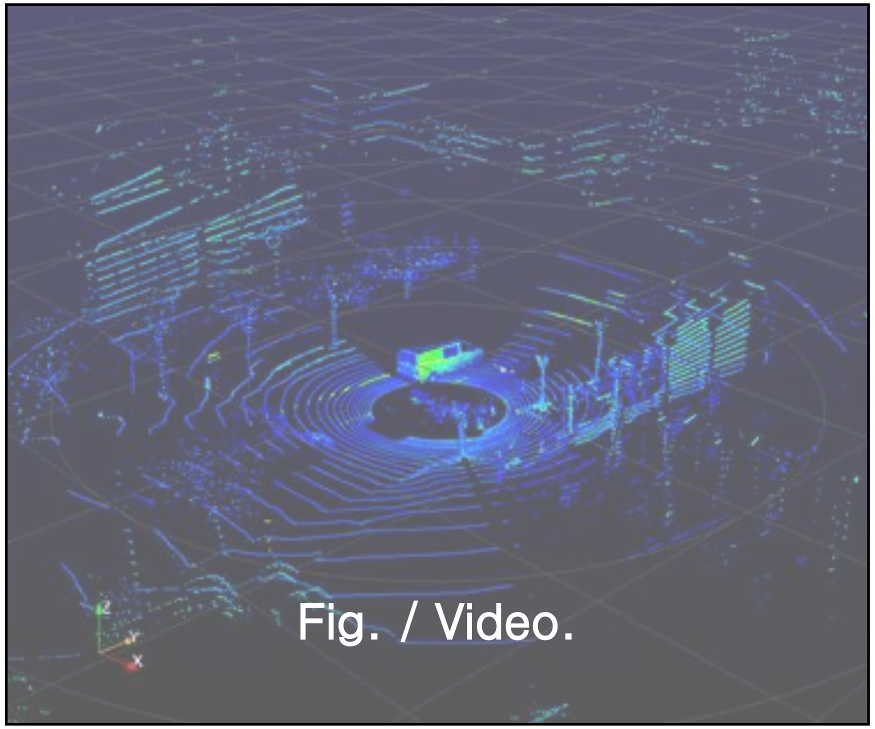
Point Cloud Analysis

2023. 09. 17

최재우 (allyouneed3002@gmail.com)

Content

- **Segmentation**
- **Registration**
  - Known Data Association
  - Unknown Data Association



Caption & Ref. (Link)

[Ref] Shan Liu, Min Zhang, Pranav Kadam, C.-C. Jay Kuo - 3D Point Cloud Analysis_ Traditional, Deep Learning, and Explainable Machine Learning Methods-Springer (2021)

Segmentation

Edge Based

Introduction:

Edge-based segmentation places emphasis on detecting discontinuities within the 3D point cloud. These discontinuities typically align with object boundaries or significant shifts in surface orientation.

Principle:

- It employs gradient computation. A pronounced gradient magnitude signifies an edge or boundary.
- A common approach involves calculating the difference between the normals of adjacent points. A notable difference indicates an edge.

Pros & Cons:

Pros:

- Directly targets the boundaries of objects.
- Can be computationally efficient for specific applications.

Cons:

- It's sensitive to noise, leading to the potential creation of false edges.
- Post-processing might be necessary to refine the detected edges.

Segmentation

Region Based

Introduction:

Region-based segmentation focuses on grouping points that share similar characteristics, thereby forming coherent regions within the point cloud.

Principle:

- Commonly employs clustering algorithms such as K-means or DBSCAN.
- Also utilizes region-growing techniques, where seeds are expanded by adding neighboring points that share similar attributes.

Pros & Cons:

Pros:

- Generally robust against noise.
- Can produce smooth and consistent segments.

Cons:

- Might merge distinct objects if they possess similar attributes.
- The choice of similarity metric is crucial.

Segmentation

Attributes Based

Introduction:

Attributes-based segmentation segments the point cloud based on various point attributes such as color, density, curvature, or other derived features.

Principle:

- Involves feature extraction from the point cloud.
- Segmentation algorithms then group points based on these extracted features.

Pros & Cons:

Pros:

- Can capture subtle differences, allowing for detailed segmentation.
- Offers versatility as different attributes can be chosen for various tasks.

Cons:

- Requires careful selection and possibly normalization of attributes.
- Might be computationally intensive due to feature extraction.

Segmentation

Model Based

Introduction:

Model-based segmentation involves segmenting the point cloud using predefined models or templates. This approach is especially beneficial when the shapes of interest are known in advance.

Principle:

- Methods such as RANSAC are employed to fit predefined models to the point cloud data.
- Points that align well with a model are segmented as one group.

Pros & Cons:

Pros:

- Offers high accuracy for known shapes.
- Can directly provide shape parameters (e.g., radius for spheres).

Cons:

- Limited to segmenting known shapes.
- Can be computationally intensive, especially when dealing with multiple models.

Segmentation

Graph Based

Introduction:

Graph-based segmentation represents the point cloud as a graph, where nodes correspond to points and edges signify relationships between points, such as spatial proximity.

