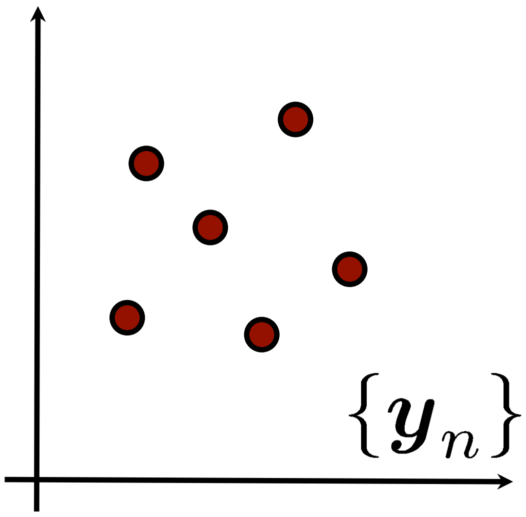
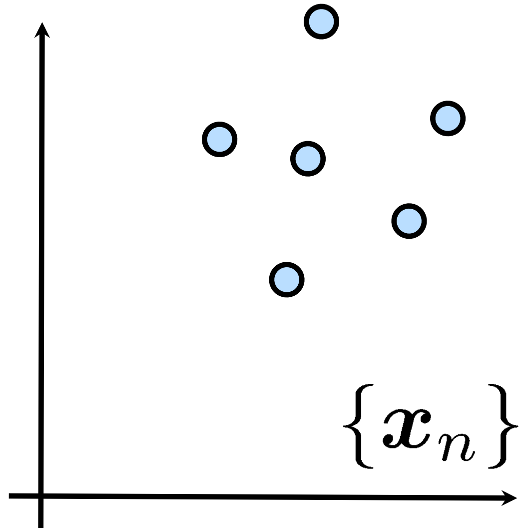
Photogrammetry & Robotics Lab

