NUS National University of Singapore | School of Computing

CS4277 / CS5477
3D Computer Vision

Lecture 7:
Absolute Pose Estimation from Points or Lines

Asst. Prof. Lee Gim Hee

AY 2019/20

Semester 2

Course Schedule

Week	Date	Topic	Assignments
1	15 Jan	2D and 1D projective geometry	
2	22 Jan	Circular points and 3D projective geometry	
3	29 Jan	No Lecture	
4	05 Feb	Absolute conic and robust homography estimation	**Assignment 1**: Panoramic stitching (15%)
5	12 Feb	Camera models and calibration	
6	19 Feb	Single view metrology	**Due:** Assignment 1 **Assignment 2**: Camera calibration (15%)
-	26 Feb	Semester Break	No lecture
7	04 Mar	The fundamental and essential matrices	**Due:** Assignment 2
8	11 Mar	Absolute pose estimation from points or lines	**Assignment 3**: Relative and absolute pose estimation (20%)
9	18 Mar	Multiple-view geometry from points and/or lines	
10	25 Mar	Structure-from-Motion (SfM) and Visual Simultaneous Localization and Mapping (vSLAM)	**Due:** Assignment 3
11	01 Apr	Two-view and multi-view stereo	**Assignment 4:** Dense 3D model from multi-view stereo (20%)
12	08 Apr	Generalized cameras	
13	15 Apr	Factorization and non-rigid structure-from-motion	**Due:** Assignment 4

*Possible make-up lecture (to be confirmed): Auto-Calibration

Learning Outcomes

- Students should be able to:

1. Define the perspective-n-point (PnP) camera pose estimation problem.
2. Estimate the camera pose of an uncalibrated camera with n-point or line 2D-3D correspondences.
3. Use the Grunert (3-point), Quan (4-point) and EPnP (n-point) algorithms to estimate the pose of a calibrated camera.
4. Describe the degeneracies of the camera pose estimation problem.

Acknowledgements

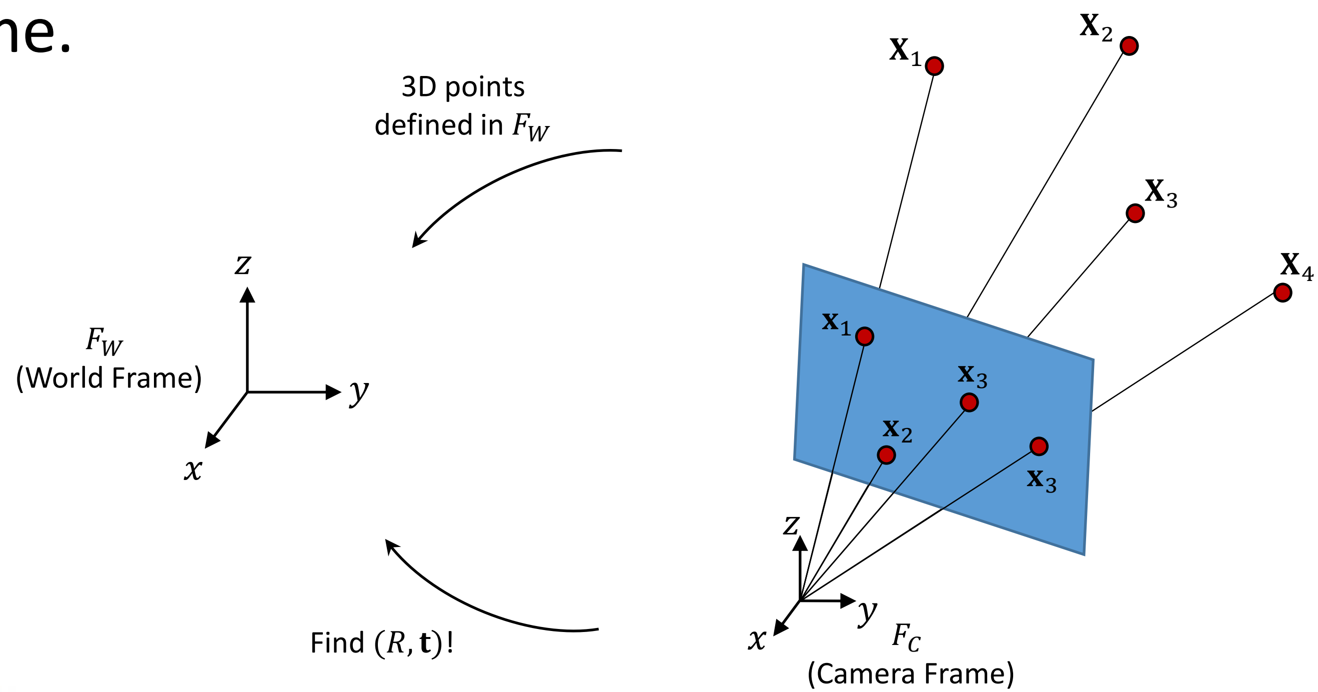
- A lot of slides and content of this lecture are adopted from:

1. R. Hartley, and A. Zisserman: "Multiple view geometry in computer vision", Chapter 7.
2. R. Haralick et. al, "Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem", IJCV 1994.
3. E. H. Thomspon, "Space resection: failure cases", Photogrammetric Record 1966.
4. L. Quan, Z. Lan, "Linear N-Point Camera Pose Determination", TPAMI 1999.
5. V. Lepetit, F. Moreno-Noguer, P. Fua, "EPnP: An Accurate O(n) Solution to the PnP Problem", IJCV 2009.

Perspective Pose Estimation Problem

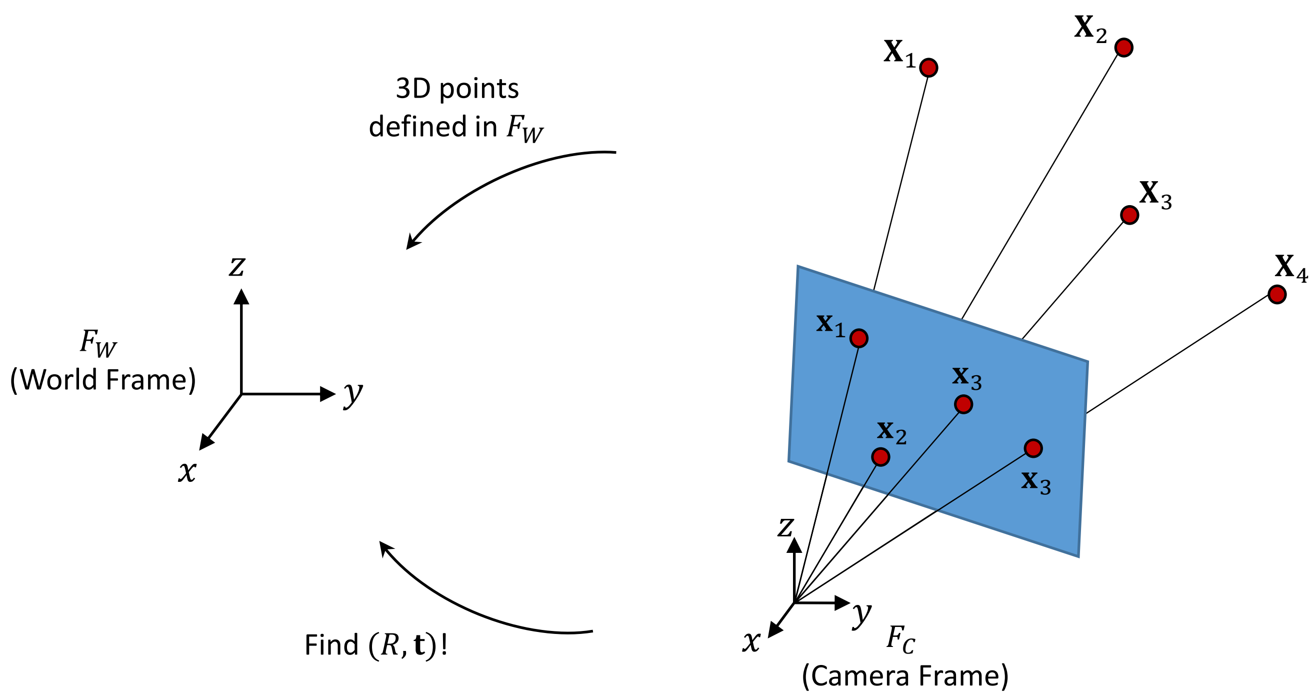
Given: a set of 3D points $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N\}$ defined in a world coordinate frame, and its corresponding 2D image points $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, i.e. $\{\mathbf{X}_i \leftrightarrow \mathbf{x}_i\}$.

