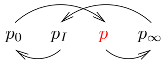
▶1D Projective Coordinates

The 1-D projective coordinate of a point P is defined by the following cross-ratio:

$$[P] = [P_\infty\, P_0\, P_I\, P] = [p_\infty\, p_0\, p_I\, p] = \frac{|\overrightarrow{p_0\, p}|}{|\overrightarrow{p_I\, p_0}|}\, \frac{|\overrightarrow{p_\infty\, p_I}|}{|\overrightarrow{p\, p_\infty}|} = [p]$$

the mnemonic is now '∞'

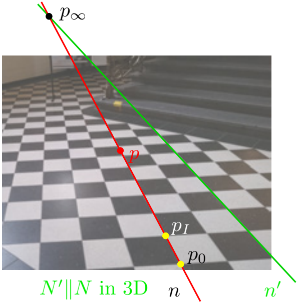


naming convention:

P_0 – the origin	$[P_0] = 0$
P_I – the unit point	$[P_I] = 1$
P_∞ – the supporting point	$[P_\infty] = \pm\infty$

$[P]$ is equal to Euclidean coordinate along N

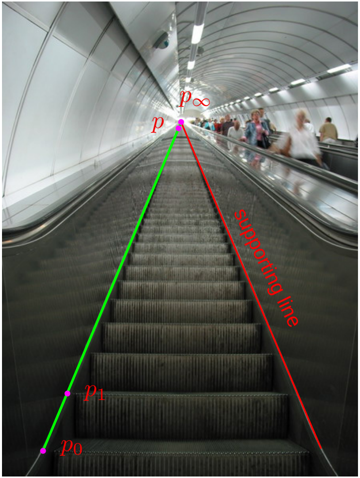
$[p]$ is its measurement in the image plane



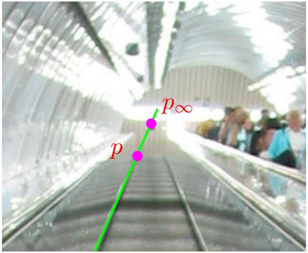
Applications

- Given the image of a 3D line N, the origin, the unit point, and the vanishing point, then the Euclidean coordinate of any point $P \in N$ can be determined →49
- Finding v.p. of a line through a regular object →50

Application: Counting Steps



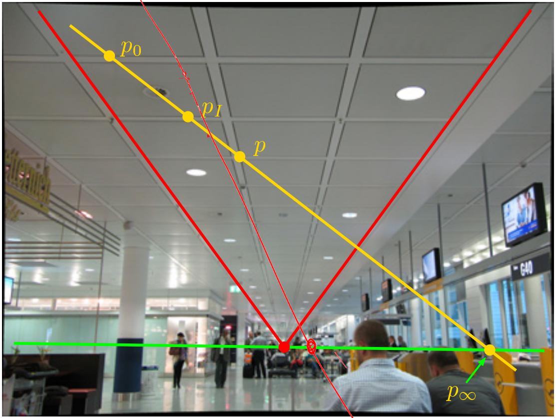
- Namesti Miru underground station in Prague

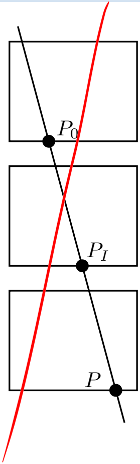


detail around the vanishing point

Result: $[P] = 214$ steps (correct answer is 216 steps) 4Mpx camera

Application: Finding the Horizon from Repetitions





in 3D: $|P_0P| = 2|P_0P_I|$ then [H&Z, p. 218]

$$[P_\infty P_0 P_I P] = \frac{|P_0P|}{|P_0P_I|} = 2 \quad \Rightarrow \quad |p_\infty p_0| = \frac{|p_0p_I| \cdot |p_0p|}{|p_0p| - 2|p_0p_I|}$$

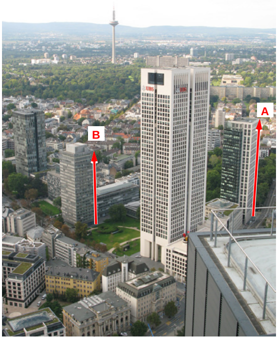
- could be applied to counting steps (→49) if there was no supporting line

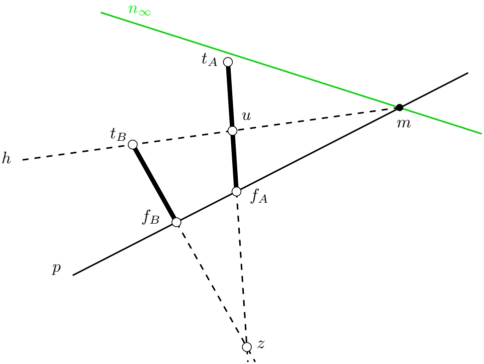
⊛ P1; 1pt: How high is the camera above the floor?

