NA 568 - Winter 2022

# Point Cloud Registration I

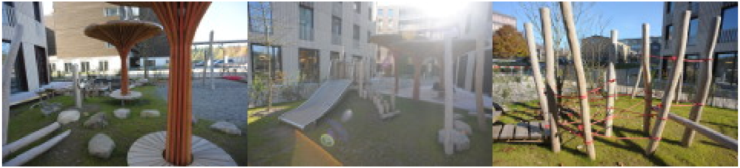
**Maani Ghaffari**

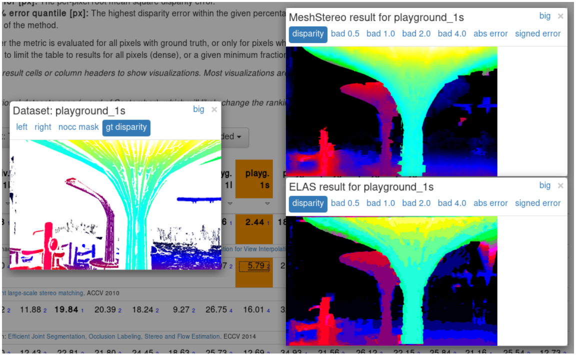
March 17, 2022 🍀

UNIVERSITY OF MICHIGAN

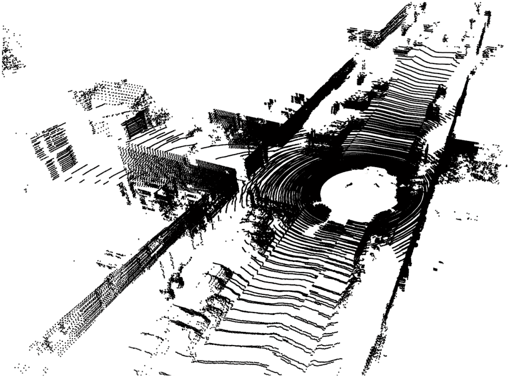
TUM RGBD SLAM Data Set:
https://vision.in.tum.de/data/datasets/rgbd-dataset

# Depth Cameras (RGB-D and Stereo) Sensors





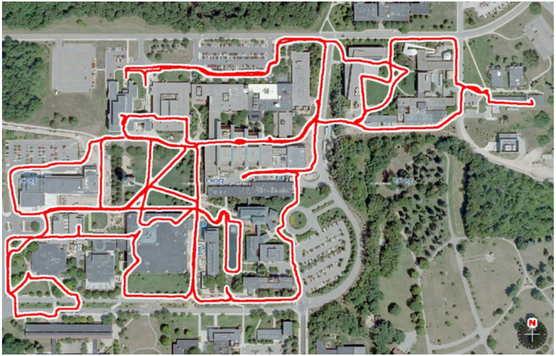

ETH3D SLAM & Stereo Benchmarks: `https://github.com/ETH3D`

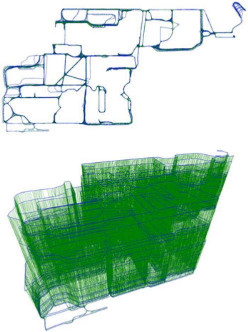


The KITTI Vision Benchmark Suite:
http://www.cvlibs.net/datasets/kitti/eval_odometry.php

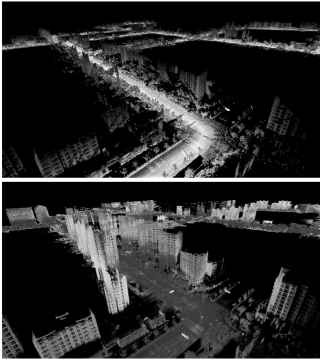
The University of Michigan North Campus Long-Term Vision and LIDAR Data Set:

http://robots.engin.umich.edu/nclt/index.html#top





http://irap.kaist.ac.kr/dataset



Find the geometric relationship between sensor data and some prior (such as previous sensor data or a map).

Approaches can vary by:

- Resolution
  - Coarse algorithms look for crude alignments, such as for place recognition.
  - Fine algorithms try to achieve better alignments, but generally need more accurate initial guesses.
- Data
  - Direct algorithms use the raw observations.
  - Indirect (feature-based) algorithms abstract dense raw data into sparse features.
- Local vs. global solvers.
- Real-time vs. offline solvers.

