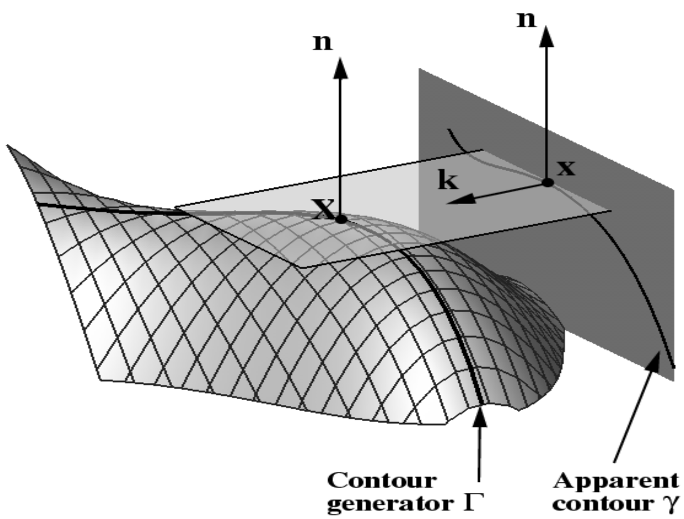
Apparent contour of a smooth surface

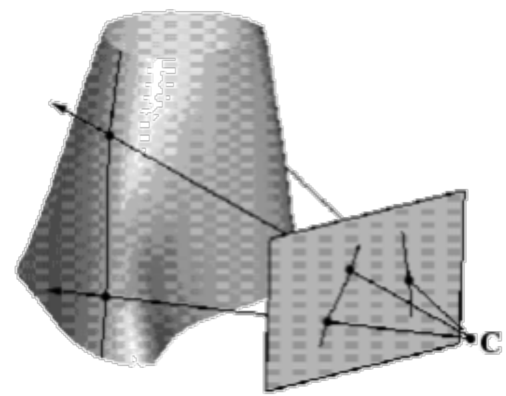
Image projection of smooth surfaces

The **contour generator** Γ is the set of points X on a surface S, whose viewing rays are tangent to S.

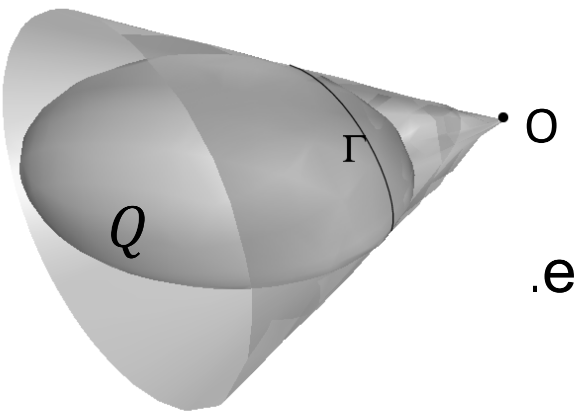
The corresponding **apparent contour** is the image of Γ

Γ depends only on position of projection center O,
Γ depends also on rest of the projection matrix P





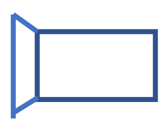
The apparent contour of a quadric



select those planes, tangent to the quadric Q,
i.e. belonging to dual quadric $Q^* = Q^{-1}$
which go through camera center O
.e., that are backprojections $\mathrm{P}^\mathrm{T}\boldsymbol{l}$ of some image lines $\boldsymbol{l}$

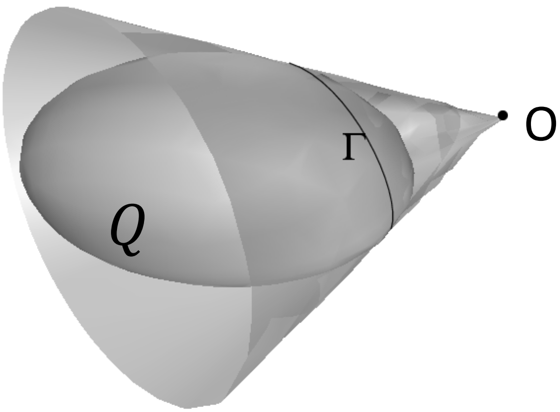
$$\Pi^\mathrm{T}\mathrm{Q}^*\Pi = l^\mathrm{T}\mathrm{P}\mathrm{Q}^*\mathrm{P}^\mathrm{T}l = 0$$

these planes Π are tangent to the contour generator Γ



these lines $\boldsymbol{l}$ are image of planes Π
→ $\boldsymbol{l}$ are tangent to image of Γ
→ $\boldsymbol{l}$ are tangent to apparent contour γ