Find: the camera pose $(R, \mathbf{t})$ in the world coordinate frame.



Perspective Pose Estimation Problem

- This problem is also known as the "Perspective-n-Point" or PnP problem.



Uncalibrated Camera: Unknown $\mathbf{K}$

- We are required to find a camera matrix P, i.e. a 3×4 matrix such that $\gamma_i \mathbf{x}_i = \mathrm{P}\mathbf{X}_i$, from the correspondences $\{\mathbf{X}_\mathrm{i} \leftrightarrow \mathbf{x}_i\}, \forall i$.

- The camera projection can be written as cross-product to eliminate the unknown scale γ_i:

$$(\gamma_i \mathbf{x}_i) \times \mathrm{P}\mathbf{X}_i = 0 \quad \Rightarrow \begin{bmatrix} \mathbf{0}^\top & -w_i\mathbf{X}_i^\top & y_i\mathbf{X}_i^\top \\ w_i\mathbf{X}_i^\top & \mathbf{0}^\top & -x_i\mathbf{X}_i^\top \\ -y_i\mathbf{X}_i^\top & x_i\mathbf{X}_i^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = \mathbf{0}.$$

- where each $\mathbf{P}^{iT}$ is a 4-vector, the i-th row of P.

Uncalibrated Camera: Unknown $\mathbf{K}$

- Only the first two equations are independent:

$$\left[\begin{array}{ccc} \mathbf{0}^\top & -w_i\mathbf{X}_i^\top & y_i\mathbf{X}_i^\top \\ w_i\mathbf{X}_i^\top & \mathbf{0}^\top & -x_i\mathbf{X}_i^\top \\ \cancel{-y_i\mathbf{X}_i^\top} & \cancel{x_i\mathbf{X}_i^\top} & \cancel{\mathbf{0}^\top} \end{array}\right]\begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = \mathbf{0} \Rightarrow \left[\begin{array}{ccc} \mathbf{0}^\top & -w_i\mathbf{X}_i^\top & y_i\mathbf{X}_i^\top \\ w_i\mathbf{X}_i^\top & \mathbf{0}^\top & -x_i\mathbf{X}_i^\top \end{array}\right]\begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = \mathbf{0}$$

- From a set of n point correspondences, we obtain a $2n \times 12$ matrix A by stacking up the equations for each correspondence.

- P is computed by solving the set of equations $\mathrm{A}\mathbf{p} = \mathbf{0}$, where $\mathbf{p}$ is the 12-vector containing the entries of the matrix P.

Uncalibrated Camera: Unknown K

Minimal solution:

- Since the matrix P has 11 dofs (12 entries – 1 dof for scale), a minimum of 5.5 correspondences are required.
- Effectively 6 point correspondences are needed, where only one of the equations is used from the sixth point.
- The solution is obtained by solving $\mathrm{A}\mathbf{p} = \mathbf{0}$, where A is an 11×12 matrix in this case.
- In general A will have rank 11, and the solution vector **p** is the 1-dimensional right null-space of A.

Uncalibrated Camera: Unknown K

Over-determined solution:

- No exact solution to $\mathrm{A}\mathbf{p} = \mathbf{0}$ when data is noisy and $n \geq 6$ point correspondences are needed.
- P may be obtained by minimizing an algebraic or geometric error.
- Minimize algebaric error $\|\mathrm{A}\mathbf{p}\|$ subject to a normalization constraint using SVD:

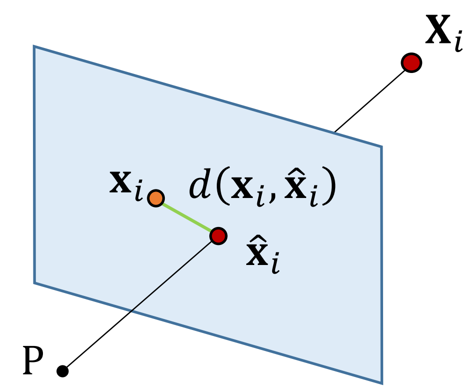
i. $\|\mathbf{p}\| = 1$;

ii. $\|\hat{\mathbf{p}}^3\| = 1$, where $\hat{\mathbf{p}}^3$ is the vector $(p_{31}, p_{32}, p_{33})^T$, i.e. the first three entries in the last row of P.

Uncalibrated Camera: Unknown $\mathbf{K}$

- Minimize the geometric error:

$$\min_{\mathrm{P}} \sum_i d(\mathbf{x}_i, \mathrm{P}\mathbf{X}_i)^2$$



- where $\mathbf{x}_i$ is the measured point and $\hat{\mathbf{x}}_i$ is the point $\mathrm{P}\mathbf{X}_i$, i.e. the point which is the exact image of $\mathbf{X}_i$ under P.

- $d(\mathbf{x}, \mathbf{y})$ is the Euclidean distance between two points $\mathbf{x}, \mathbf{y}$.

- Minimization with Levenberg–Marquardt.

Uncalibrated Camera: Unknown $\mathbf{K}$

Data normalization:

- The 2D points $\mathbf{x}_i$ should be translated for the centroid to be at the origin, and scaled for the RMS distance from the origin to be $\sqrt{2}$.

c: centroid of all 2D image points

$$\mathrm{T}_{\mathrm{norm}} = \begin{bmatrix} s & 0 & -sc_x \\ 0 & s & -sc_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$s = \frac{\sqrt{2}}{\bar{d}}$$

where $\bar{d}$: mean distance of all points from centroid.

Uncalibrated Camera: Unknown $\mathbf{K}$

Data normalization:

- Similarly, the 3D points $\mathbf{X}_i$ should be translated for the centroid to be at the origin, and scaled for the RMS distance from the origin to be $\sqrt{3}$.

$$\mathrm{U}_{\mathrm{norm}} = \begin{bmatrix} s & 0 & 0 & -sc_x \\ 0 & s & 0 & -sc_y \\ 0 & 0 & s & -sc_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

c: centroid of all 3D points

$$s = \frac{\sqrt{3}}{\bar{d}}$$

where $\bar{d}$: mean distance of all points from centroid.

Uncalibrated Camera: Unknown $\mathbf{K}$

<u>Objective</u>

Given $n \geq 6$ world to image point correspondences $\{\mathbf{X}_i \leftrightarrow \mathbf{x}_i\}$, determine the Maximum Likelihood estimate of the camera projection matrix $\mathtt{P}$, i.e. the $\mathtt{P}$ which minimizes $\sum_i d(\mathbf{x}_i, \mathtt{P}\mathbf{X}_i)^2$.

<u>Algorithm</u>

(i) **Linear solution.** Compute an initial estimate of $\mathtt{P}$ using a linear method such as algorithm 4.2(p109):

(a) **Normalization:** Use a similarity transformation $\mathtt{T}$ to normalize the image points, and a second similarity transformation $\mathtt{U}$ to normalize the space points. Suppose the normalized image points are $\tilde{\mathbf{x}}_i = \mathtt{T}\mathbf{x}_i$, and the normalized space points are $\tilde{\mathbf{X}}_i = \mathtt{U}\mathbf{X}_i$.

(b) **DLT:** Form the $2n \times 12$ matrix $\mathtt{A}$ by stacking the equations (7.2) generated by each correspondence $\tilde{\mathbf{X}}_i \leftrightarrow \tilde{\mathbf{x}}_i$. Write $\mathbf{p}$ for the vector containing the entries of the matrix $\tilde{\mathtt{P}}$. A solution of $\mathtt{A}\mathbf{p} = \mathbf{0}$, subject to $\|\mathbf{p}\| = 1$, is obtained from the unit singular vector of $\mathtt{A}$ corresponding to the smallest singular value.