## Point Cloud Registration Problem

- A set of 3D points: $\mathcal{X} := \{x_i\}$, $\mathcal{X} \subset \mathbb{R}^3$

- Action of a rigid body transformation on 3D points: $T \cdot x := Rx + p$

- $R \in \mathrm{SO}(3)$, $p \in \mathbb{R}^3$

**Definition (Target Point Cloud)**

The point cloud $\mathcal{X}_t$ which is considered to be in a fixed reference frame is called the target point cloud, i.e., the point cloud we are trying to align to.

**Definition (Source Point Cloud)**

The point cloud $\mathcal{X}_s$ that we transform by $T \in \mathrm{SE}(3)$ to align to the target point cloud.

## Iterative Closest Point (ICP) Algorithm, Besl and McKay

- **Step 1:** Determine associations, $\mathcal{I}$, between target and source by finding the nearest neighbor.
- **Step 2:** Given those associations, minimize the residual between points.

- **Step 1:**

$$i_k = \arg\min_k \|x_k^t - T \cdot x_i^s\|$$

$$\mathcal{I} := \{i_k\}$$

- **Step 2:**

$$r_k(T) := x_k^t - T \cdot x_k^s$$

$$T^{\mathsf{OPT}} = \arg\min_{T \in \mathrm{SE}(3)} \sum_{k \in \mathcal{I}} \|r_k(T)\|^2$$

We can solve this problem using the Gauss-Newton algorithm. To compute the Jacobian, we perturb the residual using $T \leftarrow \exp(\xi^\wedge)T$ and apply a first order approximation of the exponential map $\exp(\xi^\wedge) \approx I + \xi^\wedge$.

$$\begin{aligned} r_k\left(\exp(\xi^\wedge)T\right) &= (x_k^t - \exp(\xi^\wedge)T \cdot x_k^s) \\ &\approx (x_k^t - (I + \xi^\wedge)T \cdot x_k^s) \\ &= (x_k^t - T \cdot x_k^s) - (\xi^\wedge T \cdot x_k^s) \\ &= r_k(T) - (\xi^\wedge \cdot z_k), \end{aligned}$$

where we define $z_k := T \cdot x_k^s$ to be the source point after applying the transformation $T$.

$$r_k\left(\exp(\xi^\wedge)T\right) \approx r_k(T) - (\xi^\wedge \cdot z_k).$$

To find the Jacobian we need to solve $-(\xi^\wedge \cdot z_k) = J_k \xi$.

$$\begin{aligned}(-\xi^\wedge \cdot z_k) &= (-\phi^\wedge z_k - \rho) \\ &= (z_k^\wedge \phi - \rho) \\ &= \begin{bmatrix} z_k^\wedge & -I \end{bmatrix} \xi.\end{aligned}$$

We learn that $J_k = \begin{bmatrix} z_k^\wedge & -I \end{bmatrix}$ (a $3 \times 6$ matrix).

- FLANN is a library for performing fast approximate nearest neighbor searches in high dimensional spaces:

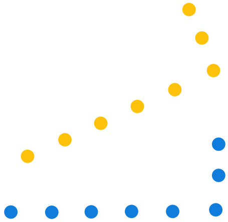
  `https://www.cs.ubc.ca/research/flann/`

- nanoflann: a C++11 header-only library for Nearest Neighbor (NN) search with KD-trees

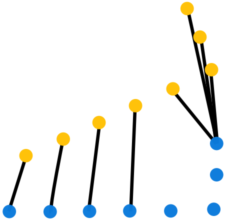
  `https://github.com/jlblancoc/nanoflann`

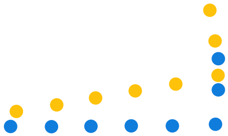
- How to use a k-d tree to search using the Point Cloud Library (PCL):

