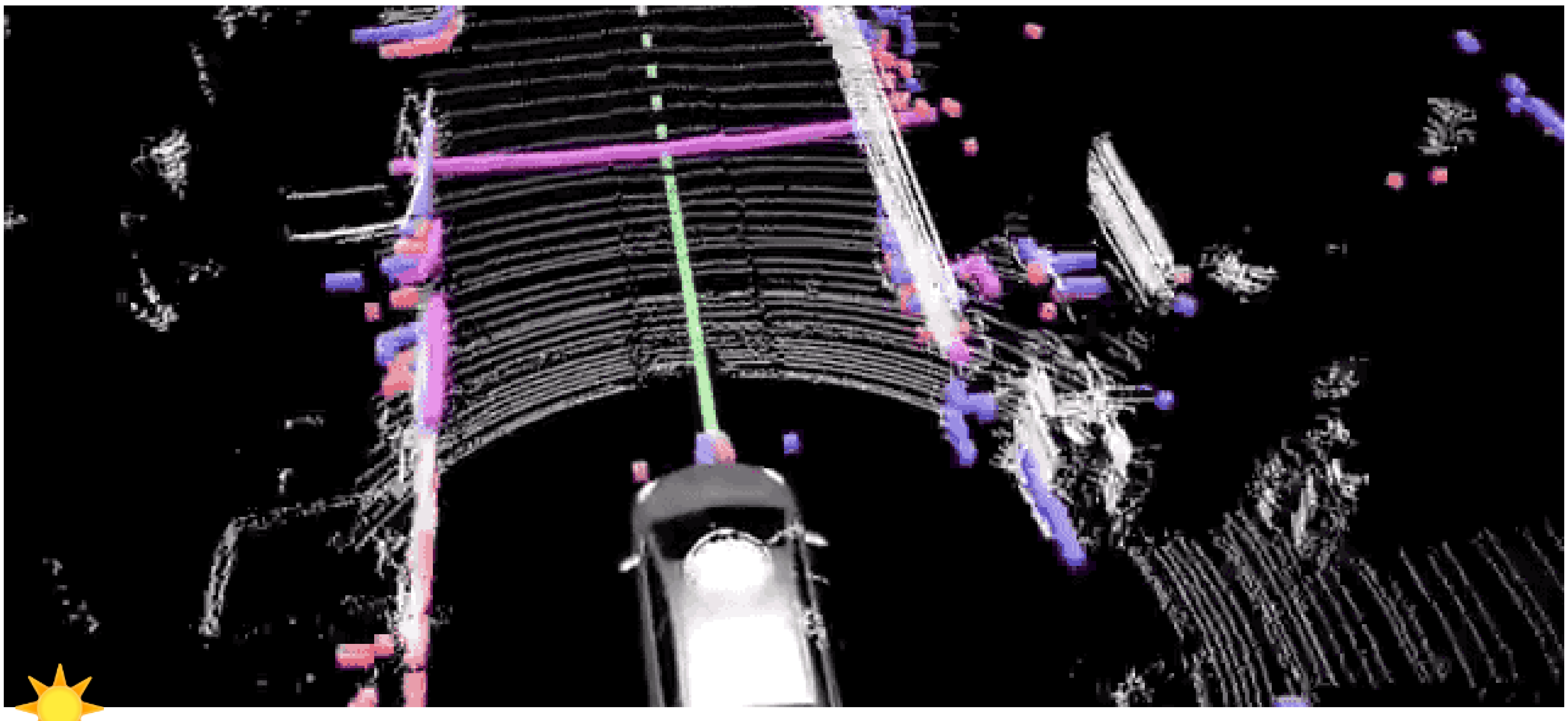


Point Cloud Registration/Iterative Closest Point (ICP) (3D-3D)

Comments	
Dates Taught	@September 1, 2020
Lecture No.	L6
Links of Videos	L6
Module	SLAM: Mapping
Related to All Questions (Property)	

> Partial Lecture Notes for Sep 1. Please read this note first.