Point Cloud Registration & ICP #2: Unknown Data Association

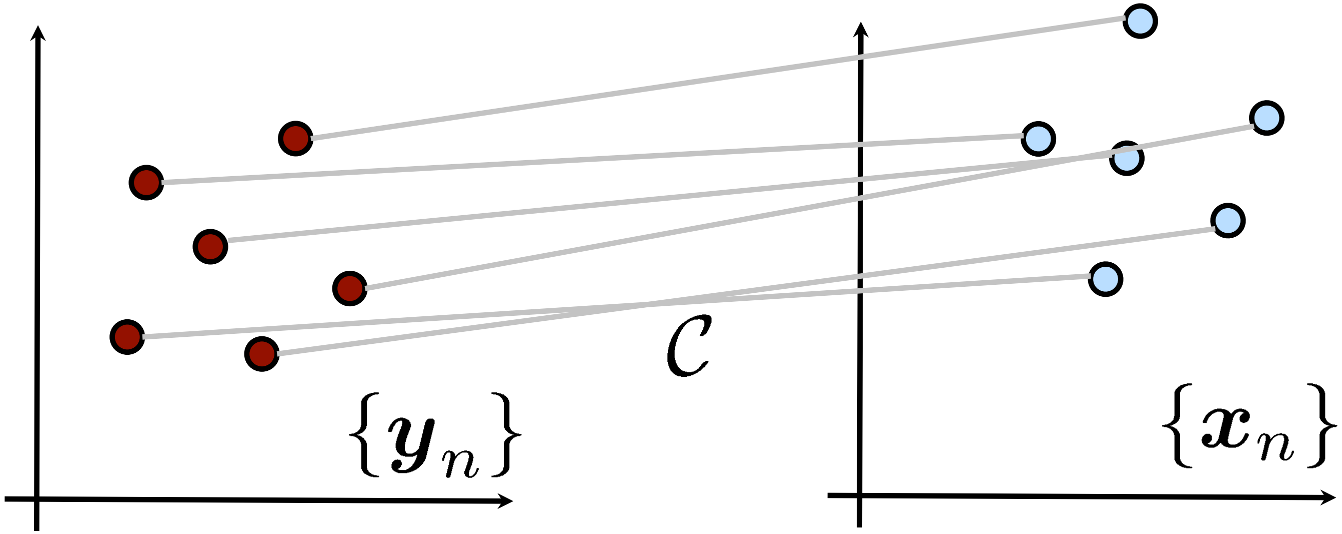
Cyrill Stachniss

Simple Form of Point Cloud Registration



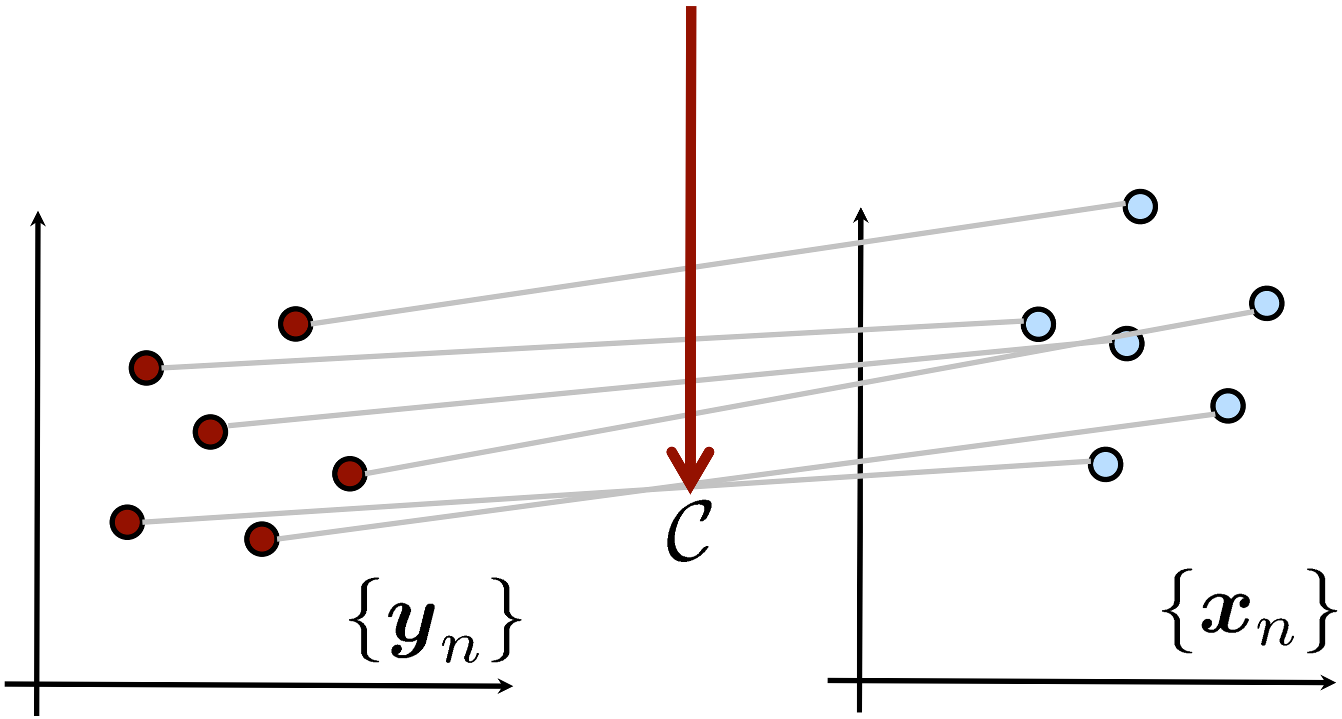


Simple Form of Point Cloud Registration



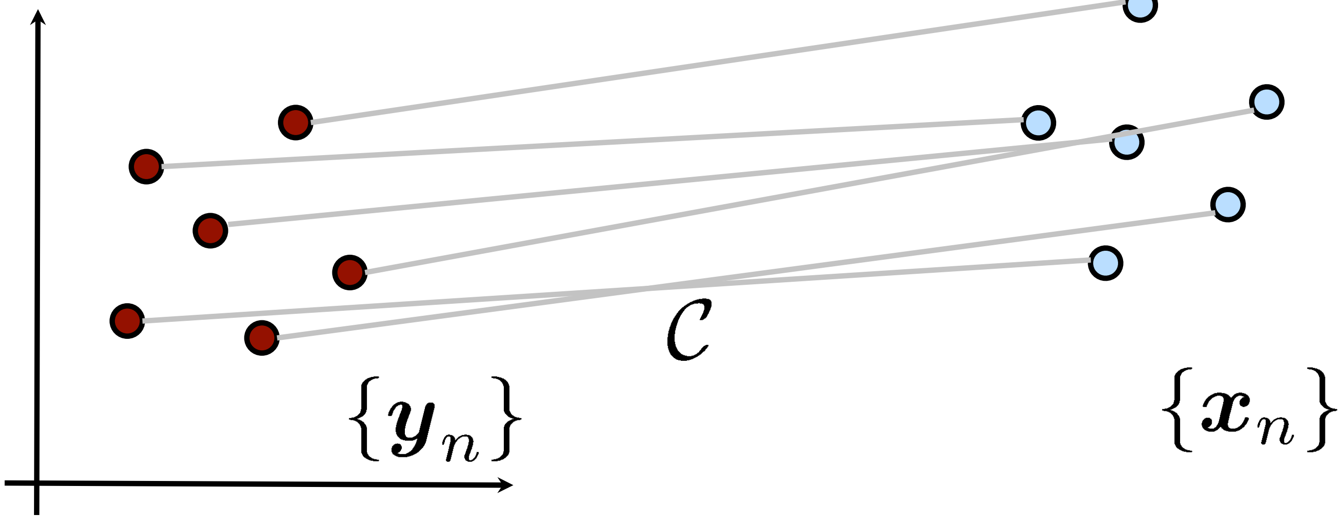
Simple Form of Point Cloud Registration

So far: assumed to be known!



Simple Form of Point Cloud Registration

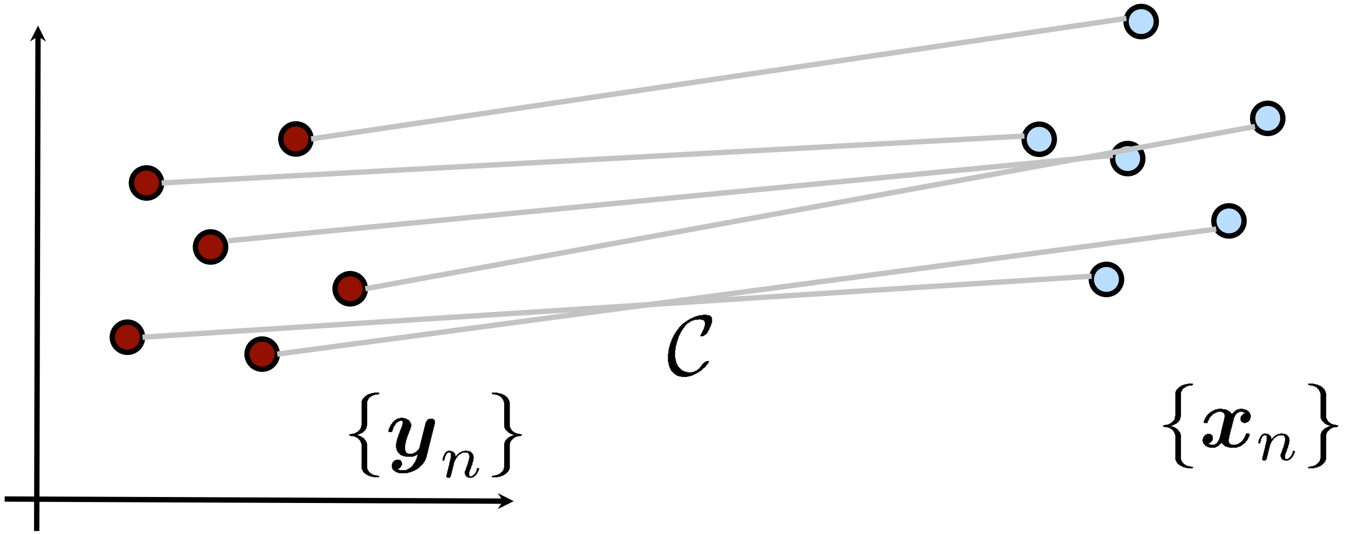
$$\bar{\boldsymbol{x}}_n = R\boldsymbol{x}_n + \boldsymbol{t}$$