  `https://pcl.readthedocs.io/projects/tutorials/en/latest/kdtree_search.html`

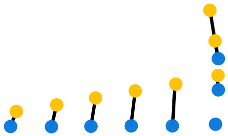


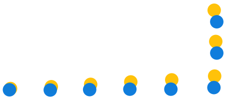
# ICP





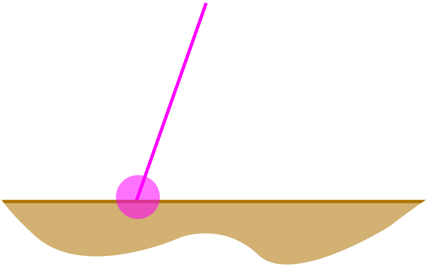
# ICP



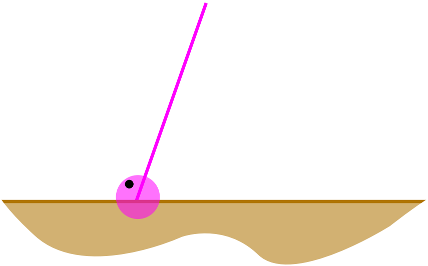


# Generalized ICP (GICP), Segal et al.

## Generalized ICP (GICP), Segal et al.

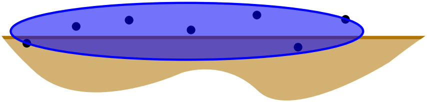


# Generalized ICP (GICP), Segal et al.

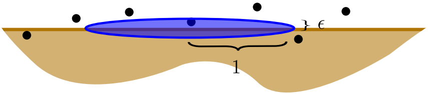


# Generalized ICP (GICP), Segal et al.

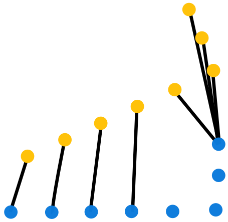
$$\Sigma_{\mathbf{x}} = \sum_i^N \frac{(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top}{N-1}$$



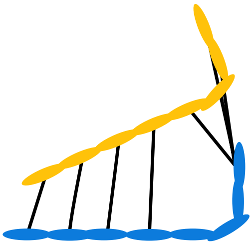
$$\Sigma_{\mathbf{x}} = Q \Lambda Q^{\top}$$
$$\lambda_1 = 1$$
$$\lambda_2 = \epsilon$$



# GICP



# GICP



**Remark**

*We use the following notation for the weighted norm of a vector:* $\|r\|_{\Sigma}^2 = r^{\mathsf{T}} \Sigma^{-1} r$.

- The residual negative log-likelihood:

$$-\log p(r_k | x_k^t, x_k^s, i_k; T) = \|x_k^t - T \cdot x_k^s\|^2_{\Sigma_k^t + R\Sigma_k^s R^\mathsf{T}} + \text{const.}$$

- Define $C_k := \Sigma_k^t + R\Sigma_k^s R^\mathsf{T}$

- GICP cost function:

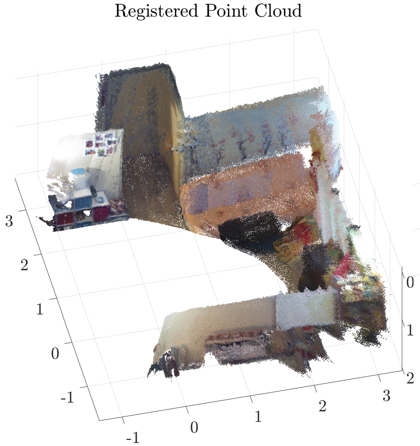
$$\begin{aligned} f_{\texttt{GICP}}(T; \mathcal{R} | \mathcal{X}_t, \mathcal{X}_s, \mathcal{I}) &:= \sum_k^n \|x_k^t - T \cdot x_k^s\|^2_{C_k} \\ &\propto -\log p(\mathcal{R} | \mathcal{X}_t, \mathcal{X}_s, \mathcal{I}; T) \end{aligned}$$