Resources

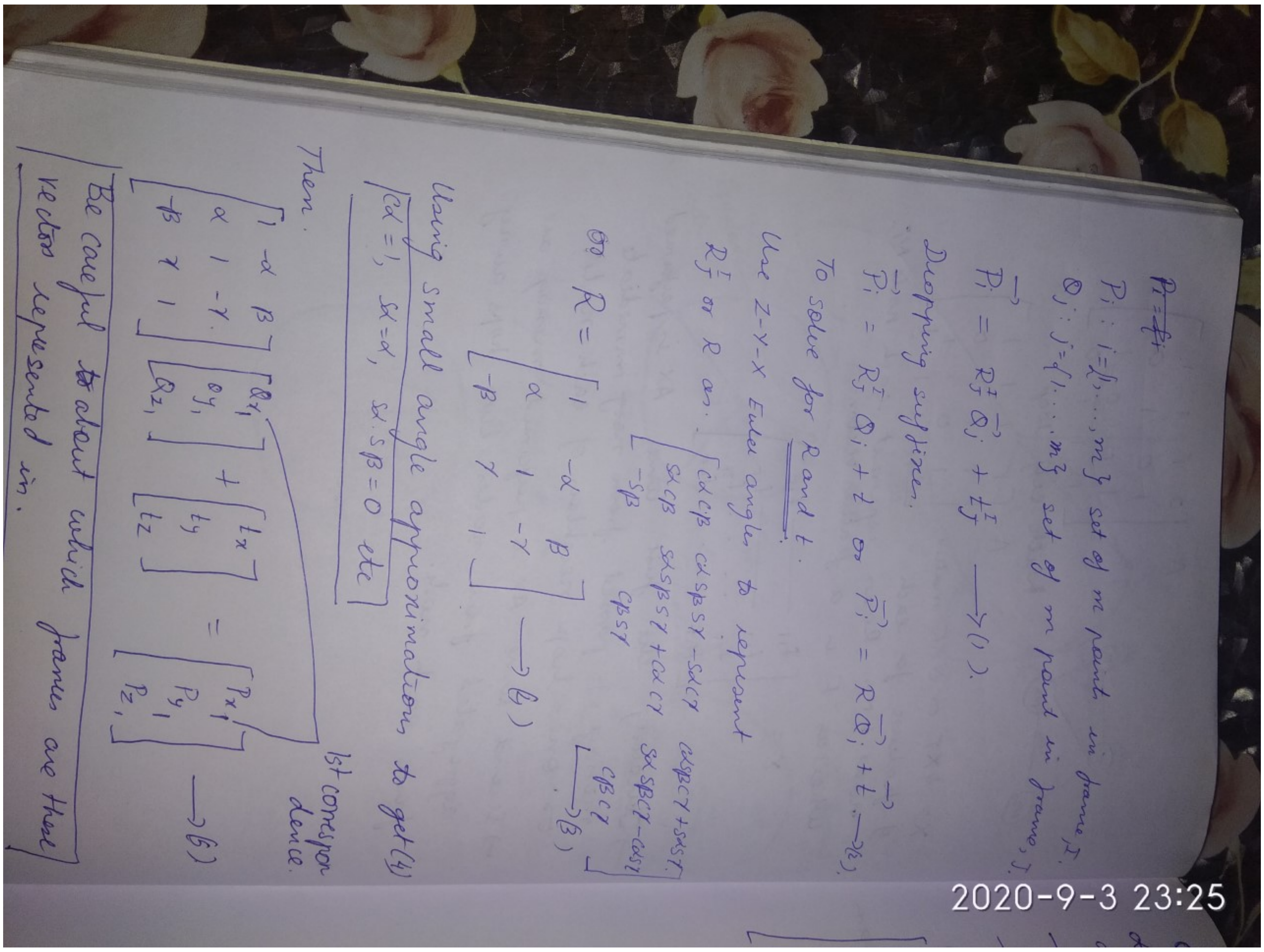
▼ Resources

- Cyrill Stachniss' Lectures: Slides

Prof Madhav's ICP Notes

▼ ICP-LS Derivation

The following two images only work for very small angles, for example when a robot is moving. You can apply this incrementally.



Proof.

$P_i : i = \{1, \dots, m\}$ set of m point in frame I.

$Q_j : j = \{1, \dots, m\}$ set of m point in frame J.

$\vec{P_i} = R_J^I \vec{Q_j} + t_J^I \longrightarrow (1)$.

Dropping suffixes,

$\vec{P_i} = R_J^I Q_i + t$ or $\vec{P_i} = R \vec{Q_i} + \vec{t} \longrightarrow (2)$.

To solve for R and t.

Use Z-Y-X Euler angles to represent R_J^I or R as:

$$\begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix} \longrightarrow (3)$$

So $R = \begin{bmatrix} 1 & -\alpha & \beta \\ \alpha & 1 & -\gamma \\ -\beta & \gamma & 1 \end{bmatrix} \longrightarrow (5)$

Using small angle approximation to get (4)