Homework Problem

⊛ H2; 3pt: What is the ratio of heights of Building A to Building B?

- expected: conceptual solution; use notation from this figure
- deadline: LD+2 weeks

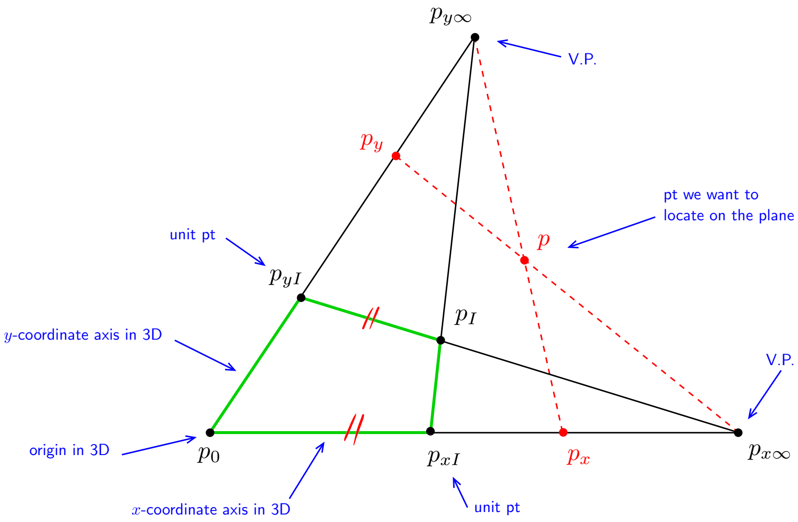




Hints

1. What are the interesting properties of line h connecting the top t_B of Buiding B with the point m at which the horizon intersects the line p joining the foots f_A, f_B of both buildings? [1 point]
2. How do we actually get the horizon n_∞? (we do not see it directly, there are some hills there. . .) [1 point]
3. Give the formula for measuring the length ratio. [formula = 1 point]

2D Projective Coordinates



$$[P_x] = [P_{x\infty}\ P_0\ P_{xI}\ P_x] \qquad [P_y] = [P_{y\infty}\ P_0\ P_{yI}\ P_y]$$

Application: Measuring on the Floor (Wall, etc)

San Giovanni in Laterano, Rome

- measuring distances on the floor in terms of tile units
- what are the dimensions of the seal? Is it circular (assuming square tiles)?
- needs no explicit camera calibration

because we can see the calibrating object (vanishing points)

Part III

Computing with a Single Camera

3.1 Calibration: Internal Camera Parameters from Vanishing Points and Lines

3.2 Camera Resection: Projection Matrix from 6 Known Points

3.3 Exterior Orientation: Camera Rotation and Translation from 3 Known Points

covered by

[1] [H&Z] Secs: 8.6, 7.1, 22.1

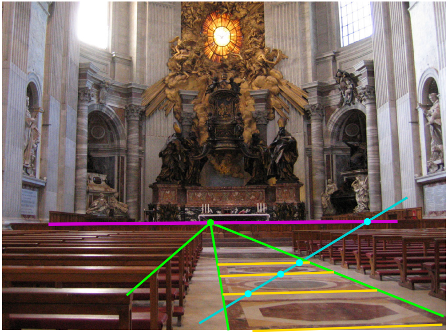
[2] Fischler, M.A. and Bolles, R.C . Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM* 24(6):381–395, 1981

[3] [Golub & van Loan 2013, Sec. 2.5]

Obtaining Vanishing Points and Lines

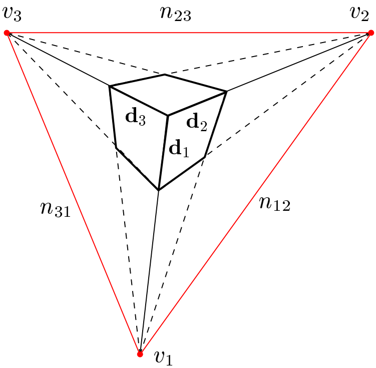
- orthogonal direction pairs can be collected from more images by camera rotation

- vanishing line can be obtained without vanishing points (→50)



▶Camera Calibration from Vanishing Points and Lines

Problem: Given finite vanishing points and/or vanishing lines, compute $\mathbf{K}$



$= \lambda_i Q^{-1} \underline{v}_i$

$$\mathbf{d}_i \simeq \mathbf{Q}^{-1}\underline{\mathbf{v}}_i, \qquad i = 1,2,3 \quad \rightarrow 44$$
$$\mathbf{p}_{ij} \simeq \mathbf{Q}^{\top}\underline{\mathbf{n}}_{ij}, \quad i,j = 1,2,3,\ i \neq j \quad \rightarrow 40 \qquad (2)$$

- simple method: solve (2) after eliminating nuisance pars.