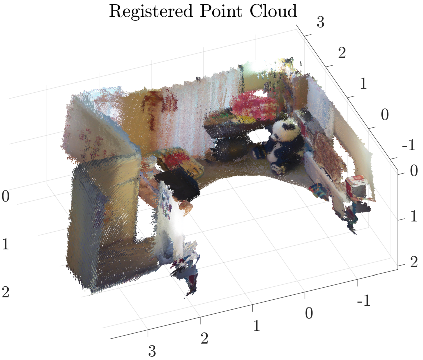
- Solve for the parameter $T \in \mathrm{SE}(3)$:

$$T^{\mathsf{OPT}} = \underset{T \in \mathrm{SE}(3)}{\arg\min}\, f_{\texttt{GICP}}(\mathcal{R} | \mathcal{X}_t, \mathcal{X}_s, \mathcal{I})$$

## Registered scene solved using the GICP on $\mathrm{SE}(3)$

https://www.mathworks.com/help/vision/examples/3-d-point-cloud-registration-and-stitching.html



- Iterative Closet Point using point-to-point distance (see previous slides and Besl and McKay (1992)).
- Iterative Closet Point using point-to-plane distance (see Y. Chen and G. Medioni (1991)).
- Generalized Iterative Closet Point; probabilistic formulation and maximum likelihood estimation (see A. Segal, D. Haehnel, and S. Thrun (2009)).
- Normal Distributions Transform (NDT) and its extensions.
- Extensions are available by introducing extra measurements or avoiding hard data associations using, for example, Expectation Maximization algorithm. For instance, see "Point Clouds Registration with Probabilistic Data Association".

## Challenges in Point Cloud Registration

- Unknown correspondences between two point clouds.
- Finding local minima and the need for a “good” initial guess.
- Partial and unknown overlap between two point clouds.
- Variable density of the overlapping area, i.e., sparse to dense registration.
- Sensors with limited field of view or highly noisy sensory data make the registration problem harder to solve.

# Small Scale Problems or Offline Processing

If the size of the problem is not too large and we have unlimited time for processing data:

- We can exhaustively use multiple initializations to find the best solution.
- We can simultaneously solve for a permutation matrix to find the best correspondences and the optimal rigid body transformation.
- An example is Go-ICP that uses Branch-and-Bound (BnB) method for global optimization.
- Probably you can come up with more ideas ...

# GICP-SE(3) Algorithm

https://github.com/UMich-CURLY-teaching/UMich-ROB-530-public/blob/main/code-examples/MATLAB/gicp/gicp_SE3.m

**Algorithm 1** GICP-SE(3)

**Require:** Initial transformation $T^{\text{init}}$, target cloud $\mathcal{X}_t$, source cloud $\mathcal{X}_s$;

$T^{\text{OPT}} \leftarrow T^{\text{init}}$

**while** not converged **do**

  $T^{\text{old}} \leftarrow T^{\text{OPT}}$

  $\mathcal{I} \leftarrow \texttt{nnsearch}(\mathcal{X}_s, \mathcal{X}_t, T^{\text{old}})$ ▷ Find Association

  $T^{\text{OPT}} \leftarrow \arg\min_{T \in \mathrm{SE}(3)} f_{\texttt{GICP}}(T; \mathcal{R} | \mathcal{X}_t, \mathcal{X}_s, \mathcal{I})$ ▷ Optimize over SE(3)

  // Check convergence using distance threshold $\epsilon$

  **if** $\mathrm{d}_{\mathrm{SE}(3)}(T^{\text{old}}, T^{\text{OPT}}) < \epsilon$ **then**

    converged $\leftarrow$ true

**return** $T^{\text{OPT}}$

- The distance between two rigid body transformation is define as:

  $$\mathrm{d}_{\mathrm{SE}(3)}(T_1, T_2) := \|\log(T_1 T_2^{-1})^{\vee}\|$$

  where $\log(\cdot)$ is the Lie logarithm map.

- We define $\mathrm{d}_{\mathrm{SO}(3)}(T_1,T_2) := \|\log(R_1 R_2^{\mathsf{T}})^{\vee}\|$ and $\mathrm{d}_{\mathbb{R}^3}(T_1,T_2) := \|t_1 - R_1 R_2^{\mathsf{T}} t_2\|$ which are consistent with the $\mathrm{d}_{\mathrm{SE}(3)}(\cdot,\cdot)$ definition.