The apparent contour of a quadric



select those planes, tangent to the quadric Q,
i.e. belonging to dual quadric $Q^* = Q^{-1}$
which go through camera center O
i.e., that are backprojections of some image lines $\boldsymbol{l}$

$$\Pi^{\mathrm{T}} \mathrm{Q}^* \Pi = l^{\mathrm{T}} \mathrm{P} \mathrm{Q}^* \mathrm{P}^{\mathrm{T}} l = 0$$

these image lines $\boldsymbol{l}$ satisfy a quadratic equation
→ they belong to a dual conic $\mathrm{C}^* = \mathrm{PQ}^*\mathrm{P}^T$
→ they are tangent to a conic $\mathrm{C} = {\mathrm{C}^*}^{-1} = (\mathrm{PQ}^*\mathrm{P}^T)^{-1}$

But $\boldsymbol{l}$ are tangent to conic $\mathrm{C} = {\mathrm{C}^*}^{-1} = (\mathrm{PQ}^*\mathrm{P}^T)^{-1}$ → $\boxed{\text{a.c } \gamma \text{ is the conic C}}$

Exercise: apparent contour of a cone?

- contour generator Γ: set of tangency points from O

→ two straight lines through the vertex V

- apparent contour γ : image of Γ

→ two image lines: i.e, a degenerate conic

Example: contour generator of a **right** cone?

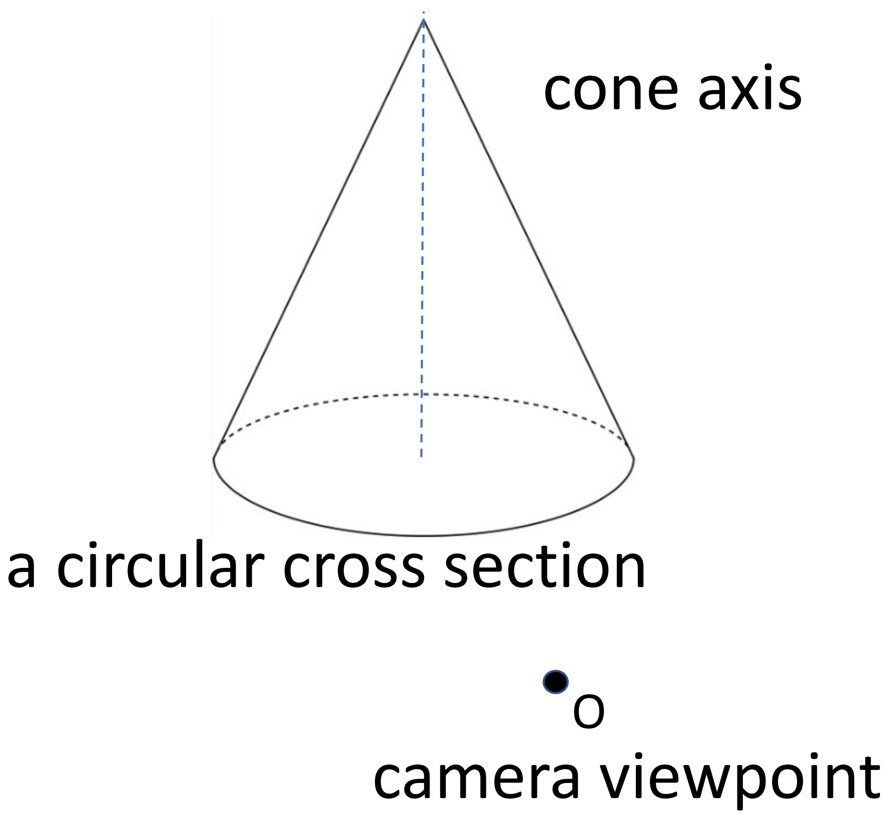
- contour generator Γ: set of tangency points from O