Special Configurations

1. orthogonal rays $\mathbf{d}_1 \perp \mathbf{d}_2$ in space then

$$0 = \mathbf{d}_1^{\top}\mathbf{d}_2 = \underline{\mathbf{v}}_1^{\top}\mathbf{Q}^{-\top}\mathbf{Q}^{-1}\underline{\mathbf{v}}_2 = \underline{\mathbf{v}}_1^{\top}\underbrace{(\mathbf{K}\mathbf{K}^{\top})^{-1}}_{\boldsymbol{\omega}\ \text{(IAC)}}\underline{\mathbf{v}}_2$$

$(QQ^T)^{-1} = (KRR^TK^T)^{-1} = (KK^T)^{-1}$

2. orthogonal planes $\mathbf{p}_{ij} \perp \mathbf{p}_{ik}$ in space

$$0 = \mathbf{p}_{ij}^{\top}\mathbf{p}_{ik} = \underline{\mathbf{n}}_{ij}^{\top}\mathbf{Q}\mathbf{Q}^{\top}\underline{\mathbf{n}}_{ik} = \underline{\mathbf{n}}_{ij}^{\top}\boldsymbol{\omega}^{-1}\underline{\mathbf{n}}_{ik}$$

3. orthogonal ray and plane $\mathbf{d}_k \parallel \mathbf{p}_{ij}$, $k \neq i,j$ — normal parallel to optical ray

$$\mathbf{p}_{ij} \simeq \mathbf{d}_k \quad \Rightarrow \quad \mathbf{Q}^{\top}\underline{\mathbf{n}}_{ij} = \lambda\mathbf{Q}^{-1}\underline{\mathbf{v}}_k \quad \Rightarrow \quad \underline{\mathbf{n}}_{ij} = \lambda\mathbf{Q}^{-\top}\mathbf{Q}^{-1}\underline{\mathbf{v}}_k = \lambda\boldsymbol{\omega}\,\underline{\mathbf{v}}_k, \qquad \lambda \neq 0$$

- n_{ij} may be constructed from non-orthogonal v_i and v_j, e.g. using the cross-ratio
- $\boldsymbol{\omega}$ is a symmetric, positive definite 3×3 matrix — IAC = Image of Absolute Conic

►cont'd

	configuration	equation	# constraints
(3)	orthogonal v.p.	$\underline{\mathbf{v}}_i^\top \boldsymbol{\omega}\, \underline{\mathbf{v}}_j = 0$	1
(4)	orthogonal v.l.	$\underline{\mathbf{n}}_{ij}^\top \boldsymbol{\omega}^{-1} \underline{\mathbf{n}}_{ik} = 0$	1
(5)	v.p. orthogonal to v.l.	$\underline{\mathbf{n}}_{ij} = \lambda \boldsymbol{\omega}\, \underline{\mathbf{v}}_k$	2
(6)	orthogonal raster $\theta = \pi/2$	$\omega_{12} = \omega_{21} = 0$	1
(7)	unit aspect $a = 1$ when $\theta = \pi/2$	$\omega_{11} - \omega_{22} = 0$	1
(8)	known principal point $u_0 = v_0 = 0$	$\omega_{13} = \omega_{31} = \omega_{23} = \omega_{32} = 0$	2

w = vec(ω)

- these are homogeneous linear equations for the 5 parameters in $\boldsymbol{\omega}$ in the form $\mathbf{D}\mathbf{w} = \mathbf{0}$

  λ can be eliminated from (5)
- we need at least 5 constraints for full $\boldsymbol{\omega}$ — symmetric 3×3
- we get $\mathbf{K}$ from $\boldsymbol{\omega}^{-1} = \mathbf{K}\mathbf{K}^\top$ by Choleski decomposition K = chol(ω)

  the decomposition returns a positive definite upper triangular matrix

  one avoids solving an explicit set of quadratic equations for the parameters in $\mathbf{K}$
- unlike in the naive method (2), we can introduce constraints on $\mathbf{K}$, e.g. (6)–(8)

Examples

Assuming orthogonal raster, unit aspect (ORUA): $\theta = \pi/2$, $a = 1$

$$\boldsymbol{\omega} \simeq \begin{bmatrix} 1 & 0 & -u_0 \\ 0 & 1 & -v_0 \\ -u_0 & -v_0 & f^2 + u_0^2 + v_0^2 \end{bmatrix}$$

Ex 1:

Assuming ORUA and known $m_0 = (u_0, v_0)$, two finite orthogonal vanishing points give f

$$\underline{\mathbf{v}}_1^\top \boldsymbol{\omega}\, \underline{\mathbf{v}}_2 = 0 \quad \Rightarrow \quad f^2 = \left|(\mathbf{v}_1 - \mathbf{m}_0)^\top (\mathbf{v}_2 - \mathbf{m}_0)\right|$$

in this formula, $\mathbf{v}_i$, $\mathbf{m}_0$ are not homogeneous!