## Robust Estimator

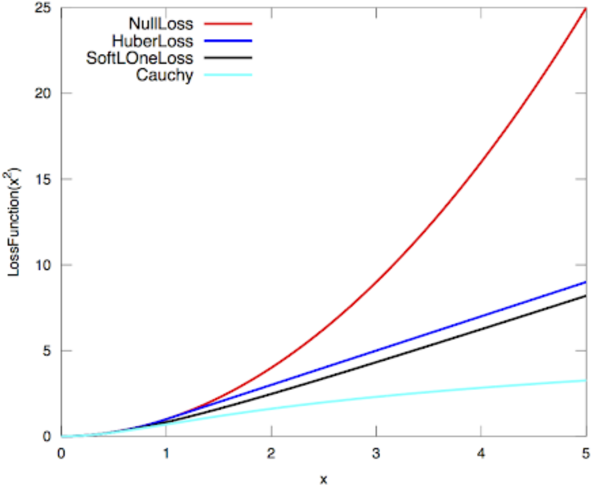
- In practice, there are outliers in the data association solution using the nearest neighbor search.

- To ensure that any point in the source cloud that does not have a counter part will not affect the solution, we introduce a robust estimator using the Cauchy loss function:

$$\rho_\alpha(x) = \alpha^2 \ln(1 + \frac{x}{\alpha^2})$$

- $\alpha$ is a parameter that controls where the loss begins to scale sublinearly.

Examples of different loss functions:




Courtesy: Ceres Solver

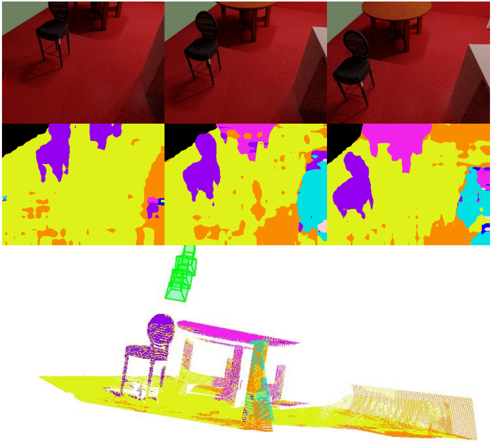
- Then the new cost function becomes

$$f_{\texttt{GICP}}(T; \mathcal{R} | \mathcal{X}_t, \mathcal{X}_s, \mathcal{I}) = \sum_{k}^{n} \rho_\alpha(\|x_k^t - T \cdot x_k^s\|_{C_k}^2)$$

- The effect of the Cauchy loss function on the gradient can be derived using the chain rule.

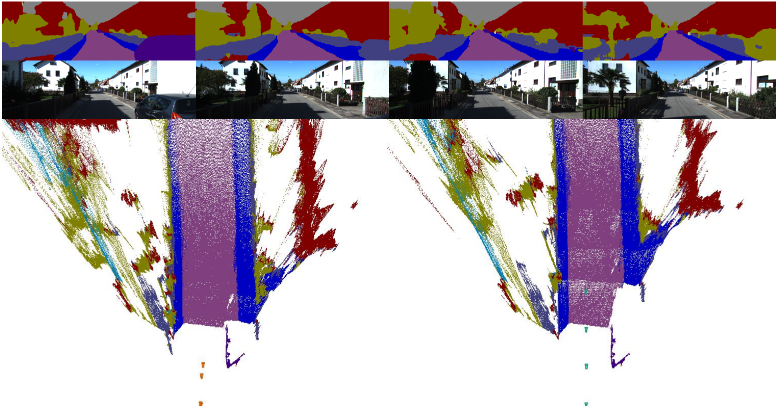
# An Extension: Semantic ICP

Joint inference using semantic and geometric measurements, e.g., semantically labeled pint clouds; EM-style inference approach: `https://bitbucket.org/saparkison/semantic-icp`



# Example: KITTI Visual Odometry Data Set

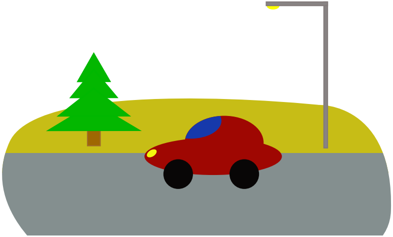
Sequential point clouds aligned using Semantic ICP on the right and GICP on the left. The repeated object on the right side of the roadway are artifacts of poor alignment by GICP.