(ii) **Minimize geometric error.** Using the linear estimate as a starting point minimize the geometric error (7.4):

$$\sum_i d(\tilde{\mathbf{x}}_i, \tilde{\mathtt{P}}\tilde{\mathbf{X}}_i)^2$$

over $\tilde{\mathtt{P}}$, using an iterative algorithm such as Levenberg–Marquardt.

(iii) **Denormalization.** The camera matrix for the original (unnormalized) coordinates is obtained from $\tilde{\mathtt{P}}$ as

$$\mathtt{P} = \mathtt{T}^{-1}\tilde{\mathtt{P}}\mathtt{U}.$$

Table Source: R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”

Uncalibrated Camera: Unknown $\mathbf{K}$

- We have seen in Lecture 4 that KR can be found from the RQ decomposition of M:

$$\mathrm{P} = [\mathrm{M} \mid -\mathrm{M}\widetilde{\mathbf{C}}] = \mathrm{K}[\mathrm{R} \mid -\mathrm{R}\widetilde{\mathbf{C}}] \ .$$

- The camera center C can be solved from the last column of P, i.e. $\widetilde{\mathbf{C}} = -\mathrm{M}^{-1}\mathbf{p}_4$.

- We can solve for the unknown depth λ using one 3D point X:

$$\mathbf{X} = \mathrm{P}^{+}\mathbf{x} + \lambda\mathbf{C}\,, \text{ where } \mathrm{P}^{+} = \mathrm{P}^{T}\,(\mathrm{P}\mathrm{P}^{T})^{-1}.$$

Uncalibrated Camera: Unknown $\mathbf{K}$

Line correspondences:

- A line in 3D may be represented by two points $\mathbf{X}_0$ and $\mathbf{X}_1$ through which the line passes.
- We know from Lecture 5 that the plane formed by back-projecting from the image line $\mathbf{l}$ is equal to $\mathrm{P}^T\mathbf{l}$.
- The condition that the point $\mathbf{X}_j$ lies on this plane is then

$$\mathbf{l}^{\mathsf{T}}\mathrm{P}\mathbf{X}_j = 0 \text{ for } j = 0, 1.$$

Uncalibrated Camera: Unknown K

Line correspondences:

- Each choice of j gives a single linear equation in the entries of the matrix P, so two equations are obtained for each 3D to 2D line correspondence.

- We can form $\mathrm{A}\mathbf{p} = \mathbf{0}$ from n 3D-2D line correspondence, where $\mathrm{A} \in \mathbb{R}^{2n\times 12}$ and $\mathbf{p} \in \mathbb{R}^{12\times 1}$.

- Similar to point correspondences, we minimize algebraic error $\|\mathrm{A}\mathbf{p}\|$ subject to a normalization constraint using SVD.

Uncalibrated Camera: Unknown $\mathbf{K}$

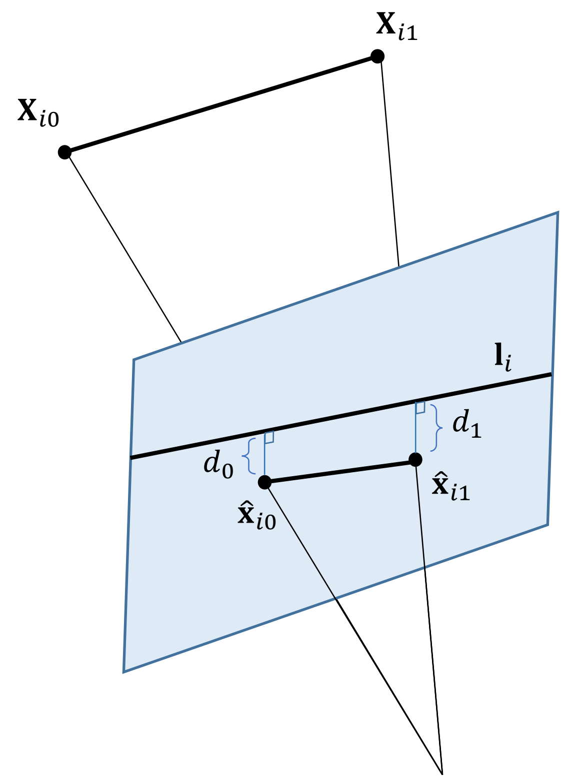
Line correspondences:

- Minimize the geometric error:

$$\min_{\mathrm{P}} \sum_i e(\mathbf{l}_i, \underbrace{\mathrm{P}\mathbf{X}_{i0}}_{\hat{\mathbf{x}}_{i0}}, \underbrace{\mathrm{P}\mathbf{X}_{i1}}_{\hat{\mathbf{x}}_{i1}})$$

- where

$$e = \frac{d_0 + d_1}{2(\|\hat{\mathbf{x}}_{i0} - \hat{\mathbf{x}}_{i1}\|)}, \quad d_j = \frac{\left|(\hat{\mathbf{x}}_{ij})^T \mathbf{l}_i\right|}{\sqrt{l_{ix}^2 + l_{iy}^2}}.$$



Source:
G. H. Lee, A Minimal Solution for Non-Perspective Pose Estimation from Line Correspondences, In ECCV, 2016

Calibrated Camera: Known K

- A two-stage approach is used to solve the PnP problem when the camera intrinsic K is known:

1. Compute the unknown depths s_1, s_2, s_3.
2. Solve for unknown $(R, \mathbf{t})$ using absolute orientation algorithm.

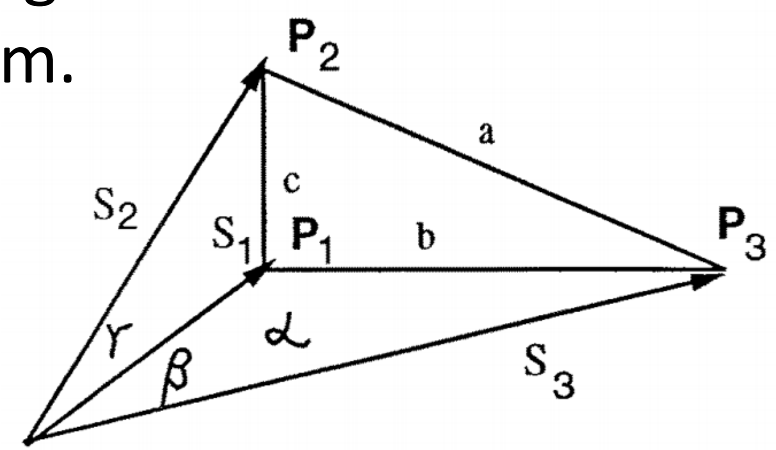


Image source: R. Haralick et. al, "Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem", IJCV 1994.

Calibrated Camera: Known $\mathbf{K}$

- The unknowns $(R, \mathbf{t})$ have 6 degree of freedom (3 for R and 3 for t).
- And each correspondence gives 2 constraints.
- This means that a minimal 3 point correspondences is needed.

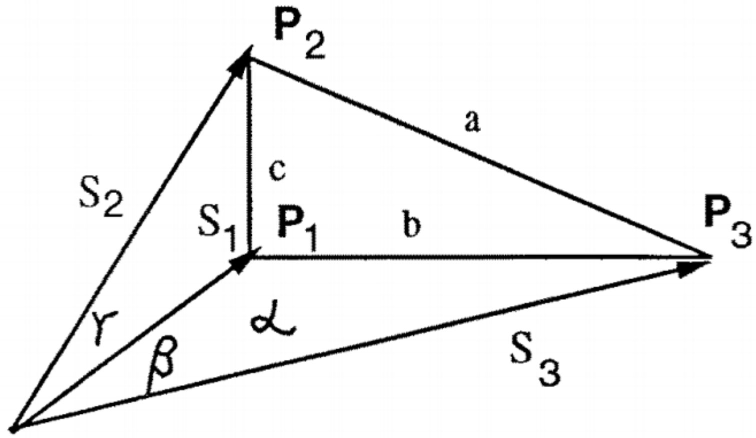


Image source: R. Haralick et. al, "Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem", IJCV 1994.