Principle:

- Graph-cut algorithms, like normalized cuts, are used to partition the graph into segments.
- The goal is often to minimize intra-segment differences while maximizing inter-segment differences.

Pros & Cons:

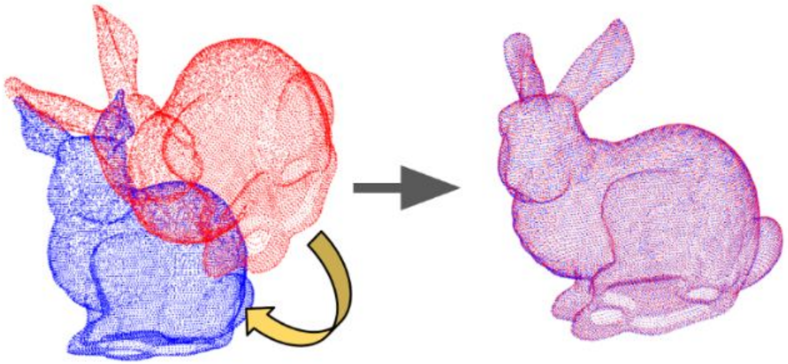
Pros:

- Capable of handling large datasets.
- Often produces segments that respect object boundaries.

Cons:

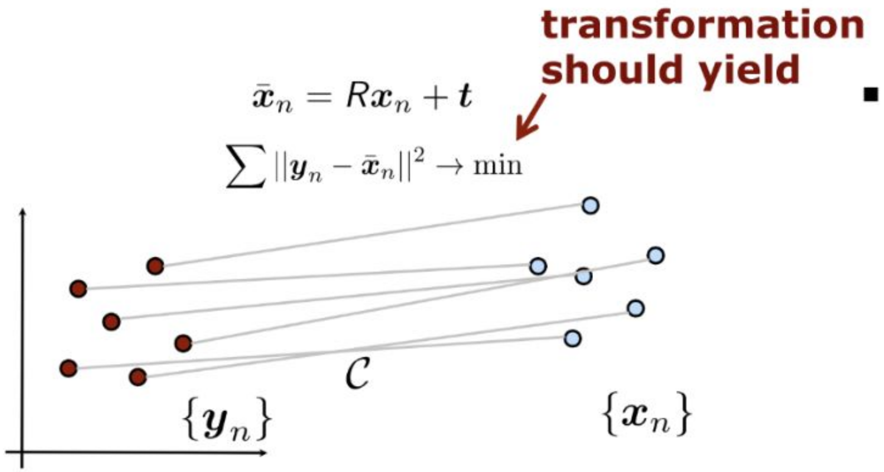
- Constructing and processing the graph can be computationally intensive.
- The choice of graph structure and edge weights is crucial.

Point Cloud Registration



- **Definition:** Find the spatial transformation that aligns the two Point Clouds.
- **Goal:** Find the best aligning Rotation Matrix and Translation Vector.
- **Method**:
    - Known Data Association.
    - Unknown Data Association.

Known Data Association



- **Given two point sets:**
  - X = $\{X_1, \ldots, X_n\}$ Y= $\{Y_1,\ldots,Y_n\}$ with correspondences C={(i, j)
  - Rotation matrix R and Translation Vector t can be utilized to align two Point Clouds can be aligned.
  - Determine R and t that minimize the Euclidean distance.

Know Data Association

- **Absolute Orientation Problem**
  - Aligning sets of 3D points and determining their transformation.
  - Scale parameter is fixed at 1.

- **Solution without Initial Guess**
  - When correspondences are known, a perfect solution can be found without needing an initial guess or iterations.
  - To determine the translation value, align the center of masses of the two Point Clouds and calculate the displacement.
  - To determine the rotation value, perform Singular Value Decomposition.

Known Data Association

$$\boldsymbol{x}_0 = \frac{\sum \boldsymbol{x}_n p_n}{\sum p_n} \qquad \boldsymbol{y}_0 = \frac{\sum \boldsymbol{y}_n p_n}{\sum p_n}$$

$$H = \sum (\boldsymbol{x}_n - \boldsymbol{x}_0)(\boldsymbol{y}_n - \boldsymbol{y}_0)^\top p_n$$

$$\mathrm{svd}(H) = UDV^\top$$

$$R = VU^\top$$

$$\boldsymbol{y}_0 = \frac{\sum \boldsymbol{y}_n p_n}{\sum p_n} \quad R = VU^\top \quad \boldsymbol{x}_0 = \frac{\sum \boldsymbol{x}_n p_n}{\sum p_n}$$

$$\boldsymbol{t} = \boldsymbol{y}_0 - R\boldsymbol{x}_0$$

Know Data Association

- **Optimal Solution Search**
  - Can find a perfect solution without an initial guess.
  - Can find the solution without iterations.
- **Redefinition & Simplification**
  - The original equation is redefined in a Local Coordinate System with origin y0.
  - Simplifies the optimization problem.