Ex 2:

Non-orthogonal vanishing points $\mathbf{v}_i$, $\mathbf{v}_j$, known angle ϕ: $\cos\phi = \dfrac{\underline{\mathbf{v}}_i^\top \boldsymbol{\omega} \underline{\mathbf{v}}_j}{\sqrt{\underline{\mathbf{v}}_i^\top \boldsymbol{\omega} \underline{\mathbf{v}}_i}\sqrt{\underline{\mathbf{v}}_j^\top \boldsymbol{\omega} \underline{\mathbf{v}}_j}}$

- leads to polynomial equations
- e.g. ORUA and $u_0 = v_0 = 0$ gives

$$(f^2 + \mathbf{v}_i^\top \mathbf{v}_j)^2 = (f^2 + \|\mathbf{v}_i\|^2) \cdot (f^2 + \|\mathbf{v}_j\|^2) \cdot \cos^2\phi$$

Image of Absolute Conic

This is the **K** matrix:

```
K = {{f, s, u0}, {0, a*f, v0}, {0, 0, 1}}
```

$$\begin{pmatrix} f & s & u_0 \\ 0 & a\,f & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

The ω matrix:

```
ω = Inverse[K.Transpose[K]] * Det[K]^2 // Simplify
```

$$\begin{pmatrix} a^2 f^2 & -a\,f\,s & a\,f\,(s\,v_0 - a\,f\,u_0) \\ -a\,f\,s & f^2 + s^2 & a\,f\,s\,u_0 - (f^2 + s^2)\,v_0 \\ a\,f\,(s\,v_0 - a\,f\,u_0) & a\,f\,s\,u_0 - (f^2 + s^2)\,v_0 & a^2 f^4 + a^2 u_0^2 f^2 - 2\,a\,s\,u_0\,v_0\,f + (f^2 + s^2)\,v_0^2 \end{pmatrix}$$

The ω matrix with no skew:

```
ω / f^2 /. s -> 0 // Simplify // MatrixForm
```

$$\begin{pmatrix} a^2 & 0 & -a^2 u_0 \\ 0 & 1 & -v_0 \\ -a^2 u_0 & -v_0 & a^2 f^2 + a^2 u_0^2 + v_0^2 \end{pmatrix}$$

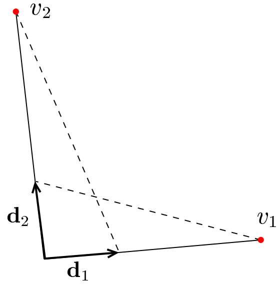
ORUA

```
ω / f^2 /. {a -> 1, s -> 0} // Simplify
```

$$\begin{pmatrix} 1 & 0 & -u_0 \\ 0 & 1 & -v_0 \\ -u_0 & -v_0 & f^2 + u_0^2 + v_0^2 \end{pmatrix}$$

▶Camera Orientation from Two Finite Vanishing Points

Problem: Given $\mathbf{K}$ and two vanishing points corresponding to two known orthogonal directions $\mathbf{d}_1$, $\mathbf{d}_2$, compute camera orientation $\mathbf{R}$ with respect to the plane.



- 3D coordinate system choice, e.g.:

$$\mathbf{d}_1 = (1,0,0), \quad \mathbf{d}_2 = (0,1,0)$$

- we know that

$$\mathbf{d}_i \simeq \mathbf{Q}^{-1}\underline{\mathbf{v}}_i = (\mathbf{K}\mathbf{R})^{-1}\underline{\mathbf{v}}_i = \mathbf{R}^{-1}\underbrace{\mathbf{K}^{-1}\underline{\mathbf{v}}_i}_{\underline{\mathbf{w}}_i}$$

