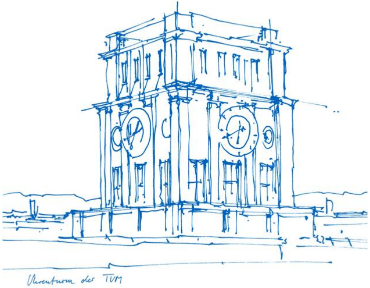
# Computer Vision II: Multiple View Geometry (IN2228)

## Chapter 08 3D-3D Geometry

Dr. Haoang Li

22 June 2023 11:00-11:45



# Explanation for Linear Systems of PnP

- Recap on System Generation

✓ DLT (direct, one-step)

$$\mathbf{t}_1^T\mathbf{P} - \mathbf{t}_3^T\mathbf{P}u_1 = 0,$$

$$\mathbf{t}_2^T\mathbf{P} - \mathbf{t}_3^T\mathbf{P}v_1 = 0.$$

Constraint of one correspondence

$$\Rightarrow \begin{pmatrix} \boxed{\begin{matrix} \mathbf{P}_1^T & 0 & -u_1\mathbf{P}_1^T \\ 0 & \mathbf{P}_1^T & -v_1\mathbf{P}_1^T \end{matrix}} \\ \begin{matrix} \vdots & \vdots & \vdots \\ \mathbf{P}_N^T & 0 & -u_N\mathbf{P}_N^T \\ 0 & \mathbf{P}_N^T & -v_N\mathbf{P}_N^T \end{matrix} \end{pmatrix} \boxed{\begin{pmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \mathbf{t}_3 \end{pmatrix}} = 0$$

Parameters of transformation

✓ EPnP (indirect, two-step)

$$\begin{cases} \sum_{j=1}^{4}\left(\alpha_{ij}f_x x_j^c + \alpha_{ij}\left(c_x - u_i\right)z_j^c\right) = 0 \\ \sum_{j=1}^{4}\left(\alpha_{ij}f_y y_j^c + \alpha_{ij}\left(c_y - v_i\right)z_j^c\right) = 0 \end{cases} \Rightarrow \mathbf{M}\mathbf{x} = 0$$

$2n \times 12$

Coordinates of control points

$$\boxed{\begin{matrix} \begin{bmatrix} x_j^c \\ y_j^c \\ z_j^c \end{bmatrix} \\ \cdots \end{matrix}} \quad \mathbf{c}_j, \quad j = 1, \cdots, 4$$

# Explanation for Linear Systems of PnP

- Use Redundant Points to Improve Accuracy

- ✓ If we have prior knowledge that all the correspondences are inliers, we can use all the correspondences to generate an **over-determined** linear system.

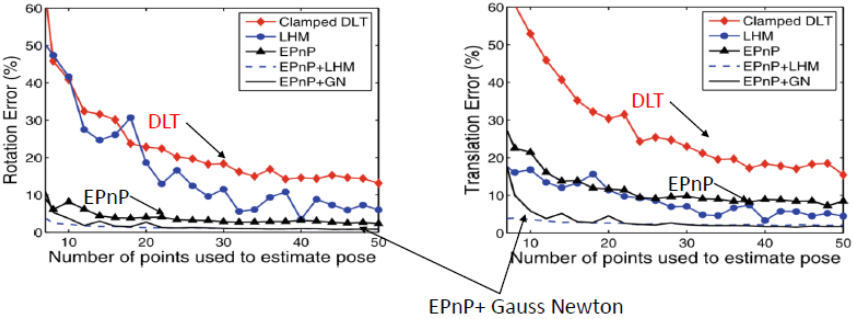
- ✓ The result is the least-squared solution.

- ✓ It is helpful for noise compensation.

# Explanation for Linear Systems of PnP

- ➢ Experimental Illustration of Redundant Case

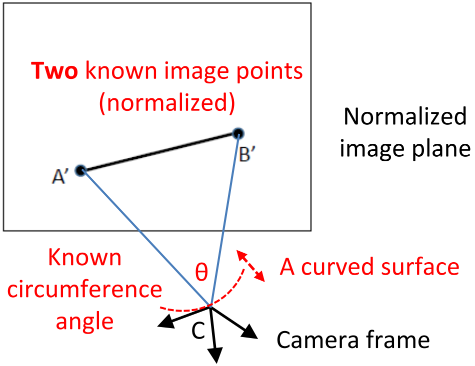
✓ The more inlier points we use, the higher the algorithm accuracy is