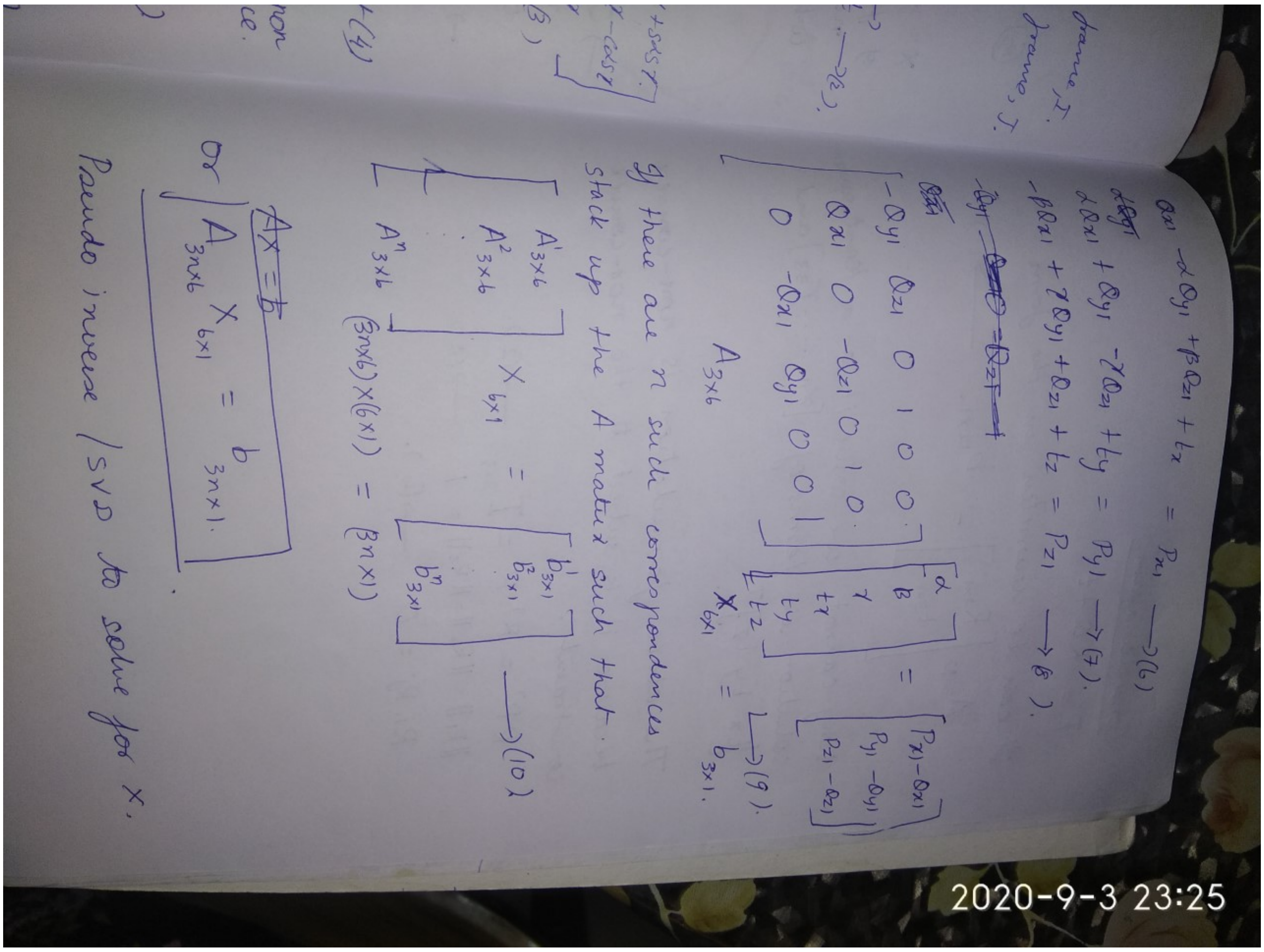
$\boxed{c\alpha = 1,\ s\alpha = \alpha,\ s\alpha . s\beta = 0 \text{ etc}}$

Then.

$$\boxed{\begin{bmatrix} 1 & -\alpha & \beta \\ \alpha & 1 & -\gamma \\ -\beta & \gamma & 1 \end{bmatrix} \begin{bmatrix} Q_{x_1} \\ Q_{y_1} \\ Q_{z_1} \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}} = \begin{bmatrix} P_{x_1} \\ P_{y_1} \\ P_{z_1} \end{bmatrix} \longrightarrow (6)$$

1st correspondence.

Be careful about which frames are these vectors represented in.



$$Q_{x1} - \alpha Q_{y1} + \beta Q_{z1} + t_x = P_{x1} \longrightarrow (6)$$

$$\alpha Q_{x1} + Q_{y1} - \gamma Q_{z1} + t_y = P_{y1} \longrightarrow (7).$$

$$-\beta Q_{x1} + \gamma Q_{y1} + Q_{z1} + t_z = P_{z1} \longrightarrow (8).$$

$$\underbrace{\begin{bmatrix} -Q_{y1} & Q_{z1} & 0 & 1 & 0 & 0 \\ Q_{x1} & 0 & -Q_{z1} & 0 & 1 & 0 \\ 0 & -Q_{x1} & Q_{y1} & 0 & 0 & 1 \end{bmatrix}}_{A_{3\times 6}} \underbrace{\begin{bmatrix} \alpha \\ \beta \\ \gamma \\ t_x \\ t_y \\ t_z \end{bmatrix}}_{X_{6\times 1}} = \underbrace{\begin{bmatrix} P_{x1} - Q_{x1} \\ P_{y1} - Q_{y1} \\ P_{z1} - Q_{z1} \end{bmatrix}}_{b_{3\times 1}} \longrightarrow (9).$$

If there are n such correspondences stack up the A matrix such that

$$\begin{bmatrix} A^1_{3\times 6} \\ A^2_{3\times 6} \\ \vdots \\ A^n_{3\times 6} \end{bmatrix} X_{6\times 1} = \begin{bmatrix} b^1_{3\times 1} \\ b^2_{3\times 1} \\ \vdots \\ b^n_{3\times 1} \end{bmatrix} \longrightarrow (10)$$

$$(3n\times 6)\times(6\times 1) = (3n\times 1)$$

$\cancel{Ax = b}$

Or $\boxed{A_{3n\times 6}\, X_{6\times 1} = b_{3n\times 1}}$.

Pseudo inverse / SVD to solve for X.

2020-9-3 23:25

The following image says that if you just rearrange the terms and express it as $Ax - b$, you cannot do it as simple least squares because x is on manifold.

Consider again:

$$P = RQ + t$$

It can be rewriten in the form

$$A_{3\times 12}\begin{bmatrix} R_{9\times 1} \\ t_{3\times 1} \end{bmatrix} = b_{12\times 1}. \longrightarrow (1)$$

Why cannot we solve it as a least squares solution in terms of $[r_{11}, \dots, r_{33}]$ and $[t_x \; t_y \; t_z]^T$.