$$\sum ||\boldsymbol{y}_n - \bar{\boldsymbol{x}}_n||^2 p_n \rightarrow \min \qquad \boldsymbol{y}_0 = \frac{\sum \boldsymbol{y}_n p_n}{\sum p_n}$$

$$\sum ||\boldsymbol{y}_n - \boldsymbol{y}_0 - R\boldsymbol{x}_n - \boldsymbol{t} + \boldsymbol{y}_0||^2 p_n \rightarrow \min$$

does not change the problem

Know Data Association

Start with $\bar{\boldsymbol{x}}_n = R\boldsymbol{x}_n + \boldsymbol{t}$

and use the shift of the origin

$$\bar{\boldsymbol{x}}_n - \boldsymbol{y}_0 = R\boldsymbol{x}_n + \boldsymbol{t} - \boldsymbol{y}_0$$

to rewrite the translation vector

$$\bar{\boldsymbol{x}}_n - \boldsymbol{y}_0 = R(\boldsymbol{x}_n + \underline{R^\top \boldsymbol{t} - R^\top \boldsymbol{y}_0})$$

Introduce a **new variable** $\boldsymbol{x}_0$:

$$\bar{\boldsymbol{x}}_n - \boldsymbol{y}_0 = R(\boldsymbol{x}_n - \boldsymbol{x}_0)$$

with $\boldsymbol{x}_0 = R^\top \boldsymbol{y}_0 - R^\top \boldsymbol{t}$

- The initially formulated problem

$$\sum \|\boldsymbol{y}_n - \bar{\boldsymbol{x}}_n\|^2 p_n \to \min$$

- turns into

$$\sum \|\boldsymbol{y}_n - \boldsymbol{y}_0 - R(\boldsymbol{x}_n - \boldsymbol{x}_0)\|^2 p_n \to \min$$

- We need to find $R, \boldsymbol{x}_0$ so that

$$R^*, \boldsymbol{x}_0^* = \underset{R, \boldsymbol{x}_0}{\operatorname{argmin}} \sum \|\boldsymbol{y}_n - \boldsymbol{y}_0 - R(\boldsymbol{x}_n - \boldsymbol{x}_0)\|^2 p_n$$

Know Data Association

- Minimize the objective function

$$\Phi(\boldsymbol{x}_0, R) = \sum \left[(\boldsymbol{y}_n - \boldsymbol{y}_0) - R(\boldsymbol{x}_n - \boldsymbol{x}_0)\right]^\top \left[(\boldsymbol{y}_n - \boldsymbol{y}_0) - R(\boldsymbol{x}_n - \boldsymbol{x}_0)\right] p_n$$

$$\begin{aligned}\Phi(\boldsymbol{x}_0, R) &= \sum (\boldsymbol{y}_n - \boldsymbol{y}_0)^\top (\boldsymbol{y}_n - \boldsymbol{y}_0)\, p_n \\ &\quad + \sum (\boldsymbol{x}_n - \boldsymbol{x}_0)^\top (\boldsymbol{x}_n - \boldsymbol{x}_0)\, p_n \\ &\quad - 2 \sum (\boldsymbol{y}_n - \boldsymbol{y}_0)^\top R (\boldsymbol{x}_n - \boldsymbol{x}_0)\, p_n\end{aligned}$$

Solve $R^*, x_0^* = \operatorname{argmin} \Phi(\boldsymbol{x}_0, R)$ **by**

- Computing the first derivatives
- Setting derivatives to zero
- Solving the resulting equations

- with respect to $\boldsymbol{x}_0$

$$\begin{aligned}\frac{\partial \Phi(\boldsymbol{x}_0, R)}{\partial \boldsymbol{x}_0} &= -2 \sum (\boldsymbol{x}_n - \boldsymbol{x}_0)\, p_n \\ &\quad + 2 \sum R^\top (\boldsymbol{y}_n - \boldsymbol{y}_0)\, p_n\end{aligned}$$

- This simplifies to

$$\sum (\boldsymbol{x}_n - \boldsymbol{x}_0)\, p_n = R^\top \sum (\boldsymbol{y}_n - \boldsymbol{y}_0)\, p_n$$

Know Data Association

- This simplifies to

$$\sum (\boldsymbol{x}_n - \boldsymbol{x}_0)\, p_n \;=\; R^\top \sum (\boldsymbol{y}_n - \boldsymbol{y}_0)\, p_n \qquad \boldsymbol{y}_0 \;=\; \frac{\sum \boldsymbol{y}_n\, p_n}{\sum p_n}$$

- As $\sum (\boldsymbol{x}_n - \boldsymbol{x}_0)\, p_n \;=\; 0$
- We obtain $\sum \boldsymbol{x}_n\, p_n - \sum \boldsymbol{x}_0\, p_n \;=\; 0$
- This leads to

$$\boxed{\boldsymbol{x}_0 \;=\; \frac{\sum \boldsymbol{x}_n\, p_n}{\sum p_n}}$$

- The optimal value for $\boldsymbol{x}_0$ is the **weighted mean** of the points $\boldsymbol{x}_n$