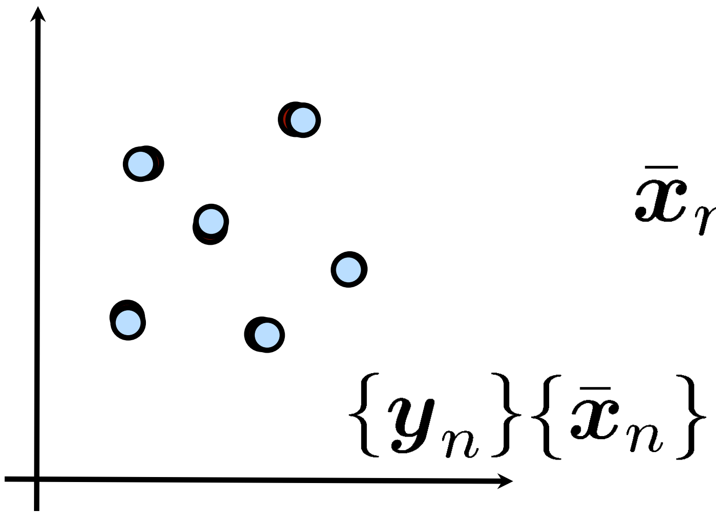
Simple Form of Point Cloud Registration

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

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



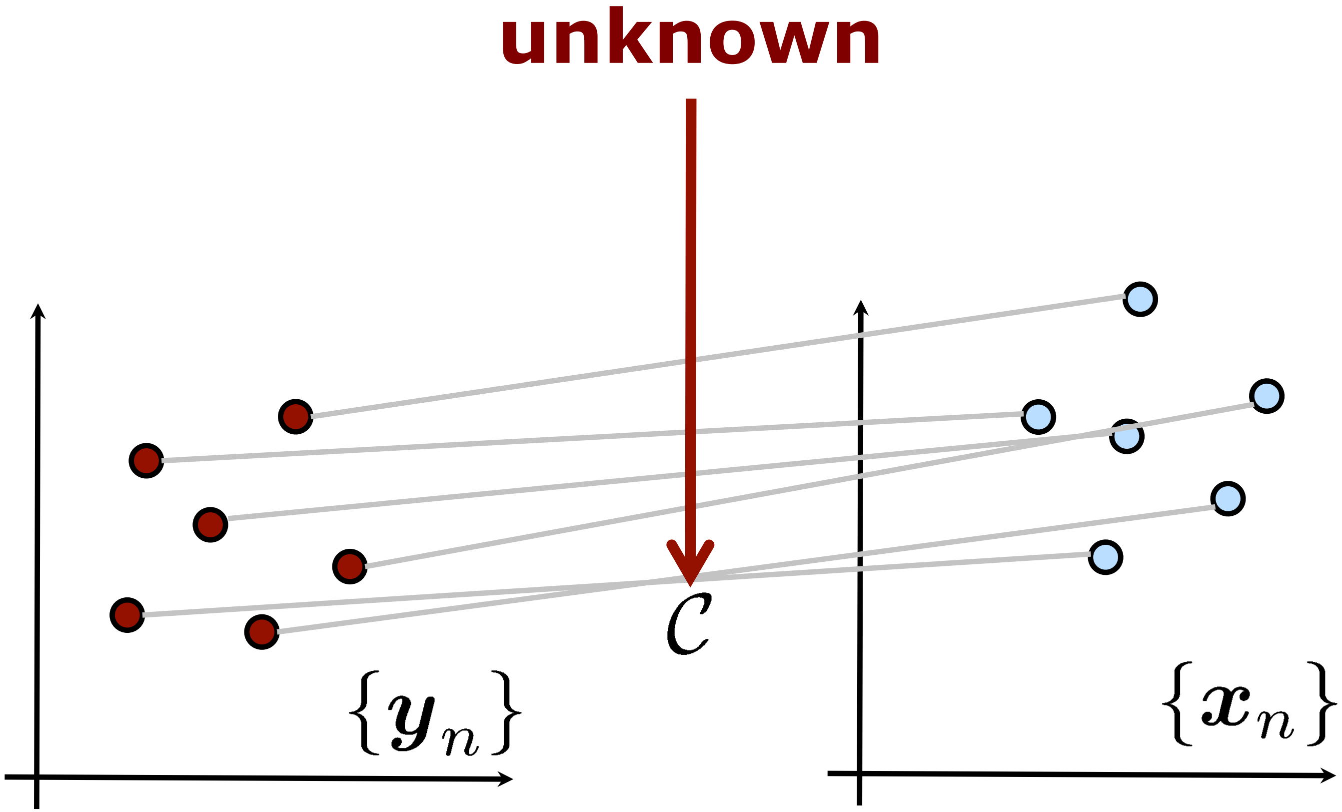
Simple Form of Point Cloud Registration



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

This Becomes Tricky if the Correspondences are Unknown

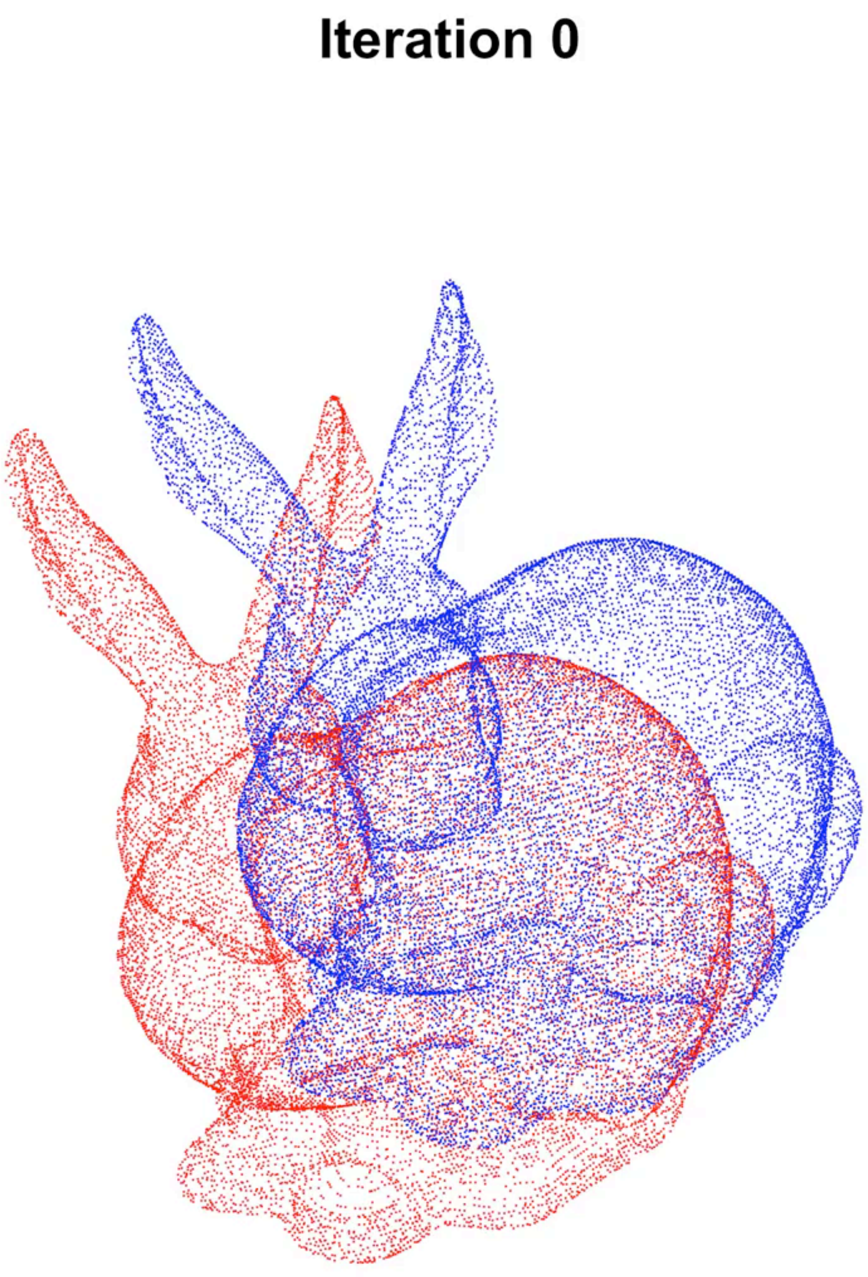
Simple Form of Point Cloud Registration



Registration of 3D Data Points

- **Goal:** find the parameters of the transformation that best align corresponding data points
- Optimization / search for parameters
  - Iterative closest point (ICP w/ SVD)
  - Robust least squares approaches (#3)
- Known (#1) vs. **estimated (#2)** correspondences

3D Point Cloud Registration Example



[Video courtesy: P. Glira]