The above formulation is non-convex because it is subject to the non-convex constraint:

$$RR^T = R^T R = I_{3\times 3} \longrightarrow (2).$$

$$\|R_1\| = \|R_2\| = \|R_3\| = 1 \longrightarrow (3)$$

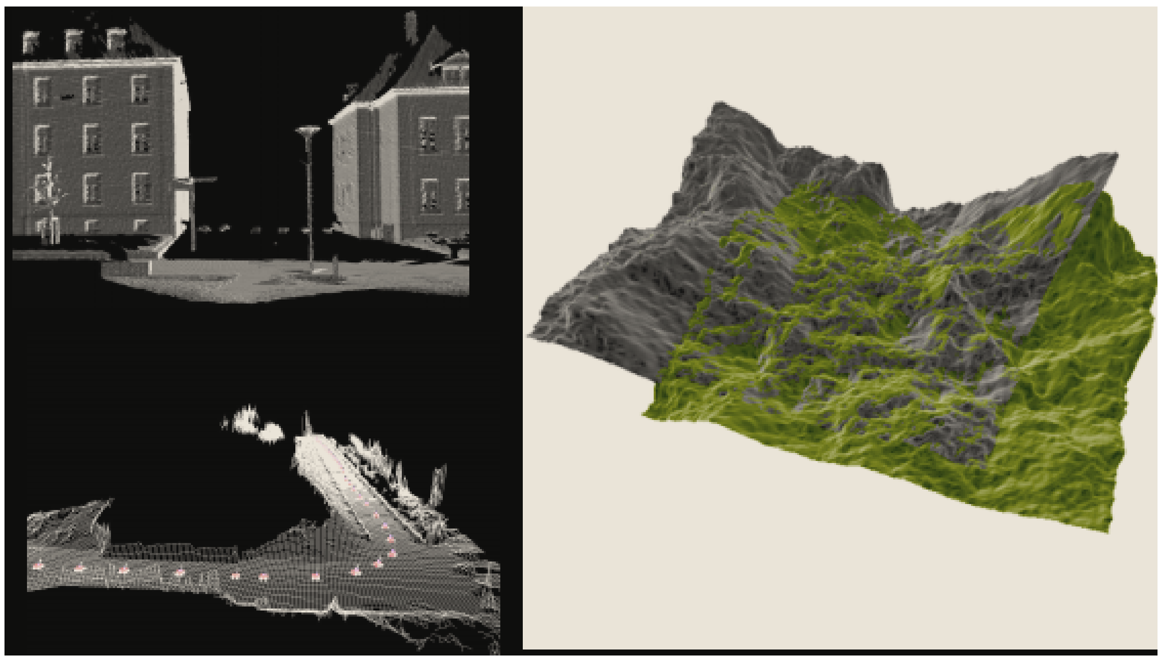
$$R_i . R_j = 0 \longrightarrow (4).$$

2020-9-3 23:25

1. Introduction

- Find the parameters of the transformation that best align corresponding data points
  - Least squares and robust least squares

- **Iterative Closest Point (ICP) — Very common and will be covered now.**



Given two corresponding point sets:

$$Q = \{q_1, \ldots, q_N\}$$
$$P = \{p_1, \ldots, p_N\}$$

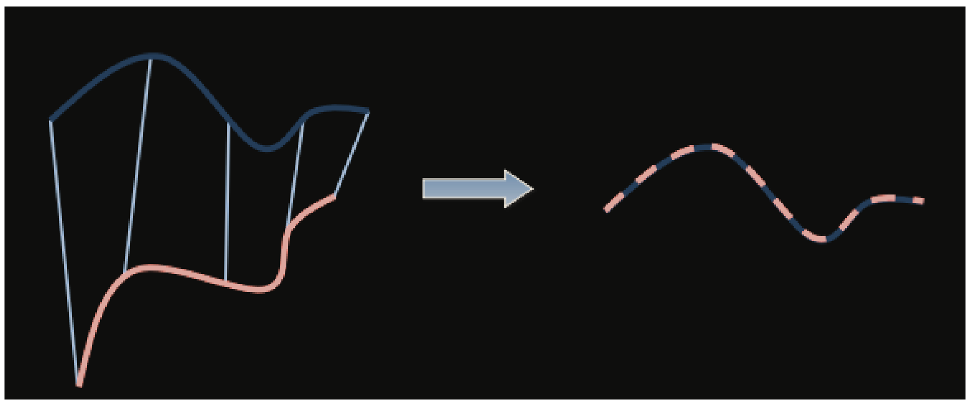
with correspondences $C = \{(i, j)\}$.

$\text{Wanted: Translation } t \text{ and rotation } R \text{ that minimize the sum of the squared errors:}$

$$E(R, t) = \sum_{(i,j)\in C} \left\| \boldsymbol{q}_i - R\boldsymbol{p}_j - t \right\|^2$$

subject to $\mathbf{R}\mathbf{R}^T = \mathbf{I}$

Note: As of now, i and j are being used to denote that they refer to q and p respectively, but it is unnecessarily confusing to use both and it is redundant notation. So below, let us use only i instead of using both i and j.



If the correct correspondences are known, the correct relative rotation/translation can be calculated in closed form.