Know Data Association

$$\Phi(\boldsymbol{x}_0, R) = \sum (\boldsymbol{y}_n - \boldsymbol{y}_0)^\top (\boldsymbol{y}_n - \boldsymbol{y}_0)\, p_n$$
$$+ \sum (\boldsymbol{x}_n - \boldsymbol{x}_0)^\top (\boldsymbol{x}_n - \boldsymbol{x}_0)\, p_n$$
$$\boxed{-2 \sum (\boldsymbol{y}_n - \boldsymbol{y}_0)^\top R (\boldsymbol{x}_n - \boldsymbol{x}_0)\, p_n}$$

- So we need to find R that maximizes

$$R^* = \operatorname*{argmax}_R \sum (\boldsymbol{y}_n - \boldsymbol{y}_0)^\top R (\boldsymbol{x}_n - \boldsymbol{x}_0)\, p_n$$

- Given we know $\boldsymbol{x}_0$, compute mean-reduced coordinates as

$$\boldsymbol{a}_n = (\boldsymbol{x}_n - \boldsymbol{x}_0)$$
$$\boldsymbol{b}_n = (\boldsymbol{y}_n - \boldsymbol{y}_0)$$

- This leads to the compact form

$$R^* = \operatorname*{argmax}_R \sum \boldsymbol{b}_n^\top R \boldsymbol{a}_n\, p_n$$

Rewrite Using the Trace

- We can directly rewrite

$$R^* = \operatorname*{argmax}_R \sum \boldsymbol{b}_n^\top R \boldsymbol{a}_n\, p_n$$

- using the trace as

$$R^* = \operatorname*{argmax}_R \operatorname{tr}(R^\top H)$$

- with the cross covariance matrix

$$H = \sum (\boldsymbol{a}_n \boldsymbol{b}_n^\top)\, p_n$$

- **Thus, find R that maximizes** $\operatorname{tr}(R^\top H)$

Know Data Association

- SVD gives us

$$\operatorname{svd}(H) = UDV^\top$$

- with

$$U^\top U = I \qquad V^\top V = I \qquad D = \operatorname{diag}(d_i)$$

- Let's see what happens if we set

$$R = VU^\top$$

- Then, we obtain

$$\operatorname{tr}(R^\top H) = \operatorname{tr}(\underbrace{VU^\top}_{R}\underbrace{UDV^\top}_{H}) = \operatorname{tr}(V\underbrace{U^\top U}_{I}DV^\top) = \operatorname{tr}(VDV^\top)$$

- and we can rewrite this as

$$\operatorname{tr}(VDV^\top) = \operatorname{tr}(VD^{\frac{1}{2}}D^{\frac{1}{2}}V^\top)$$

$$VD^{\frac{1}{2}} = A$$

$$tr(R^T H) = tr(AA^T)$$

- For every pos. definite matrix A holds

$$\operatorname{tr}\left(AA^\top\right) \geq \operatorname{tr}\left(R'AA^\top\right)$$

for any rotation matrix R'

- Result of the Schwarz inequality
- This means

$$\operatorname{tr}(RH) = \operatorname{tr}\left(AA^\top\right) \geq \operatorname{tr}\left(R'AA^\top\right) = \operatorname{tr}(\underline{R'}RH)$$

any other rotation matrix

- Thus, our choice $R = VU^\top$ was optimal as it maximizes the trace

Know Data Association

- Starting from

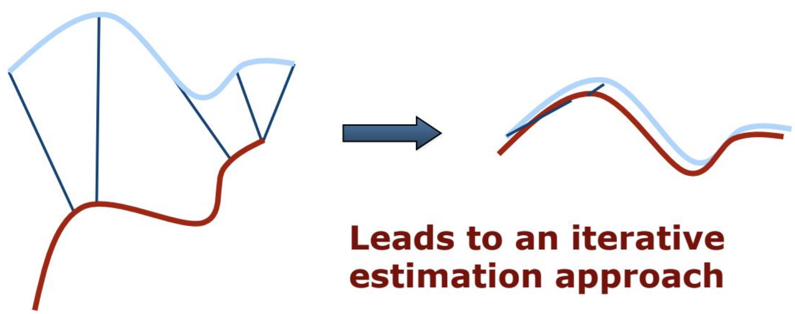
$$\boldsymbol{x}_0 = R^\top \boldsymbol{y}_0 - R^\top \boldsymbol{t}$$