Reminder

Part 1
Point Cloud Registration with Known Data Association

We have derived an efficient to compute, optimal, direct solution

Formal Problem Definition

Reminder

- **Given corresponding points:**

$$\boldsymbol{y}_n, \boldsymbol{x}_n \qquad n = 1, \ldots, N$$

- and optionally weights:

$$p_n \qquad n = 1, \ldots, N$$

- Find the parameters $R, \boldsymbol{t}$ of the rigid body transform with

$$\bar{\boldsymbol{x}}_n = R\boldsymbol{x}_n + \boldsymbol{t} \qquad n = 1, \ldots, N$$

- so that the squared error is minimized

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

Reminder

Solution for Computing the Rigid Body Transform

- Rotation
- Translation

$$\boxed{\begin{aligned} R &= VU^\top \\ \boldsymbol{t} &= \boldsymbol{y}_0 - R\boldsymbol{x}_0 \end{aligned}}$$

- with

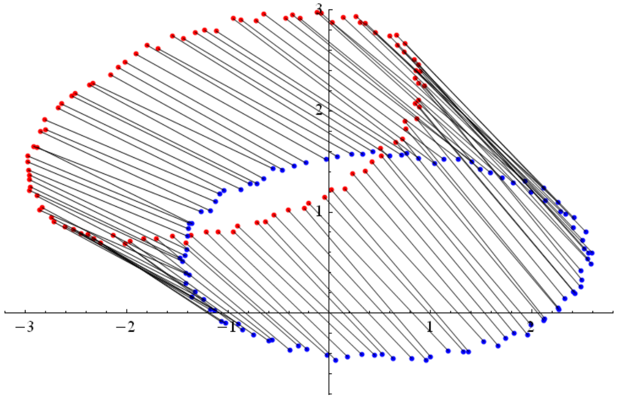
$$H = \sum (\boldsymbol{x}_n - \boldsymbol{x}_0)(\boldsymbol{y}_n - \boldsymbol{y}_0)^\top p_n \quad \mathrm{svd}(H) = UDV^\top$$

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

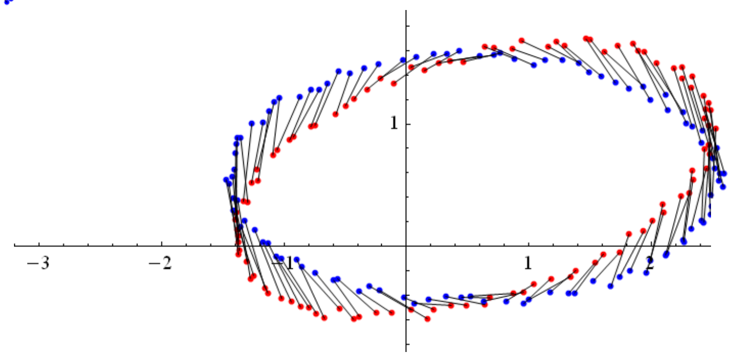
Reminder

Alignment Summary

Alignment through translation and rotation $\bar{\boldsymbol{x}}_n = R(\boldsymbol{x}_n - \boldsymbol{x}_0) + \boldsymbol{y}_0$



translate points to make the center of masses overlap



rotate points

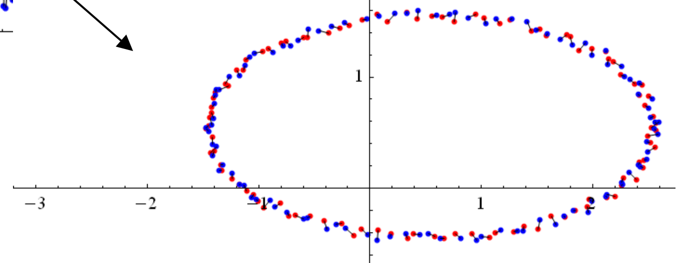


Image courtesy: Ju

Part 2
Point Cloud Registration with Unknown Data Association

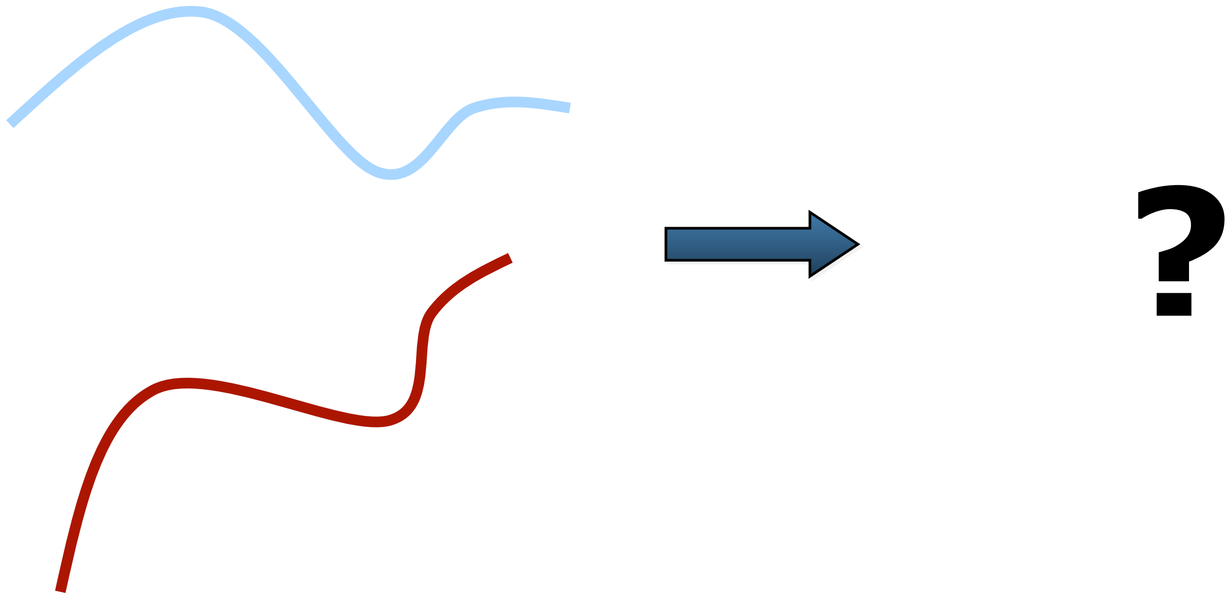
No direct and optimal solution exists

Iterative Closest Point (ICP)

[Chen & Medioni '91, Besl & McKay '92]