2. ICP: Known Correspondences

[Besl & McKay 92]

- Shift/Translate via the center of mass
- Rotational Alignment

Center of Mass

The centers of mass of the corresponding points in both sets

$$\boldsymbol{\mu}_Q = \frac{1}{|C|} \sum_{(i)\in C} \boldsymbol{q}_i \quad \text{and} \quad \boldsymbol{\mu}_P = \frac{1}{|C|} \sum_{(i)\in C} \boldsymbol{p}_i$$

Idea

- Subtract the corresponding center of mass from every point before calculating the transformation, normalized point:

$$
\begin{aligned}
Q' &= \{\boldsymbol{q}_i - \boldsymbol{\mu}_Q\} = \{\boldsymbol{q}'_i\} \\
P' &= \{\boldsymbol{p}_i - \boldsymbol{\mu}_P\} = \{\boldsymbol{p}'_i\}
\end{aligned}
$$

Orthogonal Procrustes Problem

- Original Error Function

$$E(R,t) = \sum_{(i)\in C} \|\boldsymbol{q}_i - R\boldsymbol{p}_i - t\|^2$$

- Equivalent to minimizing the "Orthogonal Procrustes" problem

$$E'(R) = \|\,[\boldsymbol{q}'_1 \dots \boldsymbol{q}'_n] - R\,[\boldsymbol{p}'_1 \dots \boldsymbol{p}'_n]\,\|_F^2$$

  Note that in $\| \dots \|_F$, F stands for Frobenius Norm.

- Solved through SVD ——

▼ Question) (Added as a part of Project 1)

  Prove that minimizing the original error function is equivalent to minimizing the "Orthogonal Procrustes" problem.

Singular Value Decomposition

Define cross-covariance matrix

$$W = \sum_{(i)\in C} q'_i p'^T_i$$

Compute SVD of W:

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

Theorem

If $\operatorname{rank}(W) = 3$, the optimal solution of $\mathrm{E}(R,t)$ is unique and is given by:

▼ Made blog post here for questions like:

- What if rank is not 3?
- What is min number of point correspondenses?

$$
\begin{aligned}
R &= UV^T \\
t &= \mu_Q - R\mu_p
\end{aligned}
$$

The minimal value of error function at (R,t) is:

$$E(R,t) = \sum_{i=1}^{N_p} \left(\|x'_i\|^2 + \|y'_i\|^2 \right) - 2\,(\sigma_1 + \sigma_2 + \sigma_3)$$

Algorithm

- Form the cross-covariance matrix

$$W = \sum q'_i p'_i$$

- Compute SVD

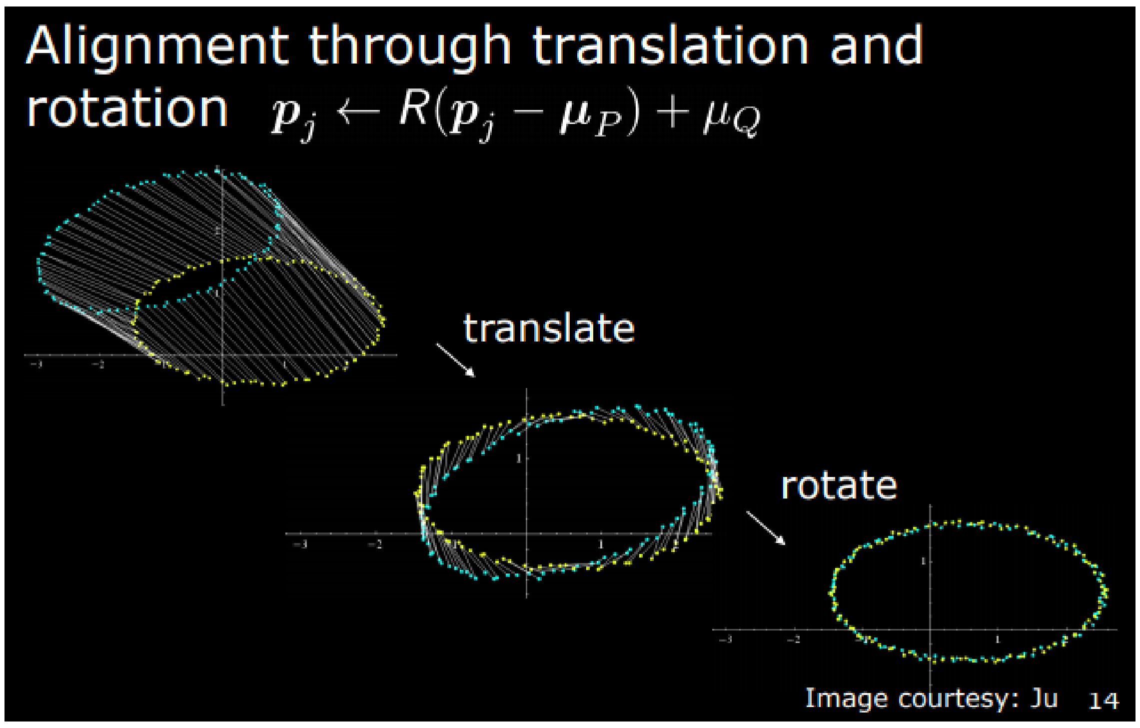
$$W = UDV^\top$$

- The rotation matrix is

$$R = UV^{\top}$$

- Translate and rotate points:

$$p_j \leftarrow R\,(p_j - \mu_P) + \mu_Q$$



3. ICP: Unknown Correspondences

- Happens in most practical scenarios
- If the correct correspondences are not known, it is generally impossible to determine the optimal relative rotation and translation in one step.
- Converges if starting positions are "close enough"

ICP Variants

Variants on the following stages of ICP have been proposed:

1. Point subsets (from one or both point sets)
2. Weighting the correspondences
3. Data association (For example, occlusions, dynamic objects)
4. Rejecting certain (outlier) point pairs