- directly leads to

$$\boxed{\boldsymbol{t} = \boldsymbol{y}_o - R\boldsymbol{x}_0}$$

- Rotation $R = VU^\top$
- Translation $t = y_0 - Rx_0$

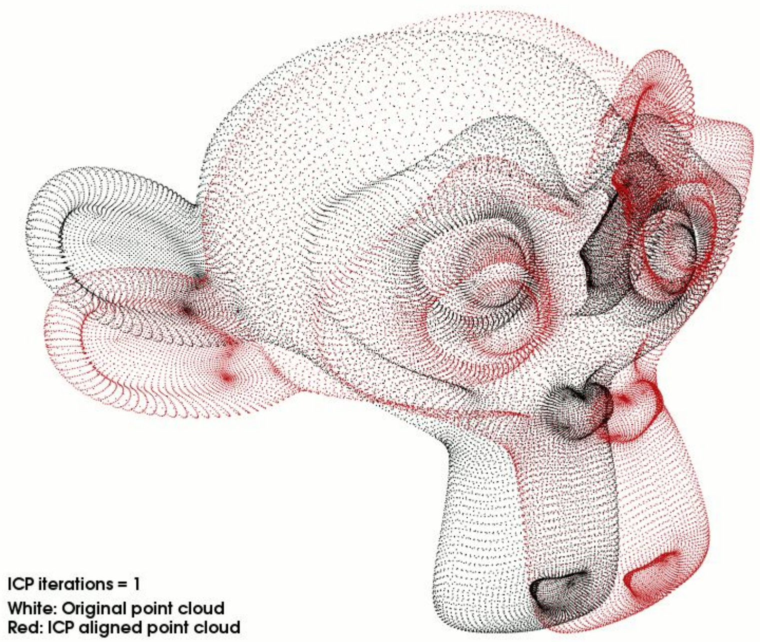
Unknown Data Association



- ❖ **No direct optimal solution exists.**
- ❖ **Reliable correspondence estimates:**
  - ➢ **Utilize Rotation Matrix and Translation Vector.**
  - ➢ **Leads to an optimal solution in Point Cloud Registration.**
- ❖ **Definition:** Align points based on estimated correspondence, especially when data association is unclear.
- ❖ **Purpose:** Align two Point Clouds when Data Association is unknown.
- ❖ **Method:**
  - ➢ Estimate correspondence using the closest point.
  - ➢ Iteratively refine alignment to minimize errors.
- ❖ **Goal:**
  - ➢ Efficiently align Point Clouds and reduce errors.

Unknown Data Association - Valina ICP

- **Definition:** Align points based on estimated correspondence, especially when data association is unclear.
- **Purpose:** Align two Point Clouds when Data Association is unknown.
- **Method:**
  - Estimate correspondence using the closest point.
  - Iteratively refine alignment to minimize errors.
- **Goal:**
  - Efficiently align Point Clouds and reduce errors.

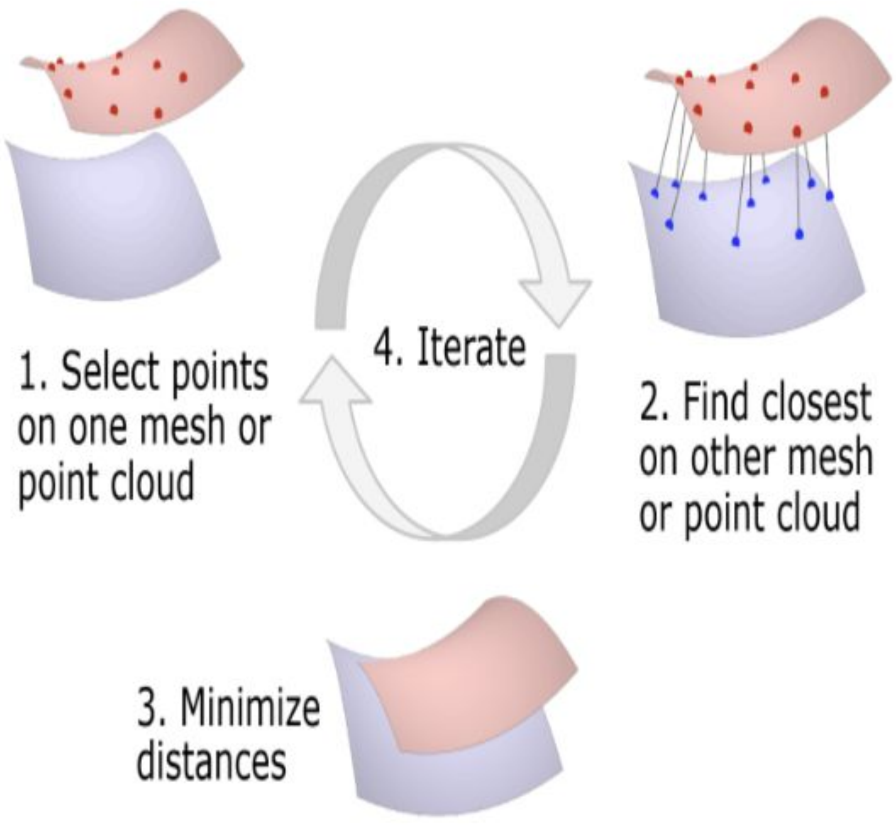


https://github.com/yassram/iterative-closest-point

Valina ICP

Point Cloud Alignment Process

1. **Initial Correspondence Setup**
   - For every point in point cloud x_n, locate the nearest point in y_n.
   - Establish a correspondence between the two points.
2. **Calculate Rotation and Translation**
   - Use the established correspondence to deduce the rotation matrix R and the translation vector t.
3. **Align Point Clouds**
   - Adjust x_n using the formula:
     $x_n = R * x_n + t$
   - This transformation aligns x_n with y_n.
4. **Compute Error and Iterate**
   - Define the disparity between x_n and y_n as the error.
   - If the error exceeds the acceptable threshold, revert to step 1 and iterate the process.