The Grunert (1841) Solution

- Let the unknown positions of the **three points** of the known triangle be:

$$p_1, p_2, \text{ and } p_3; \; p_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}, \quad i = 1, 2, 3.$$

- Let the known side lengths of the triangle be:

$$a = \|p_2 - p_3\|$$
$$b = \|p_1 - p_3\|$$
$$c = \|p_1 - p_2\|.$$

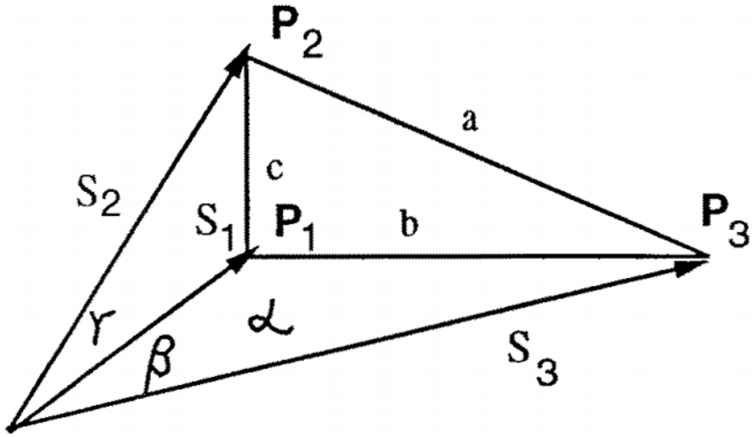


Image source: R. Haralick et. al, "Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem", IJCV 1994.

The Grunert (1841) Solution

- Let the observed perspective projection of $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ be $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$, respectively:

$$q_i = \begin{pmatrix} u_i \\ v_i \end{pmatrix}, \quad i = 1, 2, 3.$$

- By the perspective equations,

$$u_i = f\frac{x_i}{z_i} \quad , \qquad v_i = f\frac{y_i}{z_i}.$$

- The unit vectors pointing from the center of perspectivity to the observed points $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ are given by

$$j_i = \frac{1}{\sqrt{u_i^2 + v_i^2 + f^2}} \begin{pmatrix} u_i \\ v_i \\ f \end{pmatrix}, \quad i = 1, 2, 3 \ .$$

The Grunert (1841) Solution

- Let the angles at the center of perspectivity opposite sides a, b, c be α, β, and γ, then:

$$\cos\alpha = j_2 \cdot j_3\,, \quad \cos\beta = j_1 \cdot j_3\,, \quad \cos\gamma = j_1 \cdot j_2$$

- Let the unknown distances of the points $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ from the center of perspectivity be s_1, s_2, s_3, then

$$p_i = s_i j_i, \quad i = 1, 2, 3.$$

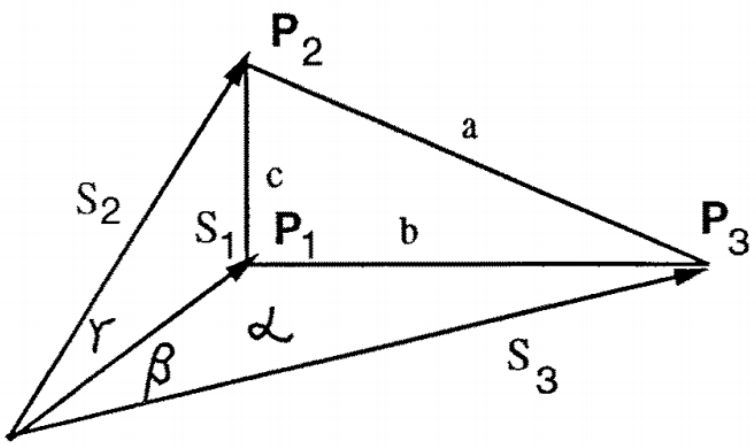


Image source: R. Haralick et. al, "Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem", IJCV 1994.

The Grunert (1841) Solution

- By the law of cosines, we have:

$$
\begin{aligned}
s_2^2 + s_3^2 - 2 s_2 s_3 \cos\alpha &= a^2 &\quad (1)\\
s_1^2 + s_3^2 - 2 s_1 s_3 \cos\beta &= b^2 &\quad (2)\\
s_1^2 + s_2^2 - 2 s_1 s_2 \cos\gamma &= c^2 &\quad (3)
\end{aligned}
$$

- Let

$$
s_2 = u s_1 \text{ and } s_3 = v s_1. \qquad (4)
$$

- Then

$$
s_1^2 = \frac{a^2}{u^2 + v^2 - 2uv\cos\alpha} = \frac{b^2}{1 + v^2 - 2v\cos\beta} = \frac{c^2}{1 + u^2 - 2u\cos\gamma} \qquad (5)
$$

The Grunert (1841) Solution

- Eliminating s_1^2 from Eq. (5), we get:

$$u^2 + \frac{b^2 - a^2}{b^2} v^2 - 2uv\cos\alpha + \frac{2a^2}{b^2} v\cos\beta - \frac{a^2}{b^2} = 0 \tag{6}$$

$$u^2 - \frac{c^2}{b^2} v^2 + 2v\frac{c^2}{b^2}\cos\beta - 2u\cos\gamma + \frac{b^2 - c^2}{b^2} = 0. \tag{7}$$

- Eliminating u from Eq (6) and (7), we get a 4-degree univariate polynomial:

$$A_4 v^4 + A_3 v^3 + A_2 v^2 + A_1 v + A_0 = 0$$

The Grunert (1841) Solution

- where the coefficients are given by:

$$A_4 = \left(\frac{a^2 - c^2}{b^2} - 1 \right)^2 - \frac{4c^2}{b^2} \cos^2 \alpha$$

$$A_3 = 4 \left[\frac{a^2 - c^2}{b^2} \left(1 - \frac{a^2 - c^2}{b^2} \right) \cos \beta - \left(1 - \frac{a^2 + c^2}{b^2} \right) \cos \alpha \cos \gamma \ + 2 \frac{c^2}{b^2} \cos^2 \alpha \cos \beta \right]$$

$$A_2 = 2 \left[\left(\frac{a^2 - c^2}{b^2} \right)^2 - 1 + 2 \left(\frac{a^2 - c^2}{b^2} \right)^2 \cos^2 \beta + 2 \left(\frac{b^2 - c^2}{b^2} \right) \cos^2 \alpha \right.$$
$$\left. - 4 \left(\frac{a^2 + c^2}{b^2} \right) \cos \alpha \cos \beta \cos \gamma \ + 2 \left(\frac{b^2 - a^2}{b^2} \right) \cos^2 \gamma \right]$$

$$A_1 = 4 \left[- \left(\frac{a^2 - c^2}{b^2} \right) \left(1 + \frac{a^2 - c^2}{b^2} \right) \cos \beta + \frac{2a^2}{b^2} \cos^2 \gamma \cos \beta \right.$$
$$\left. - \left(1 - \left(\frac{a^2 + c^2}{b^2} \right) \right) \cos \alpha \cos \gamma \right]$$

$$A_0 = \left(1 + \frac{a^2 - c^2}{b^2} \right)^2 - \frac{4a^2}{b^2} \cos^2 \gamma.$$

Solving the 4-Degree Polynomial

- Eigen-values of companion matrix are the roots of 4-degree polynomial, i.e. v:

$$\begin{bmatrix} 0 & 0 & 0 & -A_4/A_0 \\ 1 & 0 & 0 & -A_3/A_0 \\ 0 & 1 & 0 & -A_2/A_0 \\ 0 & 0 & 1 & -A_1/A_0 \end{bmatrix}.$$

- Solve for u by back-substitution of v into Eq (7).

- Solve for s_1, s_2, s_3 using u and v in Eq (5) and (4).