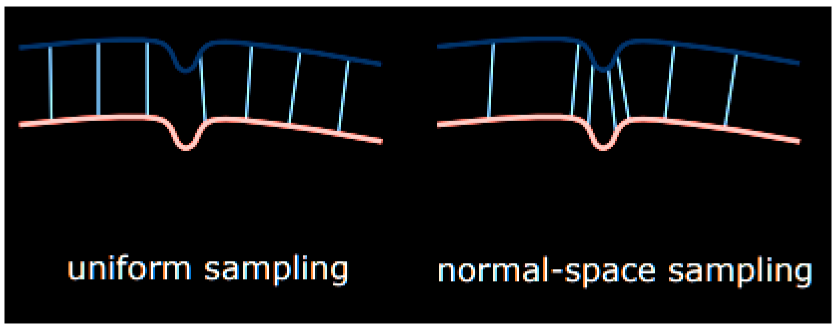
Various aspects of performance:

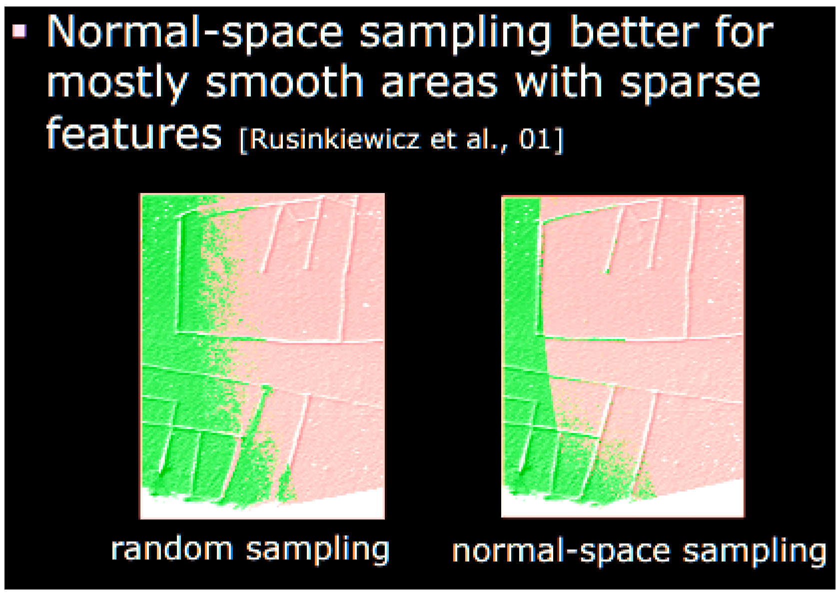
- Speed
- Stability (local minima)
- Tolerance w.r.t. noise and outliers
- Basin of convergence (maximum initial misalignment)

1. Point subsets

- Use all points
- Uniform sub-sampling (For example, points closer to sensor will be more, so do uniform samping to overcome that)
- Random sampling

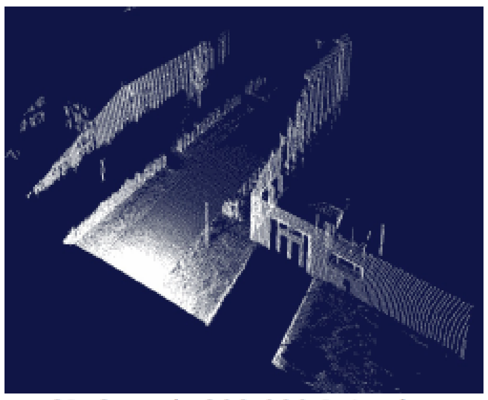
- Feature-based sampling (If some objects are more important)
- Normal-space sampling (Ensure that samples have normals distributed as uniformly as possible)



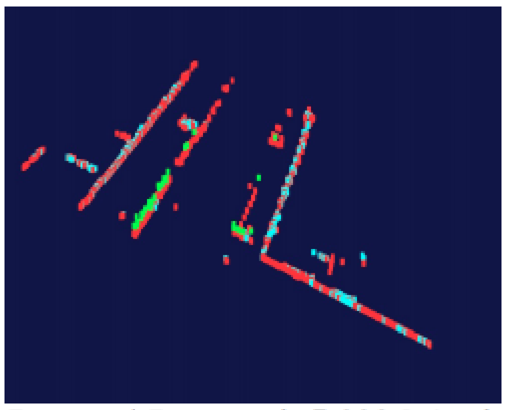


Feature Based Sampling

- Try to find "important" points
- Simplifies the search for correspondences
- Higher efficiency and higher accuracy
- Requires preprocessing



3D Scan (~200.000 Points)



Extracted Features (~5.000 Points)

ICP with Uniform Sampling

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/d3016532-1181-406b-bf64-99892ca7b193/ICP_uniform_sampling.mp4

Video Credits: Nuechter

2. Weighting the correspondences

Re-weighting

- Weight the corresponding pairs
- Noise: Weighting based on sensor uncertainty, say Give higher weight to points that are closer.
- Outlier: Assign lower weights for points with higher point-point distances
- Determine transformation that minimizes the weighted error function

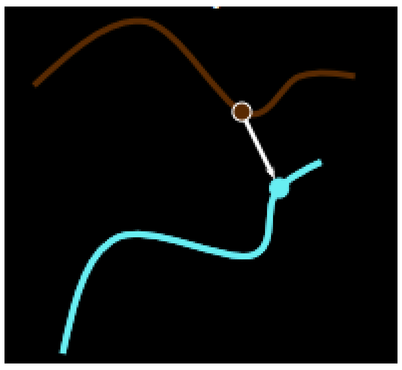
3. Data Association

- Has greatest effect on convergence and speed

Matching methods

- **Closest** point
- Normal shooting
- Closest compatible point
- **Point-to-plane**
- Projection-based
- More.....