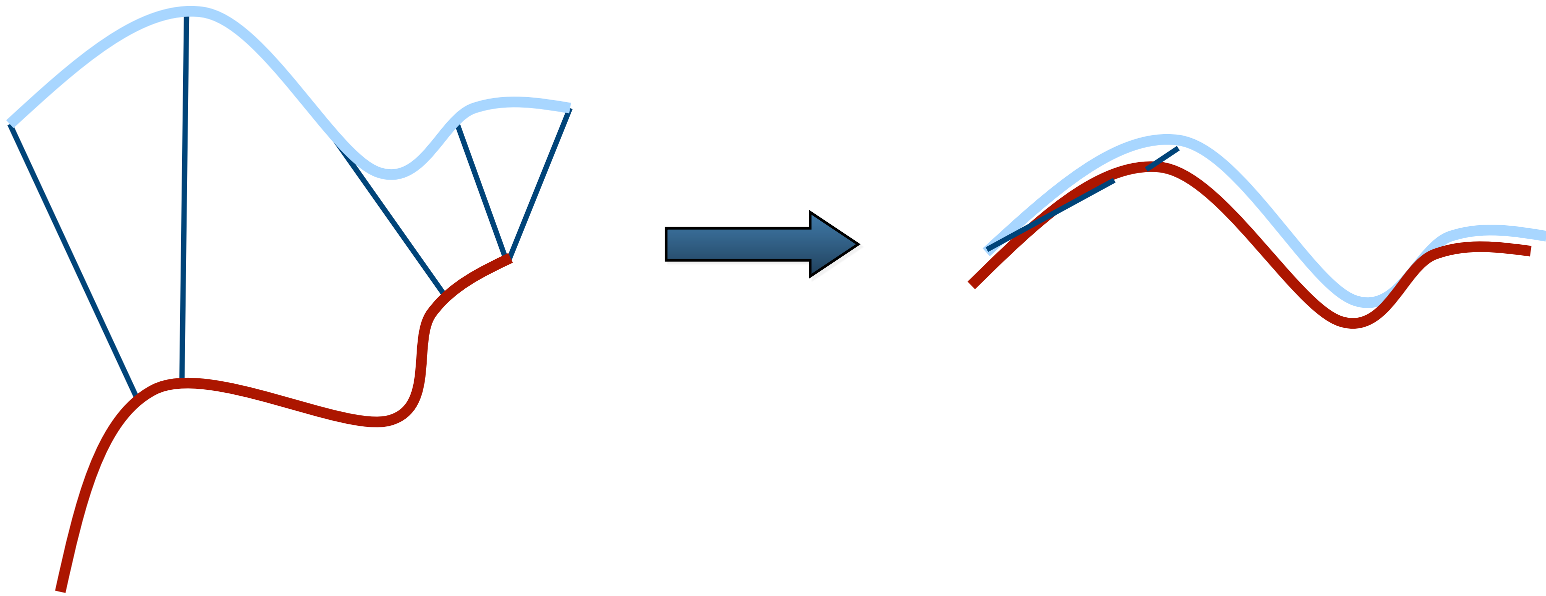
ICP: Point Cloud Registration Estimating the Data Association

If the correct correspondences are **not known**, it is generally impossible to determine the optimal parameters in one step



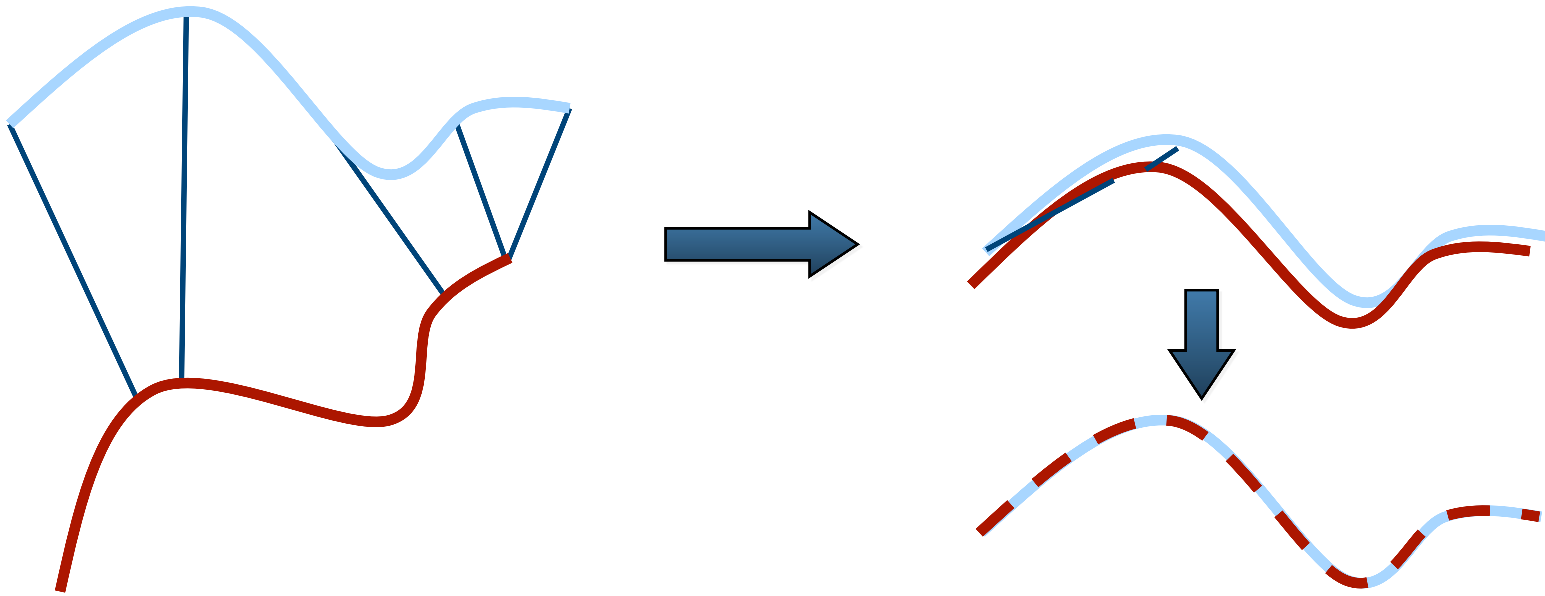
ICP: Point Cloud Registration Estimating the Data Association

If the correct correspondences are **not known**, it is generally impossible to determine the optimal parameters in one step



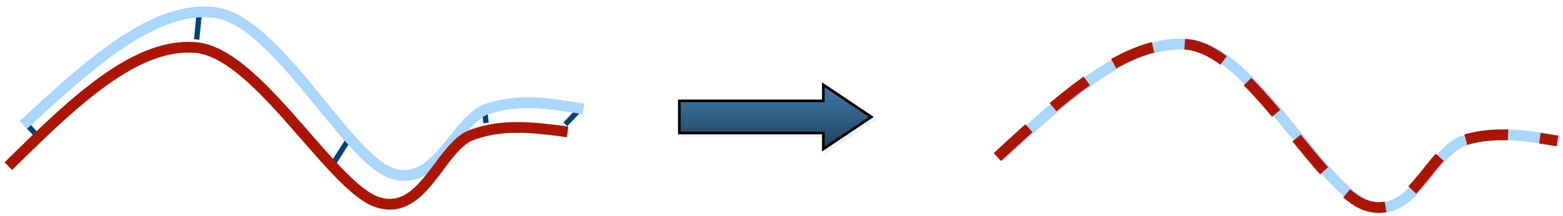
ICP: Point Cloud Registration Estimating the Data Association

If the correct correspondences are **not known**, it is generally impossible to determine the optimal parameters in one step – **but we can iterate!**