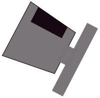


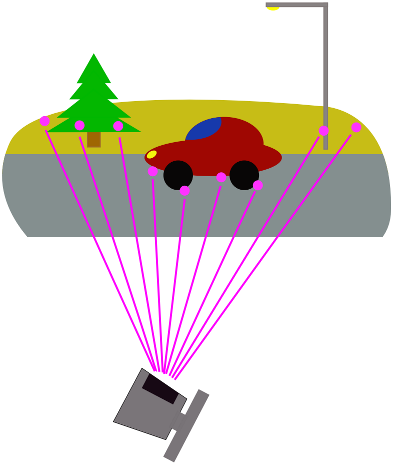
a) GICP

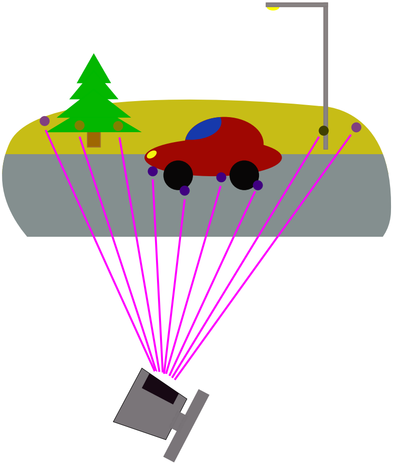
b) Semantic ICP

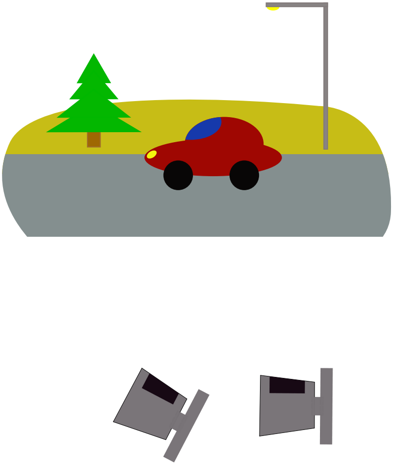
Building Vegetation Car Road Fence Sidewalk Pole