$$\mathbf{R}\mathbf{d}_i \simeq \underline{\mathbf{w}}_i$$

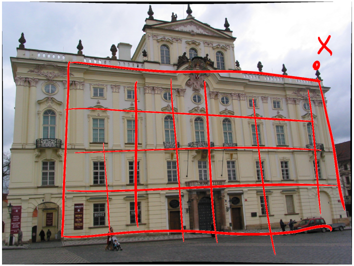
- knowing $\mathbf{d}_{1,2}$ we conclude that $\underline{\mathbf{w}}_i/\|\underline{\mathbf{w}}_i\|$ is the i-th column $\mathbf{r}_i$ of $\mathbf{R}$
- the third column is orthogonal: $\mathbf{r}_3 \simeq \mathbf{r}_1 \times \mathbf{r}_2$

$$\mathbf{R} = \begin{bmatrix} \frac{\underline{\mathbf{w}}_1}{\|\underline{\mathbf{w}}_1\|} & \frac{\underline{\mathbf{w}}_2}{\|\underline{\mathbf{w}}_2\|} & \frac{\underline{\mathbf{w}}_1 \times \underline{\mathbf{w}}_2}{\|\underline{\mathbf{w}}_1 \times \underline{\mathbf{w}}_2\|} \end{bmatrix}$$

some suitable scenes

Application: Planar Rectification

Principle: Rotate camera parallel to the plane of interest.



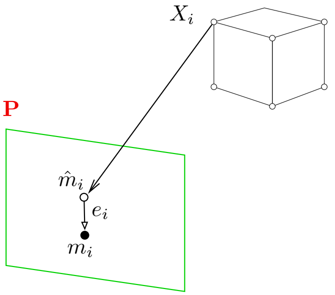
$$\underline{\mathbf{m}} \simeq \mathbf{K}\mathbf{R}\,[\mathbf{I} \quad -\mathbf{C}]\,\underline{\mathbf{X}} \qquad\qquad \underline{\mathbf{m}}' \simeq \mathbf{K}\,[\mathbf{I} \quad -\mathbf{C}]\,\underline{\mathbf{X}}$$

$$\underline{\mathbf{m}}' \simeq \mathbf{K}(\mathbf{K}\mathbf{R})^{-1}\,\underline{\mathbf{m}} = \mathbf{K}\mathbf{R}^\top\mathbf{K}^{-1}\,\underline{\mathbf{m}} = \mathbf{H}\,\underline{\mathbf{m}}$$

- $\mathbf{H}$ is the rectifying homography
- both $\mathbf{K}$ and $\mathbf{R}$ can be calibrated from two finite vanishing points assuming ORUA →58
- not possible when one (or both) of them are infinite
- without ORUA we would need 4 additional views to calibrate $\mathbf{K}$ as on →55

▶Camera Resection

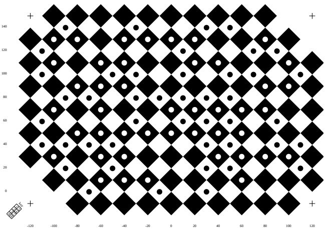
Camera calibration and orientation from a known set of $k \geq 6$ reference points and their images $\{(X_i, m_i)\}_{i=1}^{6}$.



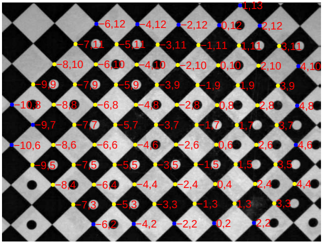
- X_i are considered exact
- m_i is a measurement subject to detection error

$$\mathbf{m}_i = \hat{\mathbf{m}}_i + \mathbf{e}_i \qquad \text{Cartesian}$$

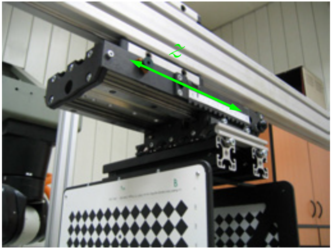
- where $\underline{\hat{\mathbf{m}}}_i \simeq \mathbf{P}\underline{\mathbf{X}}_i$



calibration chart



automatic calibration point detection



resection target with translation stage

- target translated at least once
- by a calibrated (known) translation
- X_i point locations looked up in a table based on their code

►The Minimal Problem for Camera Resection

Problem: Given $k = 6$ corresponding pairs $\{(X_i, m_i)\}_{i=1}^{k}$, find $\mathbf{P}$

$$\lambda_i \underline{\mathbf{m}}_i = \mathbf{P}\underline{\mathbf{X}}_i, \qquad \mathbf{P} = \begin{bmatrix} \mathbf{q}_1^\top & q_{14} \\ \mathbf{q}_2^\top & q_{24} \\ \mathbf{q}_3^\top & q_{34} \end{bmatrix}$$