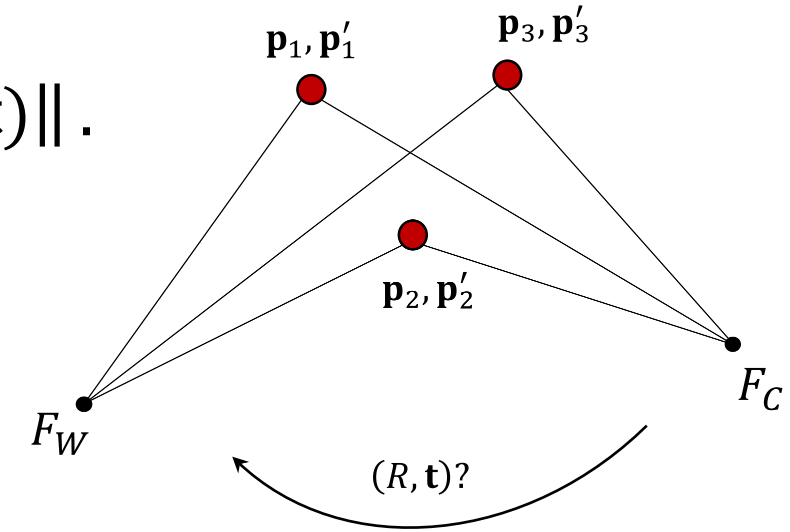
Absolute Orientation

- Once the unknown depths s_1, s_2, s_3 are computed, we have the 3D points in the camera frame, $\mathbf{p}'_1$, $\mathbf{p}'_2$, $\mathbf{p}'_3$.

- The goal is to recover the rigid transformation $(R, \mathbf{t})$ that aligns $\mathbf{p}'_1$, $\mathbf{p}'_2$, $\mathbf{p}'_3$ to $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, i.e.

$$\mathrm{argmin}_{R,\mathbf{t}} \sum_{i=1}^{i=n} \|\mathbf{p}_i - (R\mathbf{p}'_i + \mathbf{t})\| .$$

- Note that the algorithm works for $n \geq 3$ points.



Absolute Orientation

- **Step 1:** Remove translation by moving the respective centroid of $\mathbf{p}_i'$ and $\mathbf{p}_i$ to the origin of coordinate system.

$$\mathbf{r}_i = \mathbf{p}_i - \overline{\mathbf{p}}, \qquad \mathbf{r}_i' = \mathbf{p}_i' - \overline{\mathbf{p}}',$$

where

$$\overline{\mathbf{p}} = \frac{1}{n}\sum_{i=1}^{i=n} \mathbf{p}_i, \qquad \overline{\mathbf{p}}' = \frac{1}{n}\sum_{i=1}^{i=n} \mathbf{p}_i'.$$

Absolute Orientation

- **Step 2:** Compute rotation matrix as follows.

  Form matrix M from sum of outer product:

$$\mathbf{M} = \sum_{i=1}^{n} \mathbf{r}'_i \, \mathbf{r}_i^\top \, .$$

  The rotation matrix R is given by:

$$\mathbf{R} = \mathbf{M}\mathbf{Q}^{-1/2} \text{ , where } \quad \mathbf{Q} = \mathbf{M}^\top \mathbf{M}.$$

- **Step 3:** Given R, we can now compute $\mathbf{t}$ as

$$\mathbf{t} = \overline{\mathbf{p}}' - R\overline{\mathbf{p}}.$$

Inverse Square Root of a Matrix

- The inverse square root of Q can be easily computed as:

$$\mathbf{Q}^{-1/2} = \mathbf{V}\,\mathrm{diag}\left(\frac{1}{\sqrt{\lambda_1}}, \frac{1}{\sqrt{\lambda_2}}, \frac{1}{\sqrt{\lambda_3}}\right)\mathbf{V}^\top,$$

- $\mathbf{v}_i$ and λ_i are the eigenvectors and eigenvalues of Q:

$$\mathbf{Q} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top, \text{ where}$$

$$\mathbf{V} = [\mathbf{v}_1\ \mathbf{v}_2\ \mathbf{v}_3], \quad \mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3).$$

Degenerate Solution

- We are solving a system of polynomials with 3 unknowns s_1, s_2, s_3:

$$s_2^2 + s_3^2 - 2s_2s_3\cos\alpha = a^2 \qquad (1)$$
$$s_1^2 + s_3^2 - 2s_1s_3\cos\beta = b^2 \qquad (2)$$
$$s_1^2 + s_2^2 - 2s_1s_2\cos\gamma = c^2 \qquad (3)$$

which can be rewritten into:

$$f_1(x, y, z) = 0, \quad f_2(x, y, z) = 0, \quad f_3(x, y, z) = 0,$$

where $(x, y, z)^T$ is the camera center which is the function of s_1, s_2, s_3 (we will skip the full derivation).

Degenerate Solution

- With a loss in generality, suppose we wish to find n unknowns $x_1, x_2, \dots, x_n$, by the solution of n equations of the form:

$$f_i(x_1, x_2, \dots, x_n) = 0 \quad (i = 1, 2, \dots, n).$$

- For a non-degenerate solution, we must not be able to make first-order changes in any of the unknowns such that the equations remain satisfied.

Source: E. H. Thompson, “Space Resection: Failure Cases”, 1966.

Degenerate Solution

- That is:

$$\begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix} \begin{pmatrix} dx_1 \\ dx_2 \\ \vdots \\ dx_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

must have no solutions for $dx_1, dx_2, \ldots, dx_n$, other than zeros.

Source: E. H. Thompson, "Space Resection: Failure Cases", 1966.

Degenerate Solution

- In the context of the PnP problem, we take the total derivatives of:

$$f_1(x,y,z) = 0, \quad f_2(x,y,z) = 0, \quad f_3(x,y,z) = 0,$$

to get:

$$\underbrace{\frac{1}{s_1 s_2 s_3} AB}_{M_{3\times 3}} \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = \underbrace{\begin{pmatrix} df_1 \\ df_2 \\ df_3 \end{pmatrix}}_{(0,0,0)^T},$$

Homogeneous linear equation $\mathrm{Mx} = 0$!

where

$$A = \begin{pmatrix} x - x_1 & y - y_1 & z - z_1 \\ x - x_2 & y - y_2 & z - z_2 \\ x - x_3 & y - y_3 & z - z_3 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & s_2 - s_3 \cos\alpha & s_3 - s_2 \cos\alpha \\ s_1 - s_3 \cos\beta & 0 & s_3 - s_1 \cos\beta \\ s_1 - s_2 \cos\gamma & s_2 - s_1 \cos\gamma & 0 \end{pmatrix},$$

Degenerate Solution

- $\mathbf{p}_i$ are the 3D points and $\mathbf{p}$ is the camera center:

$$p_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}, \; i = 1, 2, 3, \qquad \mathbf{p} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}.$$

- The matrix $M_{3\times 3}$ must be rank deficient for degeneracy to happen!

- Two cases of degeneracy can be observed:

1. The three points are colinear.
2. The camera center and the three points lie are coplanar.

Degenerate Solution

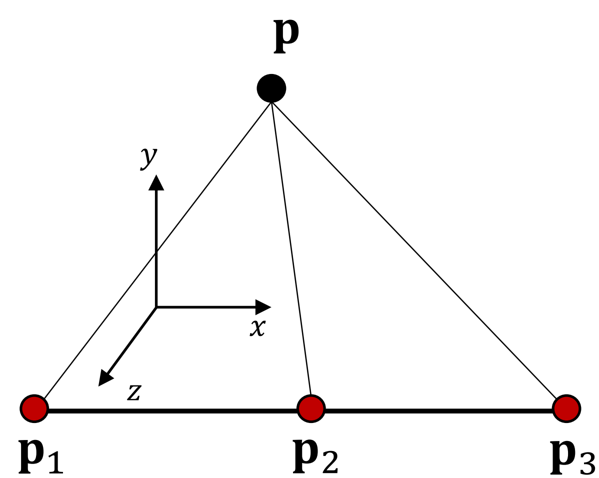
Case 1: The three points are colinear.