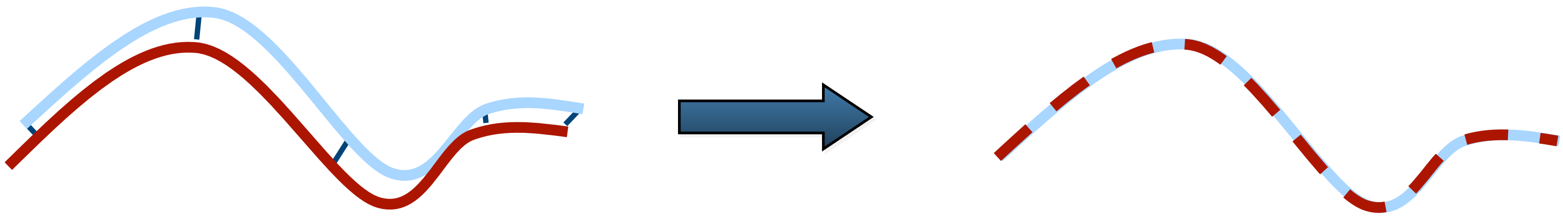
Iterative Closest Point (ICP) Algorithm

- Idea: Iteratively estimate the data association and transformation
- "A Method for Registration of 3-D Shapes" [Besl & McKay 92]
- Assumption: We have an initial guess
  - point locations or
  - point correspondences

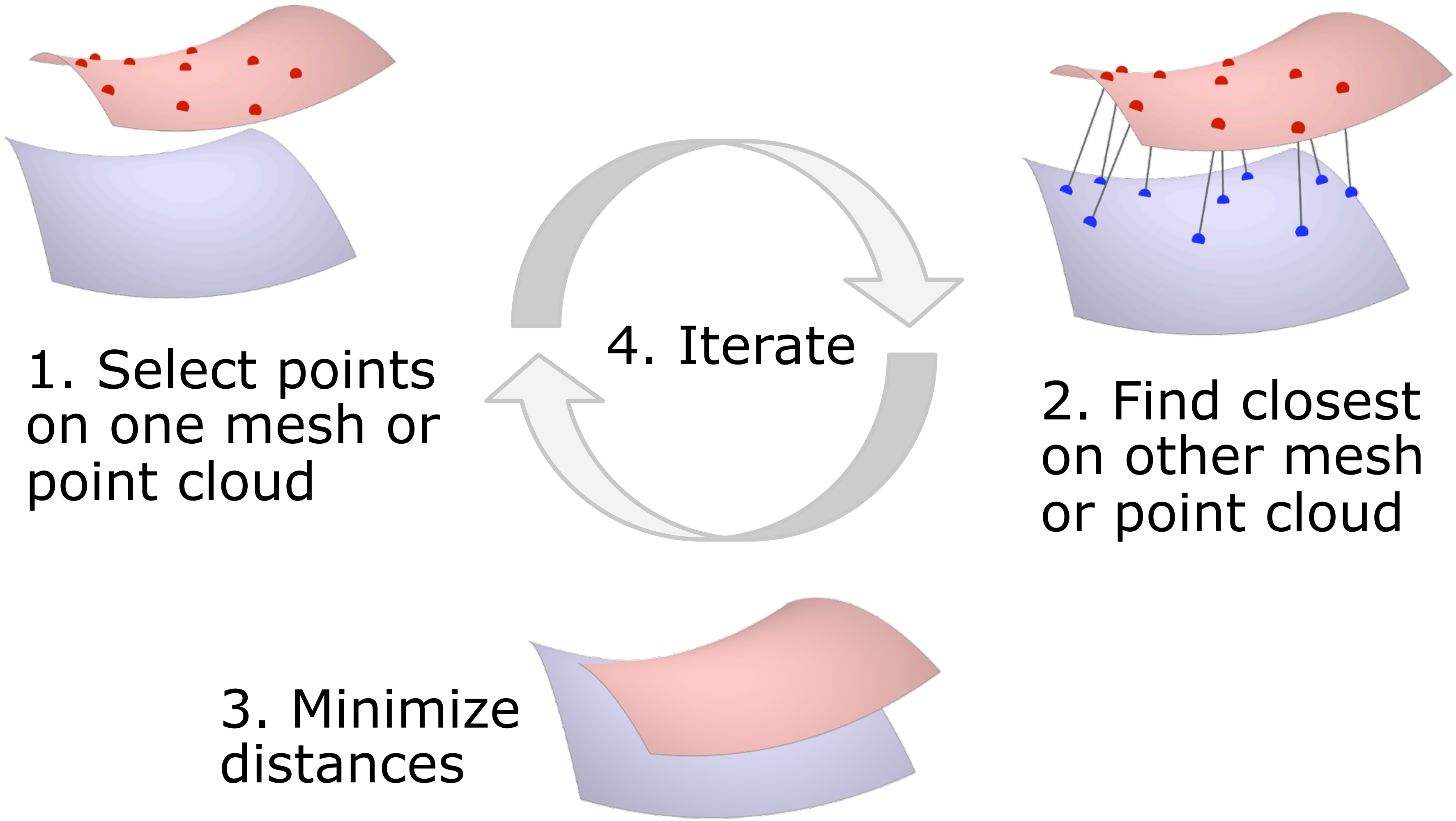


Iterative Closest Point (ICP) Algorithm

- Iterate estimating the alignment
  - Pick for every point its closest neighbor in the other point cloud (“closest point”)
  - Compute the rigid body transform & align
  - Repeat
- Converges if initial point clouds (or correspondences) are “close enough”



ICP Illustrated



[Courtesy of Rusinkiewicz]

Basic ICP Algorithm

$\bar{\boldsymbol{x}}_n = \boldsymbol{x}_n$

error $e = \infty$

while (e has decreased and $e >$ threshold)

$\mathcal{C}$ = determine_correspondences $(\{\boldsymbol{y}_n, \bar{\boldsymbol{x}}_n\})$