$$\underline{\mathbf{X}}_i = (x_i, y_i, z_i, 1), \quad i = 1, 2, \ldots, k, \ k = 6$$
$$\underline{\mathbf{m}}_i = (u_i, v_i, 1), \quad \lambda_i \in \mathbb{R}, \ \lambda_i \neq 0$$

easy to modify for infinite points X_i but be aware of →66

expanded: $\lambda_i u_i = \mathbf{q}_1^\top \mathbf{X}_i + q_{14}, \quad \lambda_i v_i = \mathbf{q}_2^\top \mathbf{X}_i + q_{24}, \quad \lambda_i = \mathbf{q}_3^\top \mathbf{X}_i + q_{34}$

after elimination of λ_i: $(\mathbf{q}_3^\top \mathbf{X}_i + q_{34})u_i = \mathbf{q}_1^\top \mathbf{X}_i + q_{14}, \quad (\mathbf{q}_3^\top \mathbf{X}_i + q_{34})v_i = \mathbf{q}_2^\top \mathbf{X}_i + q_{24}$

Then

SVD(A) = UDV^T

$$\mathbf{A}\,\mathbf{q} = \begin{bmatrix} \mathbf{X}_1^\top & 1 & \mathbf{0}^\top & 0 & -u_1\mathbf{X}_1^\top & -u_1 \\ \mathbf{0}^\top & 0 & \mathbf{X}_1^\top & 1 & -v_1\mathbf{X}_1^\top & -v_1 \\ \vdots & & & & & \vdots \\ \mathbf{X}_k^\top & 1 & \mathbf{0}^\top & 0 & -u_k\mathbf{X}_k^\top & -u_k \\ \mathbf{0}^\top & 0 & \mathbf{X}_k^\top & 1 & -v_k\mathbf{X}_k^\top & -v_k \end{bmatrix} \cdot \begin{bmatrix} \mathbf{q}_1 \\ q_{14} \\ \mathbf{q}_2 \\ q_{24} \\ \mathbf{q}_3 \\ q_{34} \end{bmatrix} = \mathbf{0} \tag{9}$$

vec(P)

- we need 11 indepedent parameters for $\mathbf{P}$
- $\mathbf{A} \in \mathbb{R}^{2k,12}, \ \mathbf{q} \in \mathbb{R}^{12}$
- 6 points in a general position give $\operatorname{rank} \mathbf{A} = 12$ and there is no non-trivial null space
- drop one row to get rank 11 matrix, then the basis vector of the null space of $\mathbf{A}$ gives $\mathbf{q}$

▶The Jack-Knife Solution for $k = 6$

- given the 6 correspondences, we have 12 equations for the 11 parameters
- can we use all the information present in the 6 points?

Jack-knife estimation

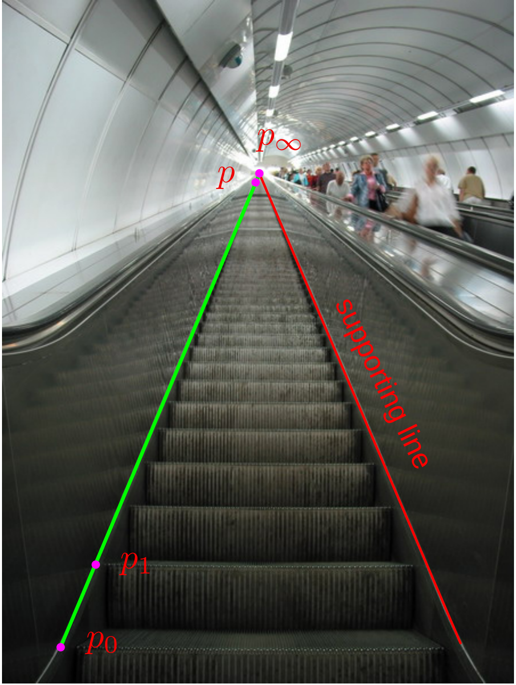
1. $n := 0$
2. for $i = 1, 2, \ldots, 2k$ do
   a) delete i-th row from $\mathbf{A}$, this gives $\mathbf{A}_i$
   b) if $\dim \operatorname{null} \mathbf{A}_i > 1$ continue with the next i
   c) $n := n + 1$
   d) compute the right null-space $\mathbf{q}_i$ of $\mathbf{A}_i$ — e.g. by 'economy-size' SVD
   e) $\hat{\mathbf{q}}_i := \mathbf{q}_i$ normalized to $q_{34} = 1$ and dimension-reduced — assuming finite cam. with $P_{3,4} = 1$
3. from all n vectors $\hat{\mathbf{q}}_i$ collected in Step 1d compute