- The three points and camera center form a plane, hence one axis can be defined to vanish, e.g. z-axis vanishes when the xy-plane is on the plane.

- This causes A (hence $M_{3\times3}$) to be rank deficient, i.e.

$$A = \begin{pmatrix} x - x_1 & y - y_1 & z - z_1 \\ x - x_2 & y - y_2 & z - z_2 \\ x - x_3 & y - y_3 & z - z_3 \end{pmatrix}$$

This column becomes 0!



Degenerate Solution

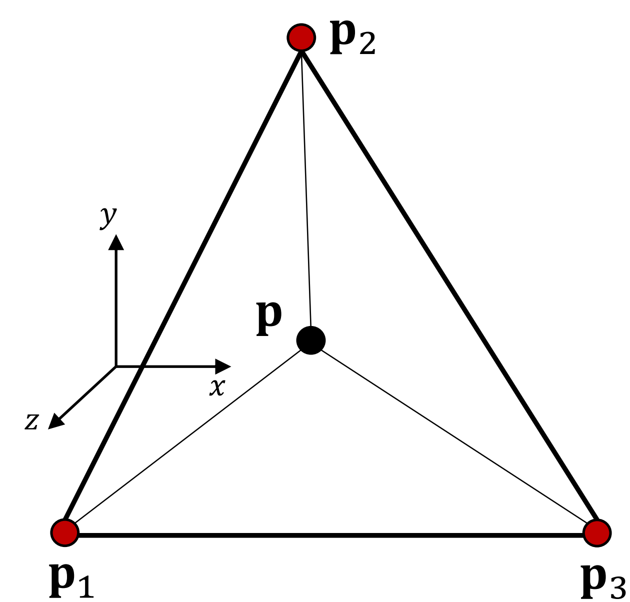
Case 2: The camera center and the three points lie are coplanar.

- Again, the three points and camera center form a plane, hence one axis can be defined to vanish, e.g. z-axis vanishes when the xy-plane is on the plane.

- This causes A (hence $M_{3\times3}$) to be rank deficient, i.e.

$$A = \begin{pmatrix} x - x_1 & y - y_1 & z - z_1 \\ x - x_2 & y - y_2 & z - z_2 \\ x - x_3 & y - y_3 & z - z_3 \end{pmatrix}$$

This column becomes 0!



NUS National University of Singapore | School of Computing

Linear N-Point Camera Pose

- The 3-point PnP algorithm gives a total of 4 possible solutions, and at least a 4th point is needed to get a unique solution.

- We will now look at how to do the two-stage camera pose estimation with $n \geq 4$ such that the solution is unique.

The Linear 4-Point Algorithm

- Recall that we have the system of 3 polynomials from the cosine rule:

$$s_2^2 + s_3^2 - 2s_2 s_3 \cos\alpha = a^2 \quad (1)$$
$$s_1^2 + s_3^2 - 2s_1 s_3 \cos\beta = b^2 \quad (2)$$
$$s_1^2 + s_2^2 - 2s_1 s_2 \cos\gamma = c^2 \quad (3)$$

which can be rewritten into:

$$f_{12}(s_1, s_2) = 0,$$
$$f_{13}(s_1, s_3) = 0,$$
$$f_{23}(s_2, s_3) = 0.$$

The Linear 4-Point Algorithm

- In general, we can eliminate s_2 and s_3 from the system of polynomials to get a 4-degree univariate polynomial in term of $x = s_1^2$:

$$g(x) = a_5 x^4 + a_4 x^3 + a_3 x^2 + a_2 x + a_1 = 0.$$

- which has at most four solutions for x; and to obtain a unique solution, we need to add one more point.

- For $n = 4$, an overconstrained system of six polynomials $f_{ij}(s_i, s_j) = 0$ is obtained for the four unknowns s_1; s_2; s_3; s_4.

The Linear 4-Point Algorithm

A straightforward solution?

- Take subsets of three of the four points, solve the fourth degree polynomial equation for each subset of three points.
- And, finally, find the common solution of the subsets.
- Not a good approach for 3 reasons!

The Linear 4-Point Algorithm

- Drawbacks to the straightforward solution:

1. Have to solve several fourth degree polynomials.
2. Need to find the common solution, which might be difficult due to noisy data.
3. Probably the most important part, is that we cannot profit from the data redundancy, which should increase stability.

The Linear 4-Point Algorithm

- A better solution proposed by Quan and Lan [TPAMI'99] is as follow.
- For $n = 4$, the three fourth degree polynomials are:

$$\begin{cases} g(x) & = & a_5x^4 & + & a_4x^3 & + & a_3x^2 & + & a_2x & + & a_1 & = & 0, \\ g'(x) & = & a'_5x^4 & + & a'_4x^3 & + & a'_3x^2 & + & a'_2x & + & a'_1 & = & 0, \\ g''(x) & = & a''_5x^4 & + & a''_4x^3 & + & a''_3x^2 & + & a''_2x & + & a''_1 & = & 0. \end{cases}$$

$$f_{12}(s_1, s_2), f_{13}(s_1, s_3), f_{23}(s_2, s_3) \implies g(x)$$

$$f_{12}(s_1, s_2), f_{14}(s_1, s_4), f_{24}(s_2, s_4) \implies g'(x)$$

$$f_{13}(s_1, s_3), f_{14}(s_1, s_4), f_{34}(s_3, s_4) \implies g''(x)$$, where $x = s_1^2$.

The Linear 4-Point Algorithm

- The three 4-dgree polynomial can be rewritten in matrix form:

$$\begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ a_1' & a_2' & a_3' & a_4' & a_5' \\ a_1'' & a_2'' & a_3'' & a_4'' & a_5'' \end{pmatrix} \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \end{pmatrix} = \mathbf{A}_{3\times 5}\mathbf{t}_5 = 0,$$

where

$$\mathbf{t}_5 = (t_0, t_1, \ldots, t_4)^T = (1, x, \ldots, x^4)^T.$$

The Linear 4-Point Algorithm

- Since the matrix $A_{3\times5}$ has at most rank 3 $= \min(3{,}5)$, let its singular value decomposition be:

$$\mathbf{U}_{3\times5}\mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, 0, 0)(\mathbf{v}_1, \ldots, \mathbf{v}_5)^T.$$

- The null space of $A_{3\times5}$ is spanned by the right singular vectors $\mathbf{v}_4$ and $\mathbf{v}_5$.

- A one-dimensional solution space for $\mathbf{t}_5$, parameterized by λ and ρ can be constructed as:

$$\mathbf{t}_5 = \lambda\mathbf{v}_4 + \rho\mathbf{v}_5 \quad \text{for} \quad \lambda, \rho \in \mathbb{R}.$$

The Linear 4-Point Algorithm

- Now consider the nonlinear constraints among the components of $\mathbf{t}_5$; it can be easily checked that:

$$t_i t_j = t_k t_l \text{ for } i + j = k + l, \quad 0 \leq i, j, k, l \leq 4 \; .$$

- Substituting the components of $\mathbf{t}_5$ into the constraint, we get:

$$b_1 \lambda^2 + b_2 \lambda \rho + b_3 \rho^2 = 0,$$

where

$$\begin{aligned}
b_1 &= \mathbf{v}_4^{(i)} \mathbf{v}_4^{(j)} - \mathbf{v}_4^{(k)} \mathbf{v}_4^{(l)}, \\
b_2 &= \mathbf{v}_4^{(i)} \mathbf{v}_5^{(j)} + \mathbf{v}_5^{(i)} \mathbf{v}_4^{(j)} - (\mathbf{v}_4^{(k)} \mathbf{v}_5^{(l)} + \mathbf{v}_5^{(k)} \mathbf{v}_4^{(l)}), \\
b_3 &= \mathbf{v}_5^{(i)} \mathbf{v}_5^{(j)} - \mathbf{v}_5^{(k)} \mathbf{v}_5^{(l)} \; .
\end{aligned}$$

The Linear 4-Point Algorithm

- As shown in the table, we have seven such equations for the seven different values of

$$\{(i, j, k, l), i + j = k + l \text{ and } 0 \leq i, j, k, l \leq 4\}$$

modulo the interchanges of i and j or k and l.