Valina ICP

Drawbacks of Valina ICP

1. High Iteration
2. Issue with Correspondences:

- Incorrect initial correspondences.
- Utilizing poor correspondences can result in highly inaccurate results.

Research aiming to address the limitations

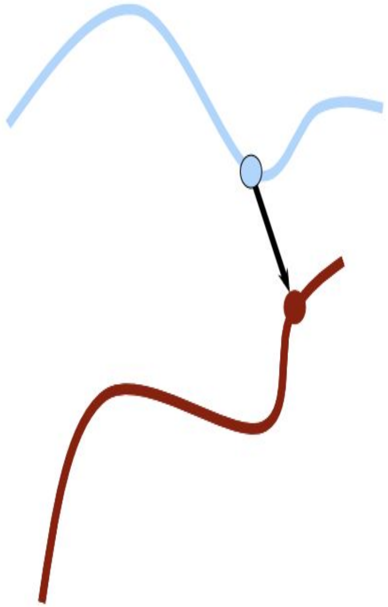
1. Consider point subsets.
2. Different data association strategies.
3. Weight the correspondences.
4. Reject potential outlier point pairs.

Data Association

- Has huge impact on convergence and speed
- Various different matching methods:
  - Closest point
  - Closest compatible point
  - Normal shooting
  - Point-to-plane
  - Projection-based approaches

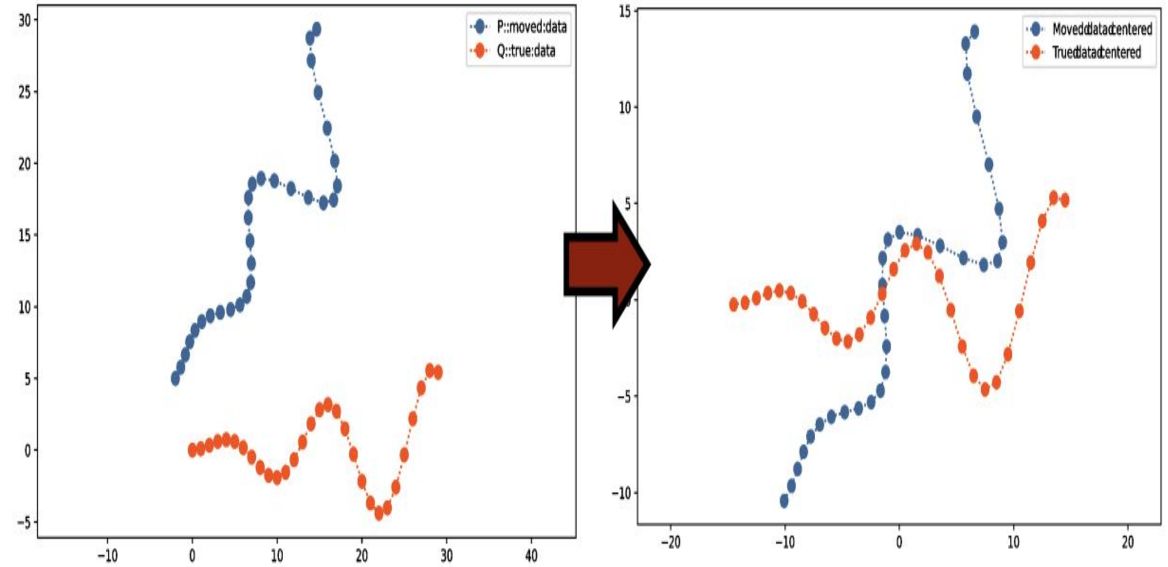
Closest Point

Find closest point in other the point set (using kd-trees)



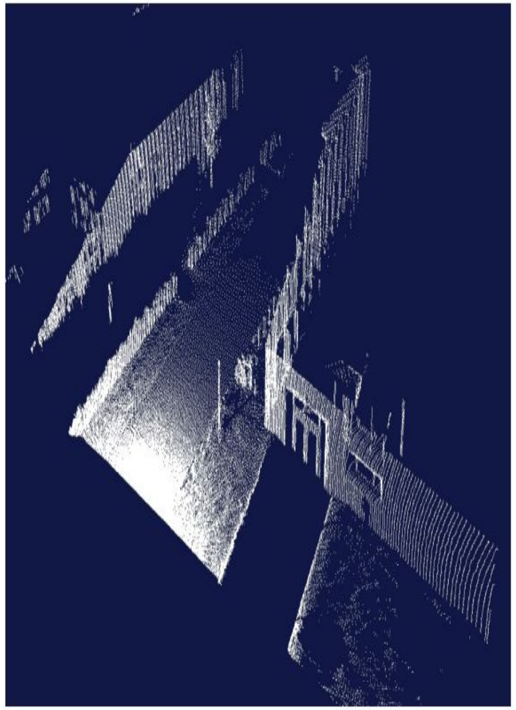
Generally stable, but slow convergence.
Often the first approach to try ("Vanilla ICP")