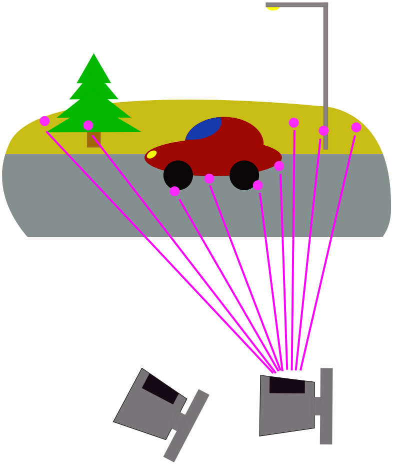


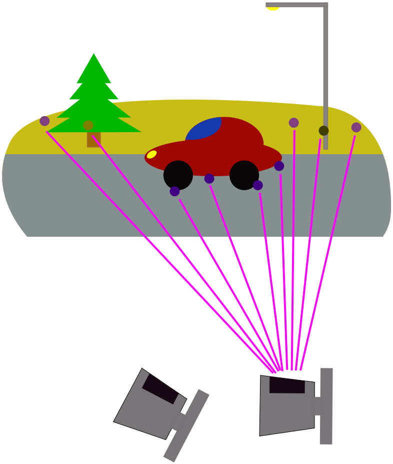


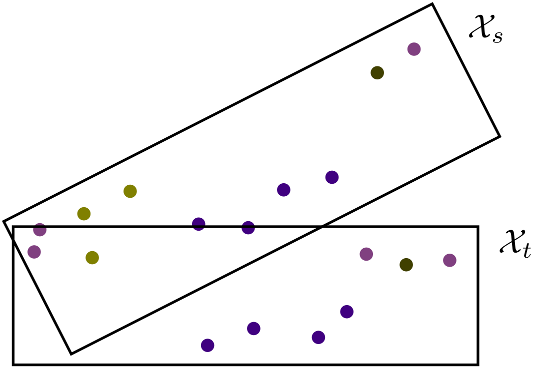


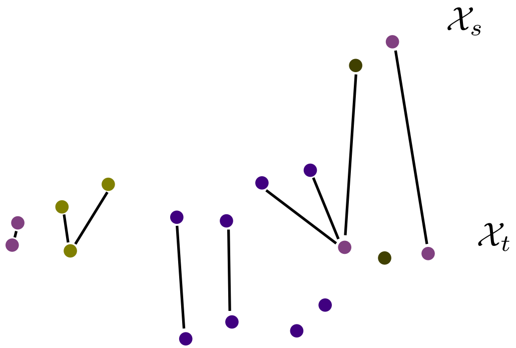


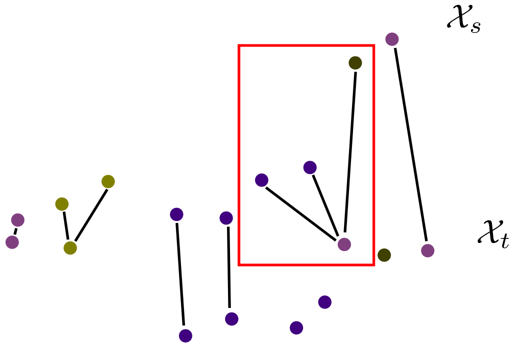


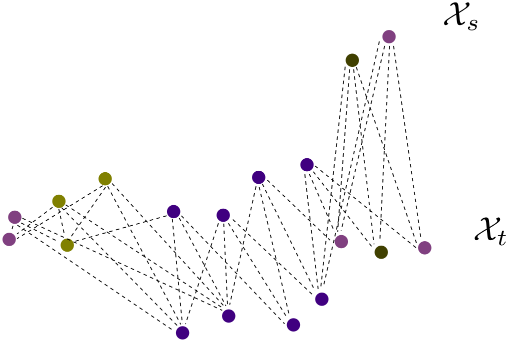


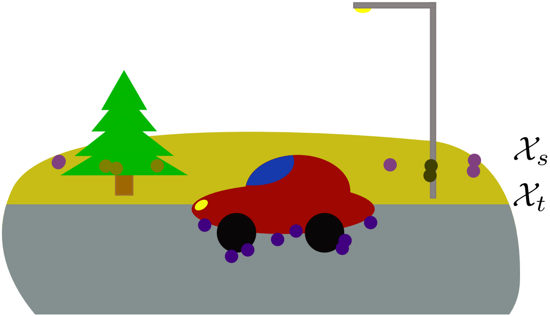



$\mathcal{X}_s$
$\mathcal{X}_t$


$\mathcal{X}_s$
$\mathcal{X}_t$


$\mathcal{X}_s$
$\mathcal{X}_t$


$\mathcal{X}_s$
$\mathcal{X}_t$


$\mathcal{X}_s$
$\mathcal{X}_t$

**Remark**

*Associations $\mathcal{I}$ are not observed but are required to minimize the registration problem.*

The standard ICP approach can be viewed as a version of “hard” Expectation-Maximization (EM):

- **Expectation:** Find the most likely associations using a nearest neighbor algorithm, i.e., soft associations as weights.
- **Maximization:** Find the transformation that optimizes the cost function.

We combine the weights into a weight array, $w = \mathrm{vec}(w_1, \ldots, w_{n\times N})$, that is $n \times N$ counting non-zero weights. Subsequently, the maximization step becomes:

$$\begin{aligned} T^{\mathsf{OPT}} &= \operatorname*{arg\,min}_{T\in \mathrm{SE}(3)} f_{\mathtt{SICP}}(T, w; \mathcal{R}|\mathcal{X}_t, \mathcal{X}_s, \mathcal{I}) \\ &= \operatorname*{arg\,min}_{T\in \mathrm{SE}(3)} \sum_{k=1}^{n\times N} \rho_\alpha(w_k \|x_k^t - T\cdot x_k^s\|_{C_k}^2) \end{aligned}$$