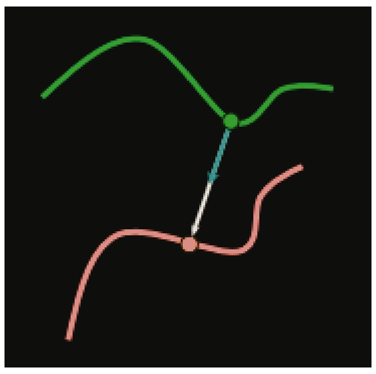
1. **Closest-point matching**
    - Find closest point in other the point set (using kd-trees)
    - Generally stable, but slow convergence and requires preprocessing



Closest point

2. Normal Shooting

- Project along normal, intersect other point set
- Slightly better convergence results than closest point for smooth structures, worse for noisy or complex structures



Normal Shooting

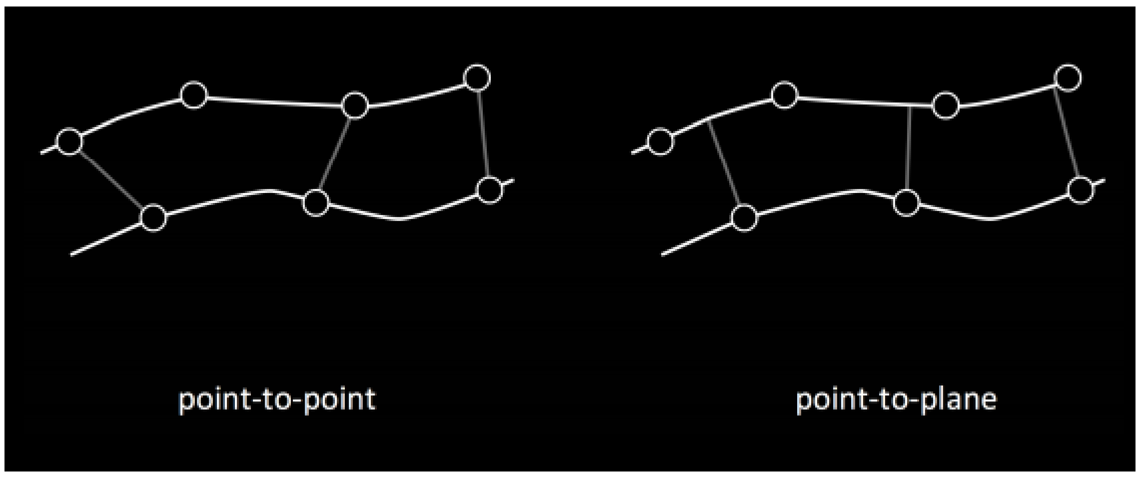
3. Closest compatible point

- Robustification by considering the compatibility of the points
- Only matches compatible points
- Compatibility can be based on
    - Normals
    - Colors
    - Curvature
    - Higher-order derivatives
    - Other local features

4. Point-to-plane Error Metric

Minimize the sum of the squared distances between a point and the tangent plane at its correspondence point [Chen & Medioni 91].

Point-to-Point vs *Point-to-Plane*



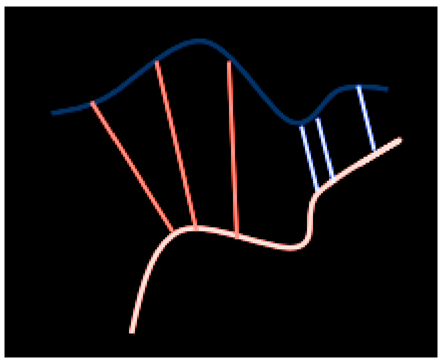
Think in the case of sparse samping. If you do p2point, there would still be residual errors...

- Each iteration generally slower than the point-to-point version, however, often significantly better convergence rates [Rusinkiewicz01]
- Using point-to-plane distance instead of point-to-point lets **flat regions slide along each other** [Chen & Medioni 91]

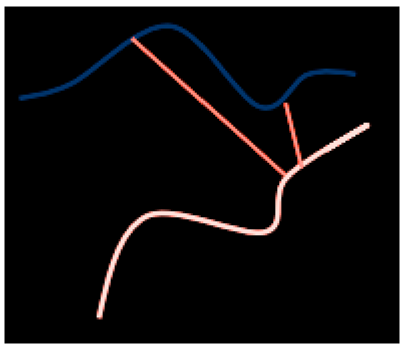
4. Rejecting certain (outlier) point pairs

(Heuristic Approaches)

- Point-to-point distance larger than a given threshold
- Rejection of pairs that are not consistent with their neighboring pairs [Dorai 98]



Ignore red ones.