Without an initial guess, align the center of masses of both point sets before searching correspondences

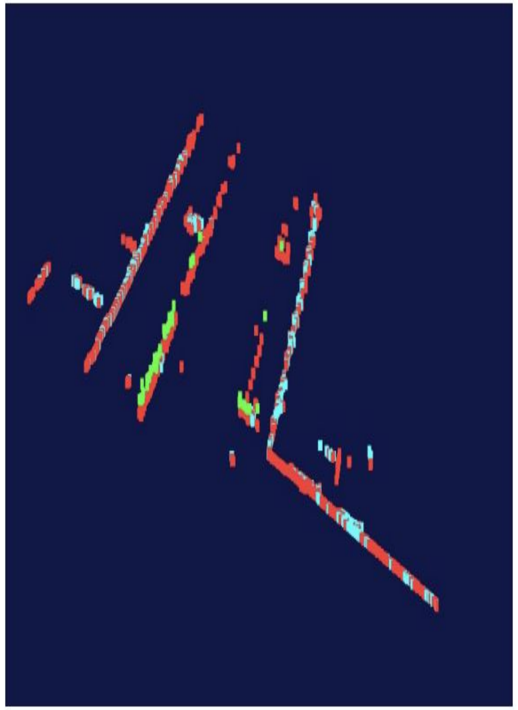


Closest compatible point

- 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



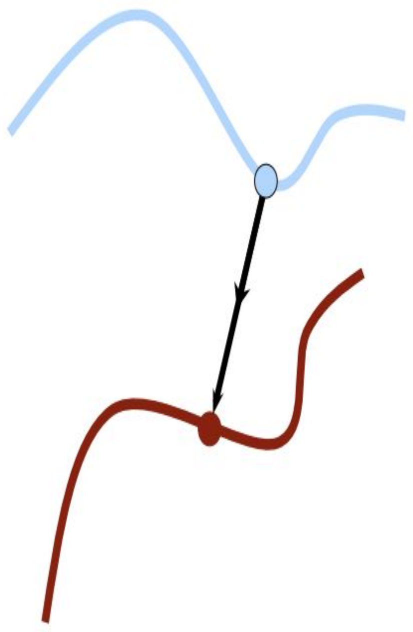
Full 3D scan (~200.000 points)



Extracted features (~5.000 points)

Normal shooting, Projection-based approaches

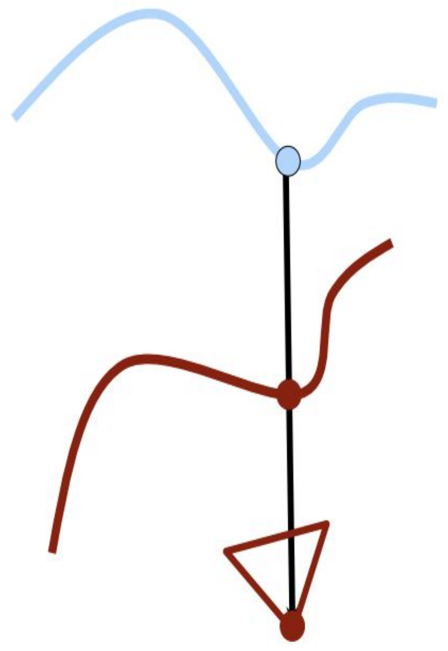
Project along normal, intersect other point set to find a correspondence



Slightly better convergence results than closest point for smooth structures, but worse for noisy or complex structures

Projective Data Association

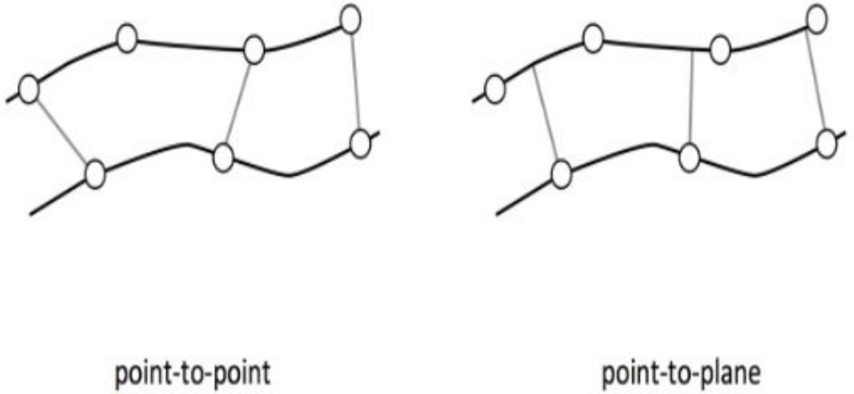
Searches for correspondences by projecting a point towards the sensor viewpoint