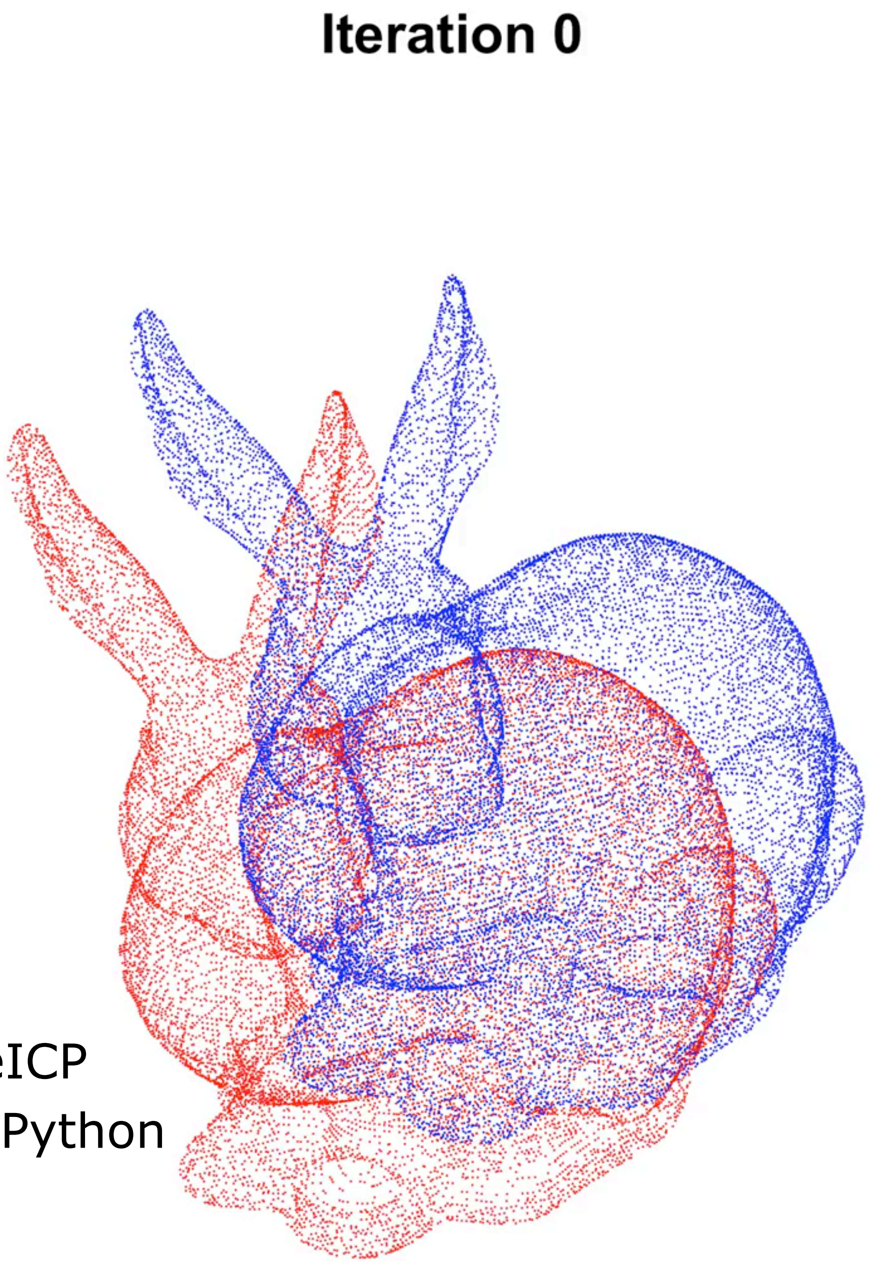
$(\boldsymbol{t}, R)$ = compute_transformation_params $(\mathcal{C})$

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

$e = E(\boldsymbol{t}, R) = \Phi(R^\top \boldsymbol{y}_0 - R^\top \boldsymbol{t}, R)$

return $\{\bar{\boldsymbol{x}}_n\}$

ICP Example



SimpleICP by Philipp Glira
https://github.com/pglira/simpleICP
C++, Matlab, Julia, Octave, and Python

[Video courtesy: Glira]

Vanilla ICP

- The Vanilla ICP approach is easy to implement
- Works if a good initial guess is available

But…

- May require many iterations
- Bad correspondences can seriously degrade the quality of the result

ICP Variants

Variants on the following stages of ICP have been proposed:

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

Performance of Variants

Various aspects of performance:

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

ICP Variants

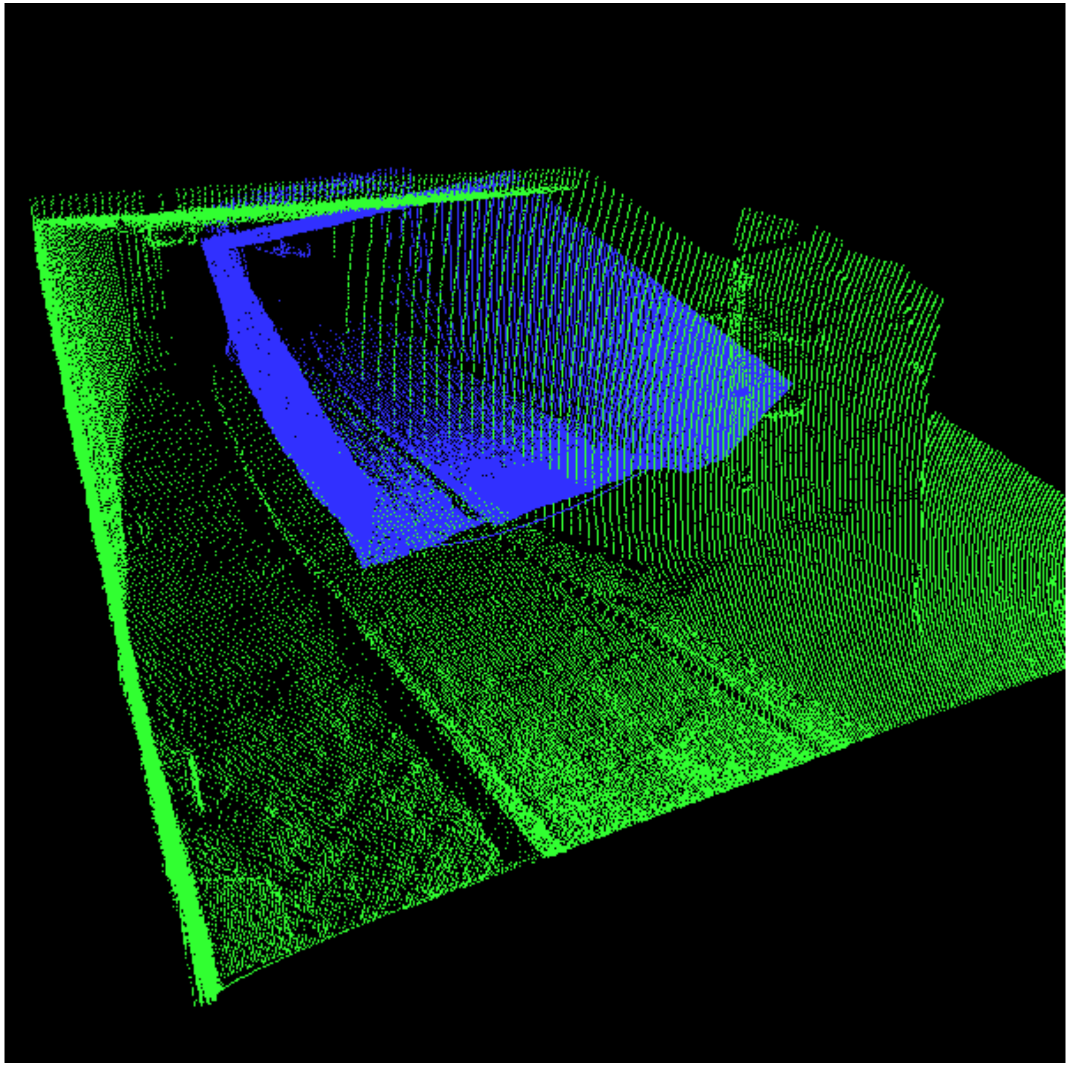
Variants on the following stages of ICP have been proposed:

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

Selecting Source Points

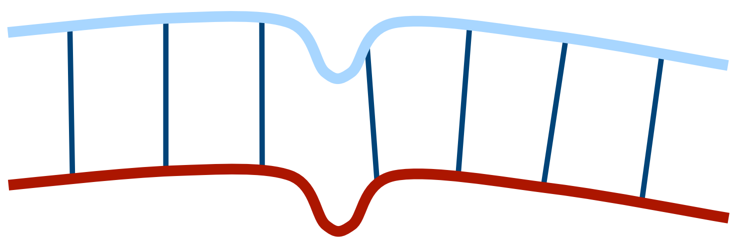
- Use all points
- Uniform sub-sampling
- Random sampling
- Feature-based sampling
- Normal-space sampling
  (Ensure that samples have normals distributed as uniformly as possible)