- These seven quadratic equations can be written in the following matrix form:

$$\begin{pmatrix} b_1 & b_2 & b_3 \\ b_1' & b_2' & b_3' \\ \vdots & \vdots & \vdots \\ b_1^{(6)} & b_2^{(6)} & b_3^{(6)} \end{pmatrix} \begin{pmatrix} \lambda^2 \\ \lambda\rho \\ \rho^2 \end{pmatrix} = \mathbf{B}_{7\times 3}\mathbf{y}_3 = 0.$$

(i,j,k,l)
(4, 2, 3, 3)
(4, 1, 3, 2)
(4, 0, 3, 1)
(4, 0, 2, 2)
(3, 1, 2, 2)
(3, 0, 2, 2)
(2, 0, 1, 1)

The Linear 4-Point Algorithm

- Again, this overdetermined system can be viewed as linear in λ^2 , $\lambda\rho$, and ρ^2.

- And solved by SVD as the right singular vector of the smallest singular value of $\mathrm{B}_{7\times 3}$.

- Given the null vector $\mathbf{y}_3$, we solve for:

$$\lambda/\rho = y_0/y_1 \quad \text{or} \quad \lambda/\rho = y_1/y_2.$$

The Linear 4-Point Algorithm

- After obtaining the ratio λ/ρ, the scalars λ and ρ can be determined using the first scalar equation from $\mathbf{t}_5$, i.e.

$$\mathbf{t}_5 = \begin{pmatrix} \boxed{1} \\ x \\ x^2 \\ x^3 \\ x^4 \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{v}_4^{(0)} \\ \mathbf{v}_4^{(1)} \\ \mathbf{v}_4^{(2)} \\ \mathbf{v}_4^{(3)} \\ \mathbf{v}_4^{(4)} \end{pmatrix} + \rho \begin{pmatrix} \mathbf{v}_5^{(0)} \\ \mathbf{v}_5^{(1)} \\ \mathbf{v}_5^{(2)} \\ \mathbf{v}_5^{(3)} \\ \mathbf{v}_5^{(4)} \end{pmatrix} \quad \Longrightarrow \quad 1 = \lambda \mathbf{v}_4^{(0)} + \rho \mathbf{v}_5^{(0)}.$$

The Linear 4-Point Algorithm

- The vector $\mathbf{t}_5$, is therefore, completely determined; and the final x is taken to be:

$$x = t_1/t_0 \text{ or } \ t_2/t_1 \ \text{ or } \ t_3/t_2 \ \text{ or } \ t_4/t_3,$$

  or the average of all these values.

- Since $x = s_1^2$, the final depth is $s_1 = \sqrt{x}$; note that $-\sqrt{x}$ is omitted since $s_1 > 0$.

The Linear 4-Point Algorithm

- s_2, s_3, s_4 are solved by back-substitution.
- Finally, apply absolute orientation to recover the camera pose $(R, \mathbf{t})$.
- Hence, the camera pose is uniquely determined by four point correspondences provided that the 4 points are not degenerate.

The Linear n-Point Algorithm

- From $n = 5$ on, there are sufficiently many fourth degree polynomials to directly solve $t_i = x^i$ linearly.

- For the $n = 5$ case, six fourth degree polynomials can be arranged into the following matrix equation:

$$\begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ a_1' & a_2' & a_3' & a_4' & a_5' \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_1^{(5)} & a_2^{(5)} & a_3^{(5)} & a_4^{(5)} & a_5^{(5)} \end{pmatrix} \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \end{pmatrix} = \mathbf{A}_{6\times 5}\mathbf{t}_5 = 0.$$

The Linear n-Point Algorithm

- Let the singular value decomposition of $A_{6\times5}$ be $U_{6\times6}\Sigma_{6\times5}V_{5\times5}^T$.

- The vector $\mathbf{t}_5$ is directly obtained as the right singular vector $\mathbf{v}_5$ of the smallest singular value of $A_{6\times5}$.

- Then x can be obtained as in the linear 4-point algorithm.

The Linear n-Point Algorithm

- The same algorithm is also valid for any $n \geq 5$ points; we just need to SVD the matrix A of

$$\frac{(n-1)(n-2)}{2} \times 5$$

  to get the solution for the vector $\mathbf{t}_5$.

- Overall complexity of the algorithm is $\mathcal{O}(n^3)$ for the SVD, where n is the number of points.

EPnP: An Accurate $\mathcal{O}(n)$ Solution

- The linear n-point algorithm becomes intractable with large number of points.

- An $\mathcal{O}(n)$ solution is introduced by [Lepetit'06]:

  - Express the points as a weighted sum of virtual control points.

  - The coordinates of the control points in the camera frame become the unknown of our problem.

  - For large n's, this is a much smaller number of unknowns than n depth values.

EPnP: An Accurate $\mathcal{O}(n)$ Solution

4 non-coplanar control points:

- Let the 3D points in the world frame be:

$$\mathbf{p}_i^w, \quad i = 1, \ldots, n.$$

- Similarly, let the 4 control points we use to express their world coordinates:

$$\mathbf{c}_j, \quad j = 1, \ldots, 4.$$

EPnP: An Accurate $\mathcal{O}(n)$ Solution

4 non-coplanar control points:

- We express each reference point as a weighted sum of the control points:

$$\mathbf{p}_i^w = \sum_{j=1}^{4} \alpha_{ij} \mathbf{c}_j^w, \quad \text{with} \sum_{j=1}^{4} \alpha_{ij} = 1,$$

- The same relation holds for the 3D points in the camera coordinate system, i.e.

$$\mathbf{p}_i^c = \sum_{j=1}^{4} \alpha_{ij} \mathbf{c}_j^c.$$

EPnP: An Accurate $\mathcal{O}(n)$ Solution

- We can compute the control points in the world frame as follow:

1. Select centroid of world points as the first control point, i.e.

$$\mathbf{c}_0^w = \frac{1}{n}\sum_{i=1}^{i=n}\mathbf{p}_i^w.$$

2. Select the other three control points as the principal axes $\mathbf{c}_1^w = \mathbf{u}_1$, $\mathbf{c}_2^w = \mathbf{u}_2$, $\mathbf{c}_3^w = \mathbf{u}_3$ of the world points, i.e.

Covariance Matrix: $V = \frac{1}{n}\sum_{i=1}^{i=n}(\mathbf{p}_i^w - \mathbf{c}_0^w)(\mathbf{p}_i^w - \mathbf{c}_0^w)^T$,

$\mathrm{SVD}(V) = U\Sigma U^T$, where $U = [\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3]^T$.

EPnP: An Accurate $\mathcal{O}(n)$ Solution

- For each 3D point $\mathbf{p}_i^w$, the weights $\alpha_{ij}, j = 1,2,3,4$ for each point can be computed from:

$$\mathbf{p}_i^w = \sum_{j=1}^{4} \alpha_{ij} \mathbf{c}_j^w, \quad \text{with} \sum_{j=1}^{4} \alpha_{ij} = 1,$$

$$\Downarrow$$

$$\begin{pmatrix} p_{ix}^w \\ p_{iy}^w \\ p_{iz}^w \\ 1 \end{pmatrix} = \alpha_{i1} \begin{pmatrix} x_1^w \\ y_1^w \\ z_1^w \\ 1 \end{pmatrix} + \alpha_{i2} \begin{pmatrix} x_2^w \\ y_2^w \\ z_2^w \\ 1 \end{pmatrix} + \alpha_{i3} \begin{pmatrix} x_3^w \\ y_3^w \\ z_3^w \\ 1 \end{pmatrix} + \alpha_{i4} \begin{pmatrix} x_4^w \\ y_4^w \\ z_4^w \\ 1 \end{pmatrix}$$

Four equations and four unknowns!