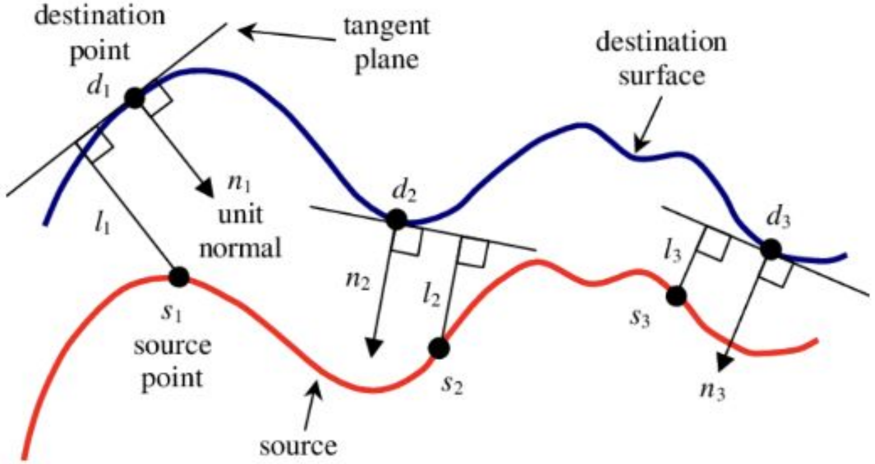
Point-to-Plane ICP

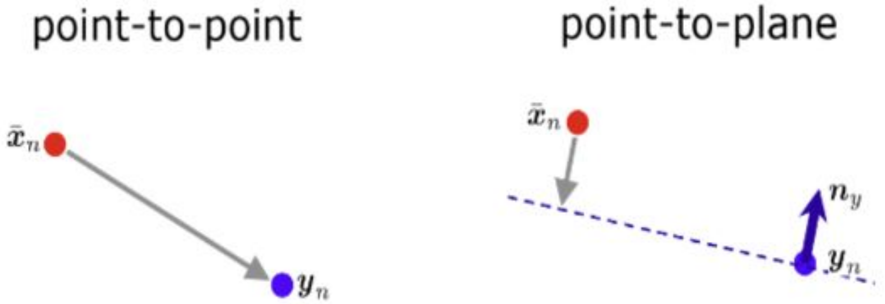
Instead of directly connecting source points to target points

- Create a virtual plane (or line) between points on the target.
- Compute the normal vector.
- Choose the closest point based on this vector.
- Calculate the Euclidean distance to establish data association



- Error = project point-to-point onto the direction of the normal, shot from the found point





$$\min \sum ||\boldsymbol{y}_n - \bar{\boldsymbol{x}}_n||^2 \qquad \min \sum ((\boldsymbol{y}_n - \bar{\boldsymbol{x}}_n) \cdot \boldsymbol{n}_y)^2$$

1. **Basic Concept**:
    - GICP replaces the cost function of the original ICP with a probabilistic model.
    - The method to find correspondences using nearest neighbor search remains consistent with the traditional ICP.
2. **Point Correspondences**:
    - Considering point correspondences between point clouds A and B, hypothetical sets $A\hat{}$ and $B\hat{}$ exist.
    - These points are assumed to be drawn from a normal distribution with specific covariance matrices.
3. **Transformation**:
    - The correct transformation T^* establishes a perfect correspondence.
    - The difference $d(T)$ for any transformation follows a specific distribution.
4. **Maximum Likelihood Estimation (MLE)**:
    - MLE is employed to iteratively determine the transformation T.
    - The original ICP can be viewed as a special case of GICP.
5. **Point-to-Plane ICP**:
    - This method aims to find a transformation that minimizes the difference projected onto a specific plane.
6. **Advantages of GICP**:
    - A significant benefit of GICP is the flexibility to choose any set of covariance matrices.
    - A direct application of GICP is the plane-to-plane ICP, which considers surface normal information from both point clouds.

Summary

1. Sub-sampling.
- Conduct sub-sampling to obtain a point cloud suitable for alignment.

2. Determine Correspondences.
- Choose the appropriate correspondences based on the situation.

3. Ensure Robust Performance.
- Assign weights or remove outlier candidates to enhance performance.

4. Utilize SVD.
- Use the SVD algorithm to compute the rotation matrix R and the translation vector t.

5. Apply Rotation and Translation.
- Apply the rotation matrix $\mathrm{R}$and translation vector t to all points.

6. Calculate Error.
- Compute the error value.

7. Iterative Process.
- Repeat the process until the error is below a certain threshold.

8. Final Alignment.
- Complete the final alignment of the point cloud.

Reference

- https://www.youtube.com/watch?v=dhzLQfDBx2Q
- https://www.youtube.com/watch?v=2hC9IG6MFD0

Thank You

🗂️[3D Sensor Data Processing Curriculum]