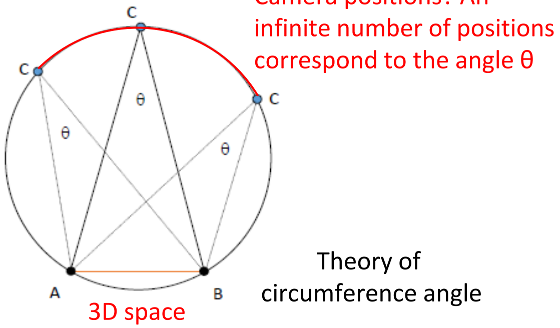


# Explanation for 2-Point Configuration

➢ Recap on Our Analysis Method

✓ Compute circumference angle based on the normalized image points.
✓ Find the optimal camera center satisfying the constraint of circumference angle.



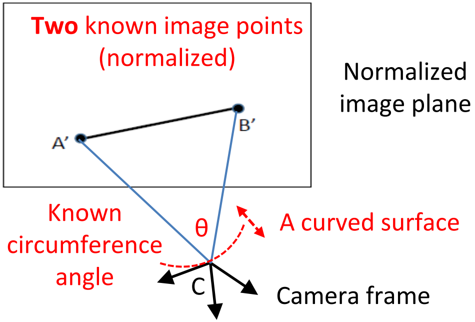



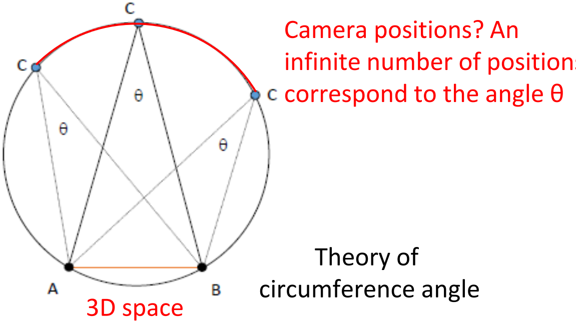
# Explanation for 2-Point Configuration

➢ Recap on Our Analysis Method

✓ Can we enforce the constraint of distance (focal length)?

- No. We do **not know image plane**. We can treat image plane and camera center as a whole part.
- The angle is computed based on image points, but we should consider the relationship between 3D point and camera center (see right figure).



# Today's Outline

- Overview of 3D-3D Geometry
- Non-iterative Method: SVD-based Method
- Iterative Method: Iterative closest point (ICP)

# Overview of 3D-3D Geometry

➢ Problem formulation

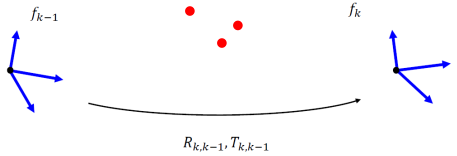
In essence, the following two types of formulations are equivalent.

✓ First type: *N* points in both first and second coordinate systems

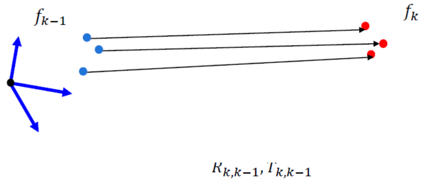
Example: in **EPnP**, four control point are static. We aim to determine their coordinate in both world frame and camera frame.

✓ Second type: *N*+*N* points in a single coordinate system

Example: Point set moves in a single coordinate system.




First type



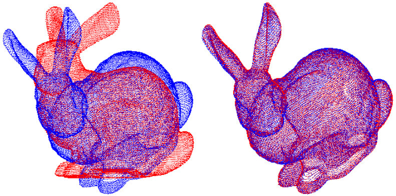

Second type

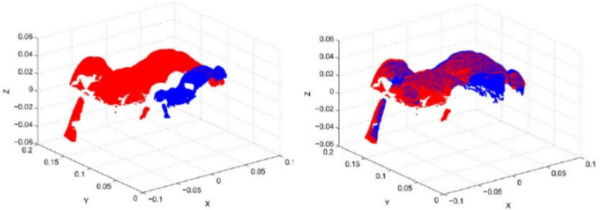
# Overview of 3D-3D Geometry

- Two Sub-problems

✓ 3D-3D Correspondence Establishment
✓ Transformation Estimation
- Case of SE(3)
- Case of Sim(3)

$$\mathrm{SE}(3) \quad \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad \Rightarrow \quad \mathrm{Sim}(3) \quad \mathbf{T}_S = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$$