# Semantic ICP Algorithm

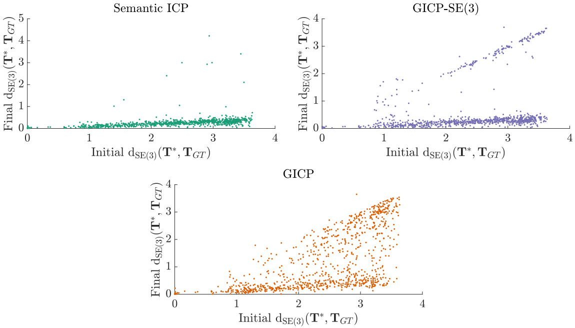
**Algorithm 2** Semantic ICP

**Require:** Initial transformation $T^{\text{init}}$, target cloud $\mathcal{X}_t$, source cloud $\mathcal{X}_s$, semantic labels;

$T^{\text{OPT}} \leftarrow T^{\text{init}}$
**while** not converged **do**
  $T^{\text{old}} \leftarrow T^{\text{OPT}}$
  $w \leftarrow$ Compute weights ▷ Expectation
  $T^{\text{OPT}} \leftarrow \arg\min_{T \in \mathrm{SE}(3)} f_{\texttt{SICP}}(T, w; \mathcal{R} | \mathcal{X}_t, \mathcal{X}_s, \mathcal{I})$ ▷ Maximization
  // Check convergence using distance threshold $\epsilon$
  **if** $\mathrm{d}_{\mathrm{SE}(3)}(T^{\text{old}}, T^{\text{OPT}}) < \epsilon$ **then**
    $\text{converged} \leftarrow \text{true}$
**return** $T^{\text{OPT}}$

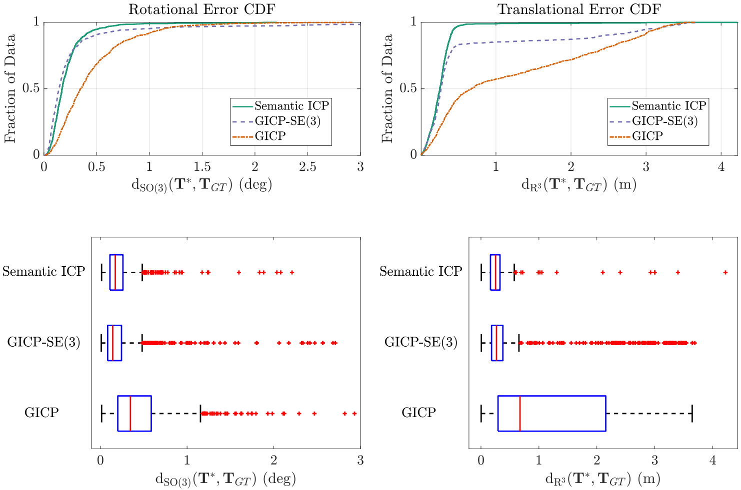
# Example: KITTI Visual Odometry Data Set

Scatter plots of the initial alignment vs. final alignment using $\mathrm{d}_{\mathrm{SE}(3)}(\cdot,\cdot)$ for each algorithm on the KITTI visual odometry dataset. We can see that GICP is less likely to converge as the initial offset gets larger, while Semantic ICP and GICP-$\mathrm{SE}(3)$ are more of a bimodal distribution, either staying near the initial transformation, or converging.



# Example: KITTI Visual Odometry Data Set

Error cumulative distribution function and box plots computed using KITTI sequence 05 data set.



- Point Cloud Library (PCL):
  `https://github.com/PointCloudLibrary/pcl`
- Open3D: `https://github.com/IntelVCL/Open3D`
- GICP-$\mathrm{SE}(3)$ & Semantic ICP:
  `https://bitbucket.org/saparkison/semantic-icp`

## Readings

- Segal, A., Haehnel, D. and Thrun, S., 2009. Generalized-ICP. RSS. http://www.roboticsproceedings.org/rss05/p21.pdf
- Parkison, S.A., Gan, L., Ghaffari Jadidi, M. and Eustice, R.M., 2018. Semantic Iterative Closest Point through Expectation-Maximization. BMVC. http://robots.engin.umich.edu/publications/sparkison-2018a.pdf