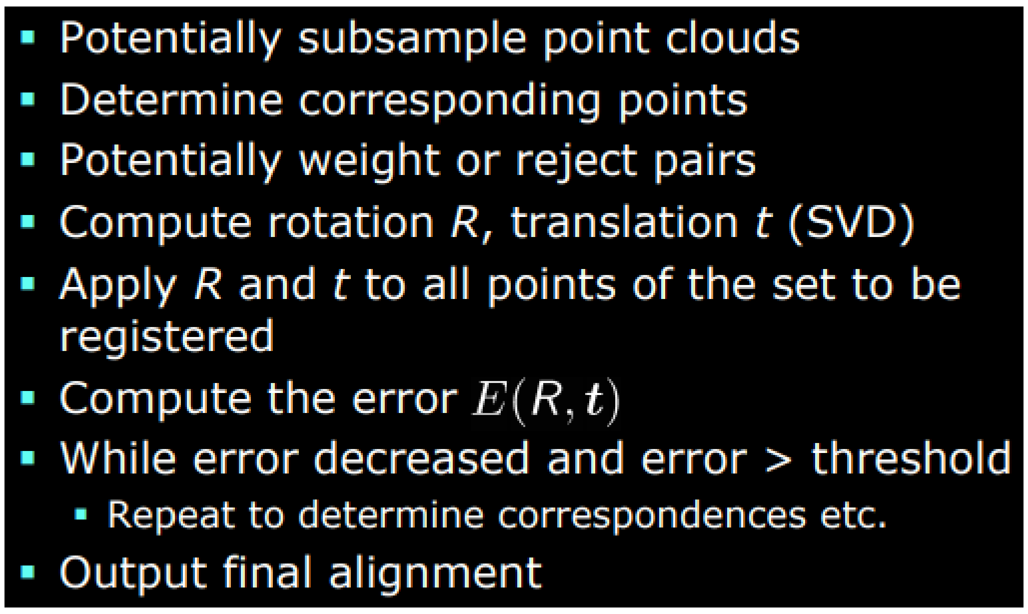


Two nearby points in one scan aren't in other → reject

- Trimmed ICP: Sort correspondences w.r.t. their error, ignore the worst $t\%$ [Chetverikov et al. 02]
  - t is related to overlap and outlier ratio
  - Knowledge about the overlap has to be estimated

More approaches include RANSAC, kernel functions etc.

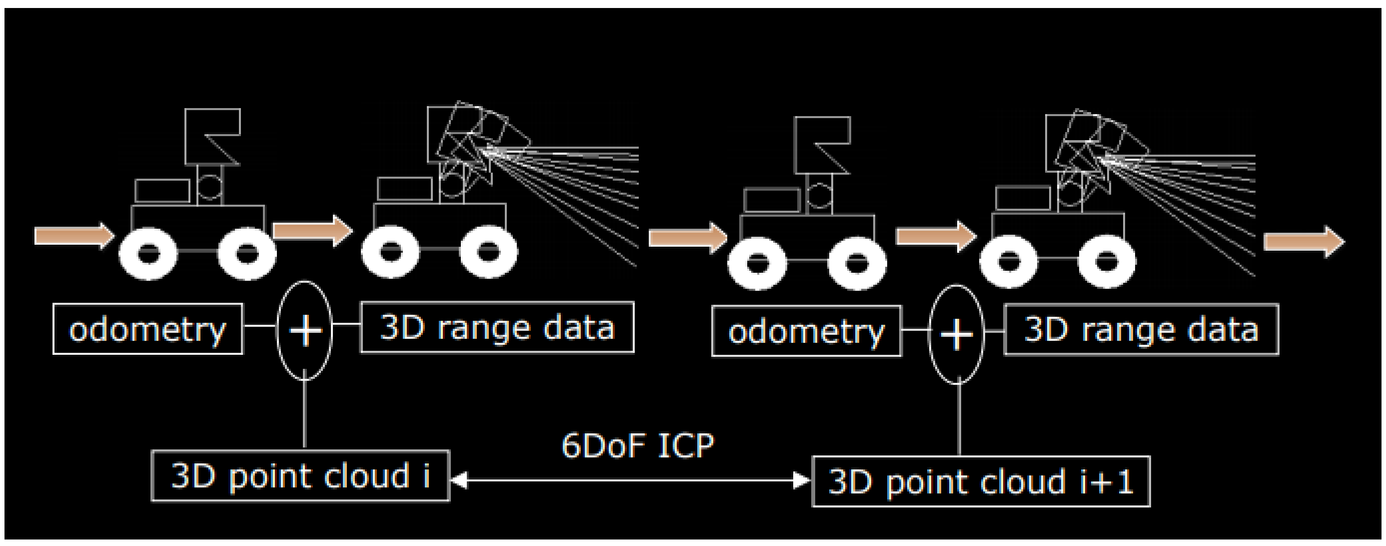
ICP Algorithm



There are much complex implementations which use nonlinear optimizations.

4. Common ICP Applications

- Laser scan matching
- Range image matching



Given odometry as initial guess, using ICP I can estimate a better transformation.

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/24de46d8-1e3a-4826-b8d3-73aa806a1f1c/ICP_auto_car.mp4

For self driving cars...

Summary

- Alignment of 2D and 3D data points is an important task in perception
- Gold standard algorithm for calculating the transform between scans
- Estimates translation and rotation between the scans
- Given the correct data associations, the transformation can be computed efficiently using SVD
- The major problem is to determine the correct data associations
- Initial guess is needed for data association
- Iterative approach
- Several variants exist
- In practice, ICP does not *always* converge to the correct solution

Open-source implementations

- Libpointmatcher
- Open3D
- Coherent Point Drift (CPD)