# Overview of 3D-3D Geometry

## ➢ Intuitive Illustration

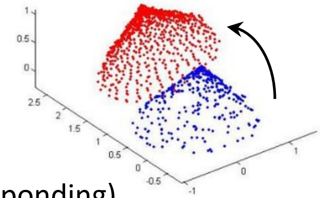
Motion estimation from 3D-to-3D feature correspondences (also known as point cloud registration problem)

- ✓ Input: Two point sets $f_{k-1}$ and $f_k$ in 3D. They are obtained by triangulation or stereo vision. They can also be virtual points (e.g., control points in EPnP).
- ✓ The minimal-case solution involves **three** 3D-3D point correspondences.
- ✓ Solving the following system of equations w.r.t. unknown R and T:

$$\begin{bmatrix} X^i{}_{k-1} \\ Y^i{}_{k-1} \\ Z^i{}_{k-1} \end{bmatrix} = \begin{bmatrix} \boxed{\begin{matrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{matrix}} & \boxed{\begin{matrix} t_1 \\ t_2 \\ t_3 \end{matrix}} \end{bmatrix} \cdot \begin{bmatrix} X^i{}_k \\ Y^i{}_k \\ Z^i{}_k \\ 1 \end{bmatrix}$$

where $i$ is the feature ID.

# Overview of 3D-3D Geometry



➢ Formal Definition

✓ Input: two point sets (we do not know which two points are corresponding)

$$X = \{x_1, ..., x_{N_x}\}$$
$$P = \{p_1, ..., p_{N_p}\}$$

Number of points are unnecessarily the same

✓ Goal: Find the optimal translation t and rotation R minimizing the sum of the squared error

$$E(R,t) = \frac{1}{N_p} \sum_{i=1}^{N_p} ||x_i - Rp_i - t||^2$$

Point to transform

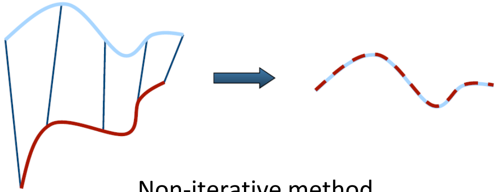
where $x_i$ and $p_i$ are **unknown-but-sought** corresponding points.

# Overview of 3D-3D Geometry

➢ Two Configurations

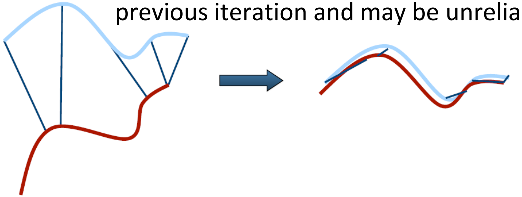
✓ If the correct correspondences are known, the correct rotation and translation can be calculated in closed form (non-iterative method).

✓ If the correct correspondences are not known, it is generally impossible to determine the optimal rotation and translation in one step. We have to perform **iterations**.



Non-iterative method
(closed-form method)

Correspondence is determined by the previous iteration and may be unreliable



Iterative method

# Overview of 3D-3D Geometry

➢ Comparison with 2D-2D Geometry

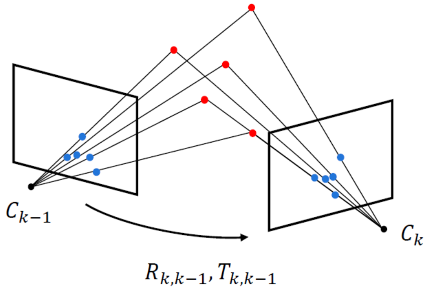
Motion estimation from 2D-to-2D feature correspondences

✓ Both feature correspondences $f_{k-1}$ and $f_k$ are in image coordinates (2D)

✓ The minimal case solution involves 5 feature correspondences

✓ Popular algorithms:

- 8-point algorithm
- 5-point algorithm



# Overview of 3D-3D Geometry

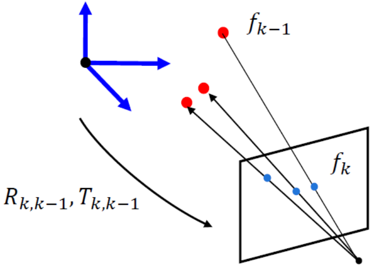
➢ Comparison with 3D-2D Geometry

Motion estimation from 3D-to-2D feature correspondences, i.e., Perspective-$n$-Points (PnP) problem)

✓ Feature $f_{k-1}$ is in 3D and feature $f_k$ in 2D

✓ Popular algorithms:

- DLT algorithm: at least 6 point correspondences
- P3P algorithm: minimal case with 3 point correspondences
- EPNP algorithm: at least 6 point correspondences



# Non-iterative Method

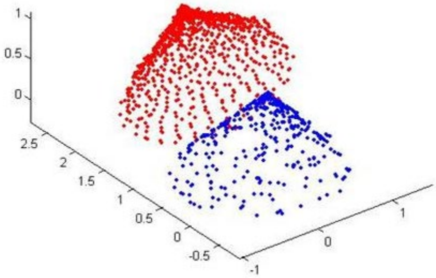
➢ SE(3)

This case is mainly introduced today

➢ Sim(3)

✓ Horn’s method [1]

✓ Umeyama’s method [2]




[1] Berthold K. P. Horn, “Closed-form solution of absolute orientation using unit quaternions,” in Journal of the Optical Society of America A, vol. 4, no. 2, pp. 629-642, 1987.
[2] Umeyama S. Least-squares estimation of transformation parameters between two point patterns. IEEE Trans Pattern Anal Mach Intell. 1991;13:376-380. doi:10.1109/34.88573.

# Non-iterative Method

➢ Preprocessing Step

✓ Computing center of mass

$$\mu_x = \frac{1}{N_x}\sum_{i=1}^{N_x} x_i \quad \text{and} \quad \mu_p = \frac{1}{N_p}\sum_{i=1}^{N_p} p_i$$

Here, we can simply assume that $N_x$=$N_p$

✓ Point set normalization

We subtract the corresponding **center of mass** from each point in the two point sets

$$X' = \{x_i - \mu_x\} = \{x'_i\}$$
$$P' = \{p_i - \mu_p\} = \{p'_i\}$$

We use the normalized point sets to calculate the transformation.

# Non-iterative Method

➢ Transformation Recovery

✓ Singular Value Decomposition

We compute matrix W by

$$W = \sum_{i=1}^{N_p} x'_i p'^T_i$$

We conduct the singular value decomposition (SVD) of W by:

$$W = U \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} V^T$$

where $\sigma_1 \geq \sigma_2 \geq \sigma_3$ are the singular values of W

# Non-iterative Method

➢ Transformation Recovery

✓ Computation of rotation and translation

The optimal solution of transformation is unique and is given by:

$$R = UV^T$$
$$t = \mu_x - R\mu_p$$

The conclusion is very precise, but how can we obtain this result? [1]

[1] “Least-Squares Fitting of Two 3-D Point Sets”, K. S. Arun, T. S. Huang, and S. D. Blostein

# Non-iterative Method

➢ Derivation Behind Conclusion

$$R = UV^T$$
$$\boxed{t = \mu_x - R\mu_p}$$

Previous conclusion

Due to limited, only some key steps are provided.

Center of mass

$$E(R,t) = \sum_{i=1}^{n} ||y_i - Rx_i - t||^2$$
$$= \sum_{i=1}^{n} ||y_i - Rx_i - t - y_o + y_o - Rx_o + Rx_o||^2$$

...

$$= \sum_{i=1}^{n} ||y_i - y_o - R(x_i - x_o)||^2 + n||y_o - Rx_o - t||^2$$

This part is only w.r.t R

Independent from specific points.
We can force this part to be 0. After obtaining R, we can obtain t

# Non-iterative Method

➢ Derivation Behind Conclusion

Due to limited, only some key steps are provided.

$$R^* = \arg\min_R \sum_{i=1}^{n} \|y_i - y_o - R(x_i - x_o)\|^2$$

$$= \arg\min_R \sum_{i=1}^{n} \|y'_i - Rx'_i\|^2$$ Normalized points

$$= \arg\min_R \sum_{i=1}^{n} \left(y_i'^T y'_i + x_i'^T \boxed{R^T R} x'_i - 2y_i'^T R x'_i\right)$$ Expansion

$$= \arg\min_R \sum_{i=1}^{n} \left(-2y_i'^T R x'_i\right)$$ Neglect the part independent from R

$$= \arg\max_R \sum_{i=1}^{n} \left(y_i'^T R x'_i\right)$$ Reformulate a minimization problem as a maximization problem

...