EPnP: An Accurate $\mathcal{O}(n)$ Solution

- Now we can solve for the control points in the camera frame.
- Let A be the camera internal calibration matrix and $\{\mathbf{u}_i\}_{i=1,\dots,n}$ the 2D projections of the $\{\mathbf{p}_i^c\}_{i=1,\dots,n}$ points, we get:

$$\forall i, \quad w_i \begin{bmatrix} \mathbf{u}_i \\ 1 \end{bmatrix} = \mathbf{A}\mathbf{p}_i^c = \mathbf{A} \sum_{j=1}^{4} \alpha_{ij} \mathbf{c}_j^c,$$

where the w_i are scalar projective parameters.

EPnP: An Accurate $\mathcal{O}(n)$ Solution

- More specifically, we have:

$$\forall i, \quad w_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_c \\ 0 & f_v & v_c \\ 0 & 0 & 1 \end{bmatrix} \sum_{j=1}^{4} \alpha_{ij} \begin{bmatrix} x_j^c \\ y_j^c \\ z_j^c \end{bmatrix},$$

where

- $\mathbf{c}_j^c = [x_j^c, y_j^c, z_j^c]^\top, \quad \mathbf{u}_i = [u_i, v_i]^\top$
- f_u, f_v are the focal length coefficients
- (u_c, v_c) is the principal point

EPnP: An Accurate $\mathcal{O}(n)$ Solution

- The last row of the equation implies:

$$w_i = \sum_{j=1}^{4} \alpha_{ij} z_j^c \ ,$$

- which is substituted back into the first two rows to eliminate w_i, hence we get:

$$\sum_{j=1}^{4} \alpha_{ij} f_u x_j^c + \alpha_{ij}(u_c - u_i) z_j^c = 0,$$

$$\sum_{j=1}^{4} \alpha_{ij} f_v y_j^c + \alpha_{ij}(v_c - v_i) z_j^c = 0.$$

EPnP: An Accurate $\mathcal{O}(n)$ Solution

- By concatenating them for all n points, we generate a linear system of the form

$$\mathbf{M}\mathbf{x} = \mathbf{0},$$

where

➢ $\mathbf{x} = [\mathbf{c}_1^{c\top}, \mathbf{c}_2^{c\top}, \mathbf{c}_3^{c\top}, \mathbf{c}_4^{c\top}]^\top$ is a 12-vector of the unknowns

➢ M is a 2n × 12 matrix from the coefficients

EPnP: An Accurate $\mathcal{O}(n)$ Solution

- The solution therefore belongs to the null space, or kernel, of M, and can be expressed as:

$$\mathbf{x} = \sum_{i=1}^{N} \beta_i \mathbf{v}_i$$

- where the set $\mathbf{v}_i$ are the columns of the right-singular vectors of M corresponding to the N null singular values of M.

- They can be computed efficiently as the null eigenvectors of matrix $\mathrm{M}^T\mathrm{M} \in \mathbb{R}^{12\times 12}$ (constant size).

- Most time consuming step is to compute $\mathrm{M}^T\mathrm{M}$ with complexity of $\mathcal{O}(n)$.

EPnP: An Accurate $\mathcal{O}(n)$ Solution

Case $N = 1$:

- This is the case when there are $n \geq 6$ points, i.e. 1 null-space vector $\mathbf{x} = \beta\mathbf{v}$.
- Let $\mathbf{v}^{[i]}$ be the sub-vector of $\mathbf{v}$ that corresponds to the coordinates of the control point $\boldsymbol{c}_i^c$.
- Maintaining the distance between pairs of control points $(\boldsymbol{c}_i, \boldsymbol{c}_j)$ implies that:

$$\|\beta\mathbf{v}^{[i]} - \beta\mathbf{v}^{[j]}\|^2 = \|\mathbf{c}_i^w - \mathbf{c}_j^w\|^2 \quad .$$

EPnP: An Accurate $\mathcal{O}(n)$ Solution

Case $N = 1$:

- Since the $\|\boldsymbol{c}_i^w - \boldsymbol{c}_j^w\|$ distances are known, we compute β in closed-form as

$$\beta = \frac{\sum_{\{i,j\}\in[1;4]} \|\mathbf{v}^{[i]} - \mathbf{v}^{[j]}\| \cdot \|\mathbf{c}_i^w - \mathbf{c}_j^w\|}{\sum_{\{i,j\}\in[1;4]} \|\mathbf{v}^{[i]} - \mathbf{v}^{[j]}\|^2} \,.$$

EPnP: An Accurate $\mathcal{O}(n)$ Solution

Case $N = 2$:

- This is the case when there are $n = 5$ points, i.e. 2 null-space vector $\mathbf{x} = \beta_1 \mathbf{v}_1 + \beta_2 \mathbf{v}_2$.

- This distance constraints become:

$$\|(\beta_1 \mathbf{v}_1^{[i]} + \beta_2 \mathbf{v}_2^{[i]}) - (\beta_1 \mathbf{v}_1^{[j]} + \beta_2 \mathbf{v}_2^{[j]})\|^2 = \|\mathbf{c}_i^w - \mathbf{c}_j^w\|^2 .$$

- 4 control points $\Rightarrow$ 4 choose 2 = 6 constraints.

EPnP: An Accurate $\mathcal{O}(n)$ Solution

Case $N = 2$:

- Which can be rewritten as: $\mathbf{L}\boldsymbol{\beta} = \boldsymbol{\rho}$, where
  - $\beta = [\beta_{11}, \beta_{12}, \beta_{22}]^\top$ and $\beta_{11} = \beta_1^2, \beta_{12} = \beta_1\beta_2, \beta_{22} = \beta_2^2$.
  - L is a 6×3 matrix formed with $\mathbf{v}_1$ and $\mathbf{v}_2$
  - ρ is a 6-vector with the squared distances $\left\|\boldsymbol{c}_i^w - \boldsymbol{c}_j^w\right\|^2$
  - $\boldsymbol{\beta} = [\beta_{11}, \beta_{12}, \beta_{22}]^\top$ is the vector of unknowns.

- Overdeterminated linear equations $\Rightarrow \beta$ can be solved with the pseudoinverse of L!

EPnP: An Accurate $\mathcal{O}(n)$ Solution

Case $N = 3$:

- This is the case when there are $n = 4$ points, i.e. 3 null-space vector $\mathbf{x} = \beta_1 \mathbf{v}_1 + \beta_2 \mathbf{v}_2 + \beta_3 \mathbf{v}_3$.
- Similar to $N = 2$ case, we get 6 distance constraints that the linear equation:

$$\mathbf{L}\boldsymbol{\beta} = \boldsymbol{\rho} \ , \quad \text{where}$$

  - L is a square 6×6 matrix formed with $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$
  - $\beta = [\beta_{11}, \beta_{12}, \beta_{13}, \beta_{22}, \beta_{23}, \beta_{33}]^\top$.

- β can be solved with the pseudoinverse of L

EPnP: An Accurate $\mathcal{O}(n)$ Solution

3 coplanar control points:

- In the case where all points lie on a plane, we need only three control points.

- The dimensionality of M is then reduced to $2n \times 9$ with 9D eigenvectors $\mathbf{v}_i$, but the above equations remain mostly valid.

- Main difference is that the number of quadratic constraints drops from 6 to 3, which can be used to solve the cases of $N = 1$ and 2.

EPnP: An Accurate $\mathcal{O}(n)$ Solution

- Finally, we recover the 3D points in the camera frame with the control points as:

$$\mathbf{p}_i^c = \sum_{j=1}^{4} \alpha_{ij} \mathbf{c}_j^c .$$

- The camera pose $(R, \mathbf{t})$ can then be computed using the absolute orientation using the known 3D points $\mathbf{p}_i^c$ and $\mathbf{p}_i^w$.

Summary

- We have looked at how to:

1. Define the perspective-n-point (PnP) camera pose estimation problem.

2. Estimate the camera pose of an uncalibrated camera with n-point or line 2D-3D correspondences.

3. Use the Grunert (3-point), Quan (4-point) and EPnP (n-point) algorithms to estimate the pose of a calibrated camera.

4. Describe the degeneracies of the camera pose estimation problem.