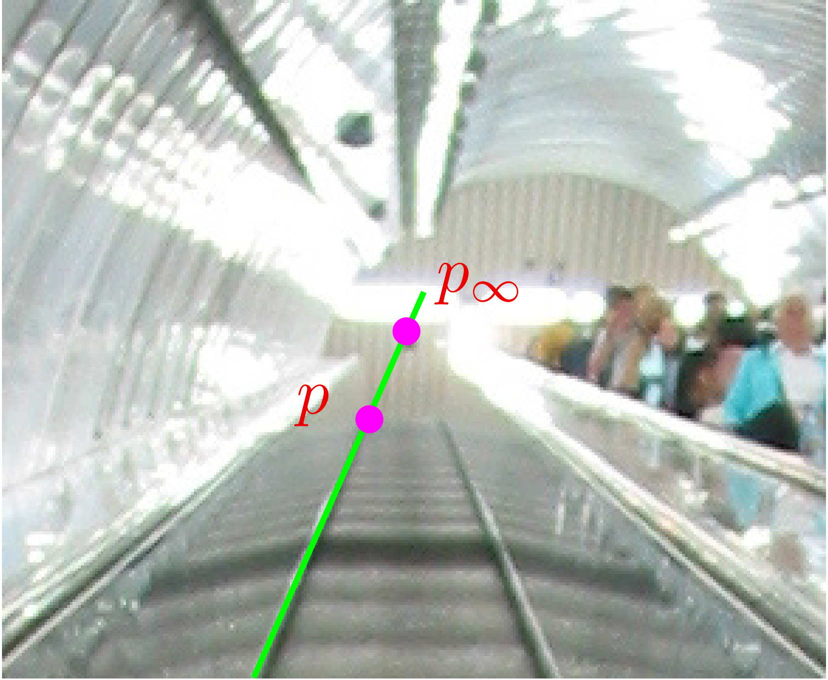
$$\mathbf{q} = \frac{1}{n}\sum_{i=1}^{n} \hat{\mathbf{q}}_i, \qquad \operatorname{var}[\mathbf{q}] = \frac{n-1}{n} \operatorname{diag} \sum_{i-1}^{n} (\hat{\mathbf{q}}_i - \mathbf{q})(\hat{\mathbf{q}}_i - \mathbf{q})^\top$$