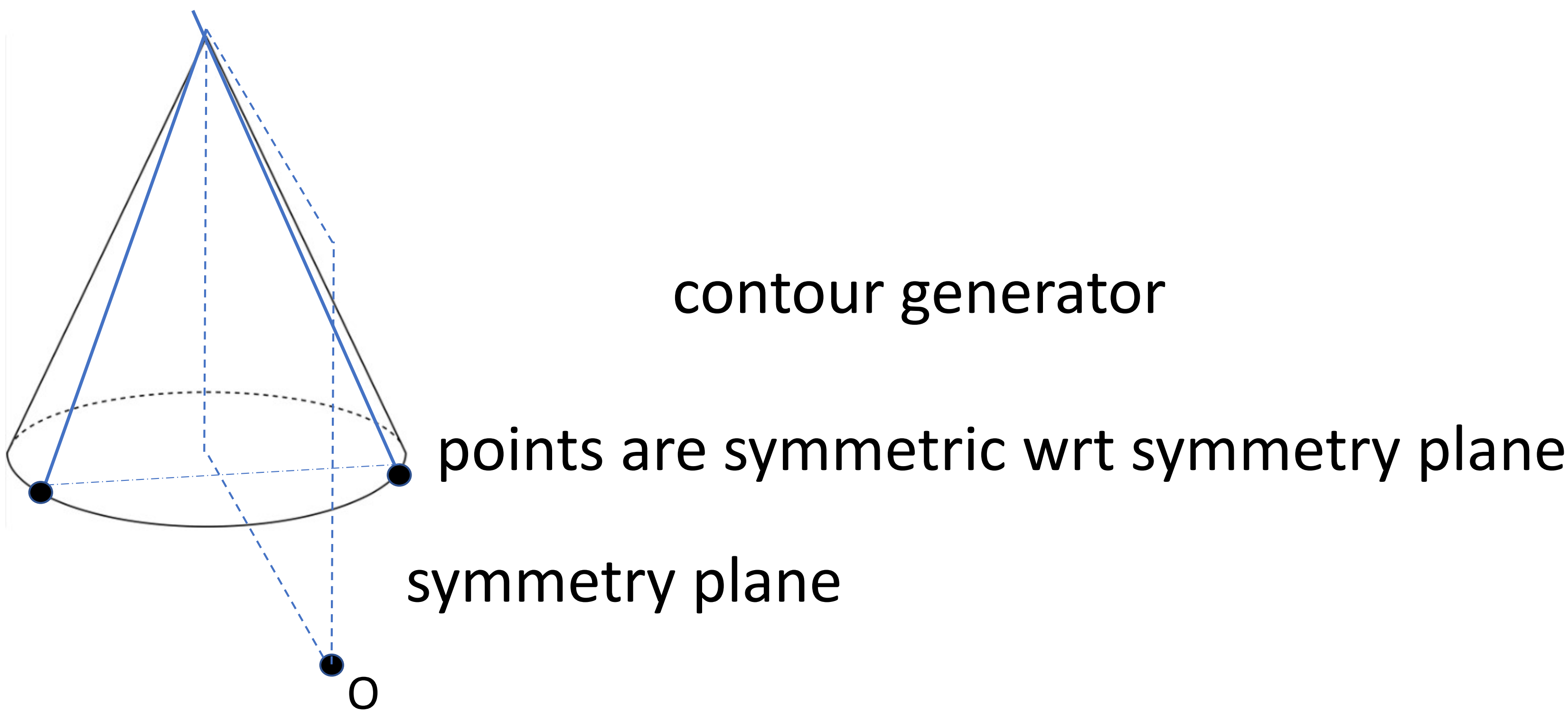
→ two straight lines through the vertex V

- **right** cone alone is **symmetric** wrt to its axis

→ right cone + viewpoint O : «less» symmetry,

which symmetry?


cone axis
a circular cross section
o
camera viewpoint



contour generator
points are symmetric wrt symmetry plane
symmetry plane
O

Example: contour generator of a **right** cone?

- contour generator Γ: set of tangency points from O

→ two straight lines through the vertex V

- **right** cone alone is **symmetric** wrt to its axis

→ right cone + viewpoint O : «less» symmetry,

which symmetry?

PLANAR SYMMETRY wrt plane through axis and O

namely, wrt backprojection plane of the imaged axis