$$= \arg\max_R trace\left(R \sum_{i=1}^{n} \boxed{x'_i y_i'^T}\right)$$

$$W = \sum_{i=1}^{N_p} \boxed{x'_i p_i'^T}$$

$$W = U \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} V^T$$

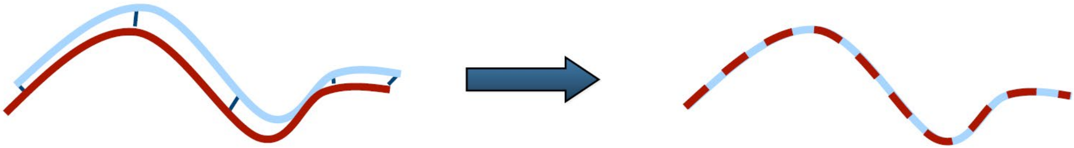
$$\boxed{R = UV^T}$$

Previous conclusion

# Iterative closest point (ICP)

➢ Overview

- ✓ Idea: Iteratively align two point sets
- ✓ Iterative Closest Points (ICP) algorithm [1]
- ✓ Converges if corresponding points are “close enough”

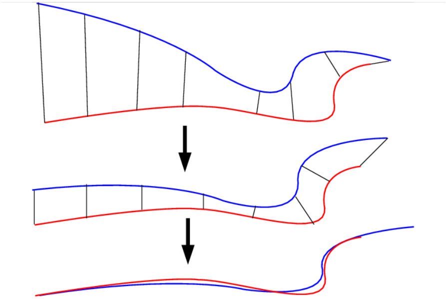


[1] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239-256, Feb. 1992

# Iterative closest point (ICP)

## ➢ Intuitive Illustration

- ✓ The major problem is to determine the correct data associations. We treat a pair of points with the smallest distance as a “temporary” 3D-3D correspondence.
- ✓ Given the associated points, the transformation can be computed efficiently using SVD.



A set of points is chosen along each line.
One point set (blue) is iteratively transformed to minimize the distance between each pair of points.

# Iterative closest point (ICP)

➢ Detailed Procedures

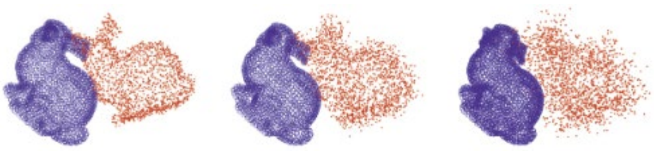
✓ Determine corresponding points based on the smallest distance
✓ Compute rotation R, translation t via SVD
✓ Apply R and t to the points of the set to be registered
✓ Compute the error E(R,t)
✓ If error decreased and error > threshold
- Repeat these steps
- Stop and output final alignment, otherwise

# Iterative closest point (ICP)

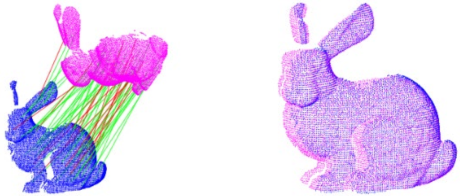
➢ Variants

✓ Several improvements have been proposed at different stages:

- Weighting the correspondences (mainly for high accuracy)
- Rejecting outlier point pairs (mainly for high robustness)



Some inlier correspondences are noisy. They should be assigned relatively small weights.



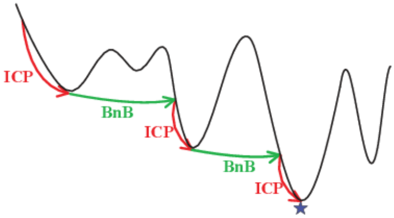
Outliers must be removed to correctly align point sets

# Iterative closest point (ICP)

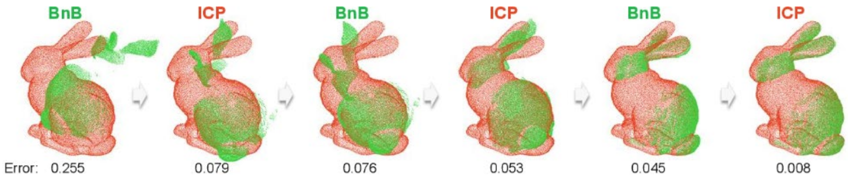
➢ Variants

✓ Several improvements have been proposed at different stages:

- Jump out of local minima based on global search method, i.e., branch-and-bound (BnB) (mainly for stability).
- Combine ICP and BnB to improve the efficiency of pure BnB.




Error evolution




Transformation of green point set (red point set remain unchanged)

# Summary

- ➢ Overview of 3D-3D Geometry
- ➢ Non-iterative Method: SVD-based Method
- ➢ Iterative Method: Iterative closest point (ICP)

Thank you for your listening!

If you have any questions, please come to me :-)