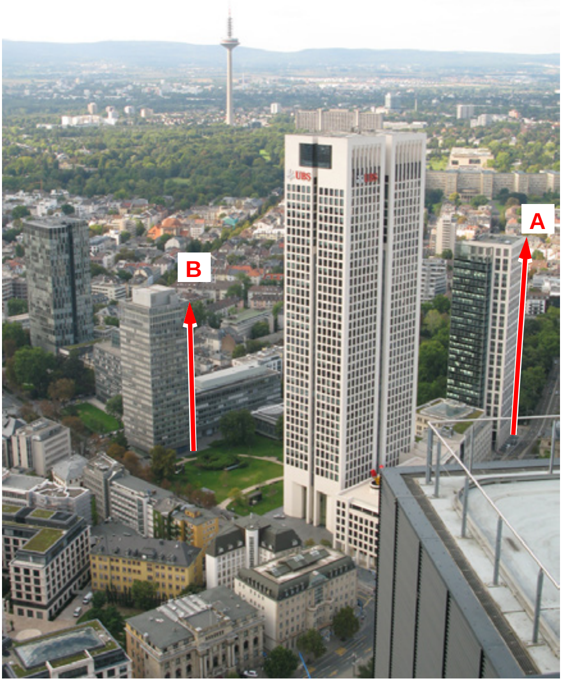
regular for $n \geq 11$

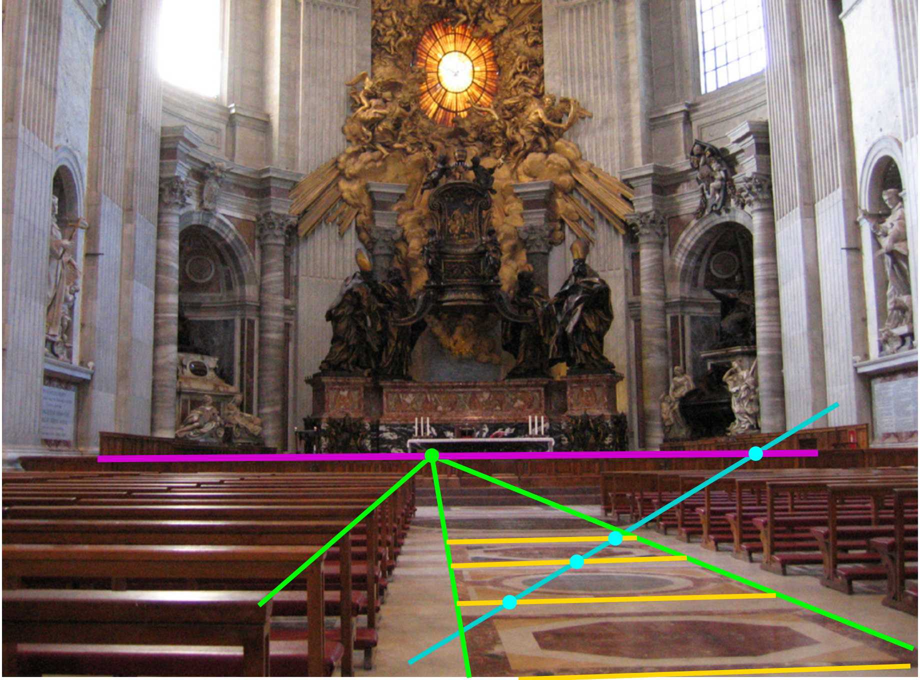
- have a solution + an error estimate, per individual elements of $\mathbf{P}$ (except P_{34})
- at least 5 points must be in a general position (→66)
- large error indicates near degeneracy
- computation not efficient with $k > 6$ points, needs $\binom{2k}{11}$ draws, e.g. $k = 7 \Rightarrow 364$ draws
- better error estimation method: decompose $\mathbf{P}_i$ to $\mathbf{K}_i$, $\mathbf{R}_i$, $\mathbf{t}_i$ (→34), represent $\mathbf{R}_i$ with 3 parameters (e.g. Euler angles, or in Cayley representation →137) and compute the errors for the parameters

Thank You


p_∞
p
supporting line
p_1
p_0

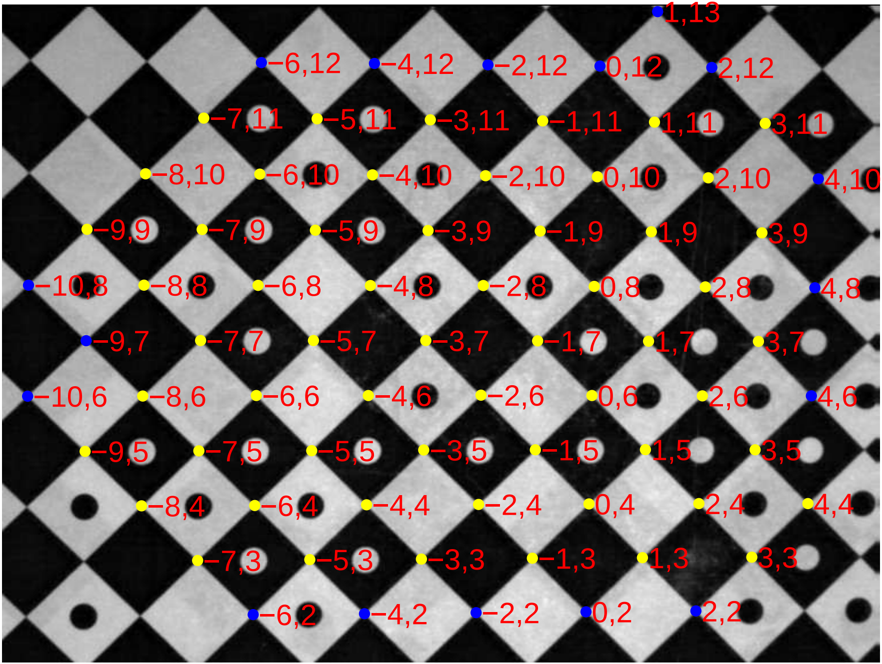

p_∞
p


A
B
UBS




CLEMENS·XII·PONT·MAX·ANN·SAC·MDCCXXXII PONT·VIII·


BOOKS
FILMS
TICKETS
BOOKS


1,13
−6,12
−4,12
−2,12
0,12
2,12
−7,11
−5,11
−3,11
−1,11
1,11
3,11
−8,10
−6,10
−4,10
−2,10
0,10
2,10
4,10
−9,9
−7,9
−5,9
−3,9
−1,9
1,9
3,9
−10,8
−8,8
−6,8
−4,8
−2,8
0,8
2,8
4,8
−9,7
−7,7
−5,7
−3,7
−1,7
1,7
3,7
−10,6
−8,6
−6,6
−4,6
−2,6
0,6
2,6
4,6
−9,5
−7,5
−5,5
−3,5
−1,5
1,5
3,5
−8,4
−6,4
−4,4
−2,4
0,4
2,4
4,4
−7,3
−5,3
−3,3
−1,3
1,3
3,3
−6,2
−4,2
−2,2
0,2
2,2