ICP with Uniform Sampling

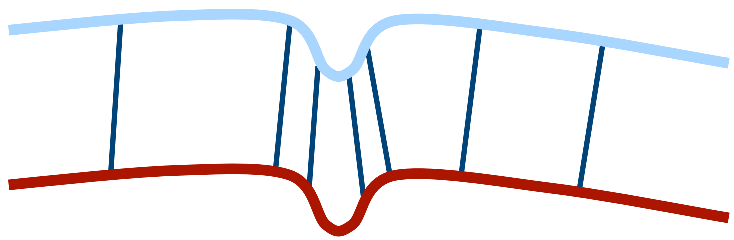


Video courtesy: Nuechter

Uniform vs. Normal-Space Sampling



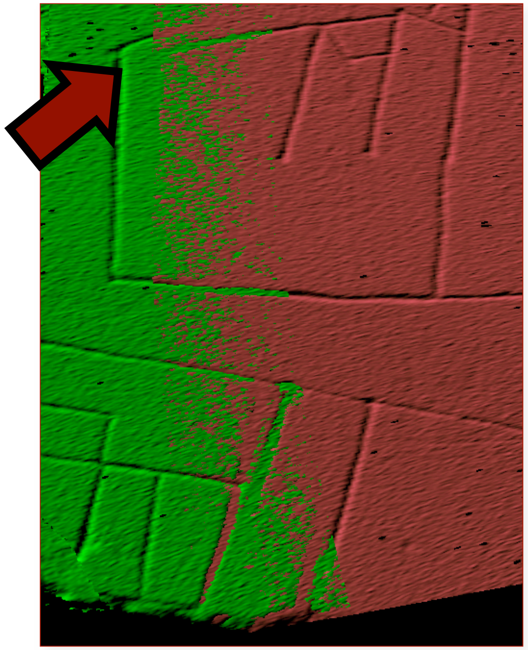
uniform sampling



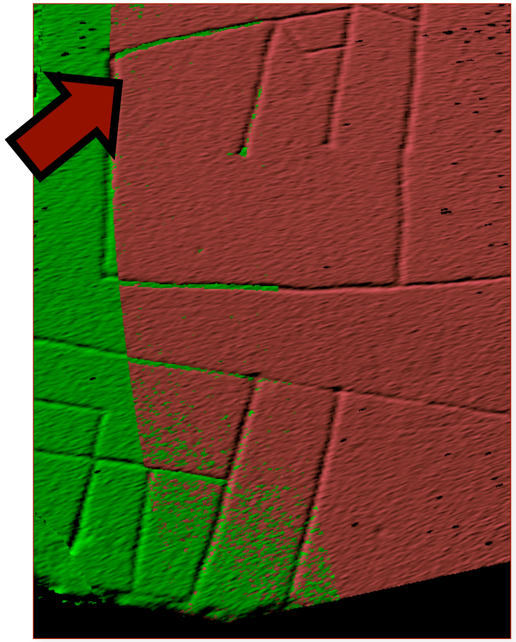
normal-space sampling

Comparison

- Normal-space sampling is better for mostly smooth areas with sparse features [Rusinkiewicz et al., 01]



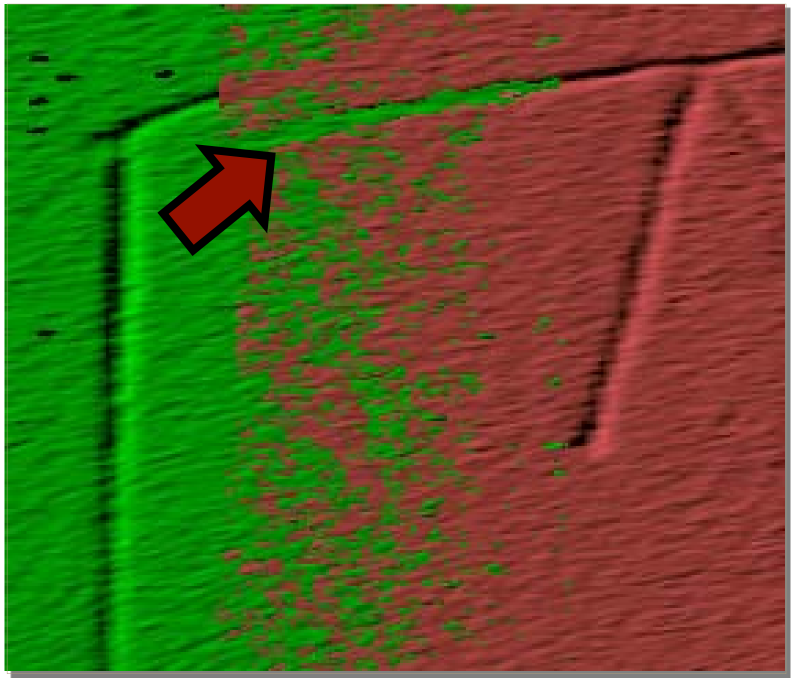
random sampling



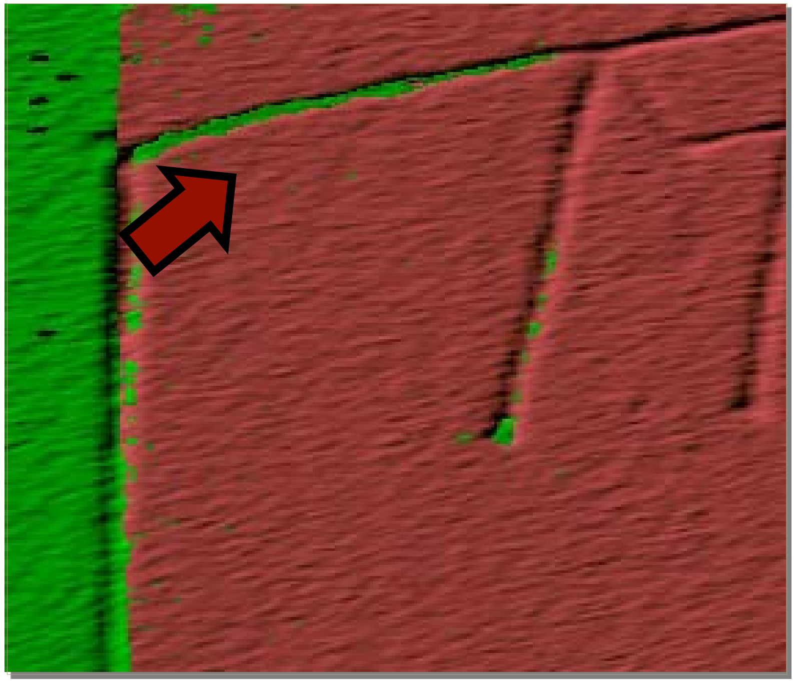
normal-space sampling

Comparison

- Normal-space sampling is better for mostly smooth areas with sparse features [Rusinkiewicz et al., 01]



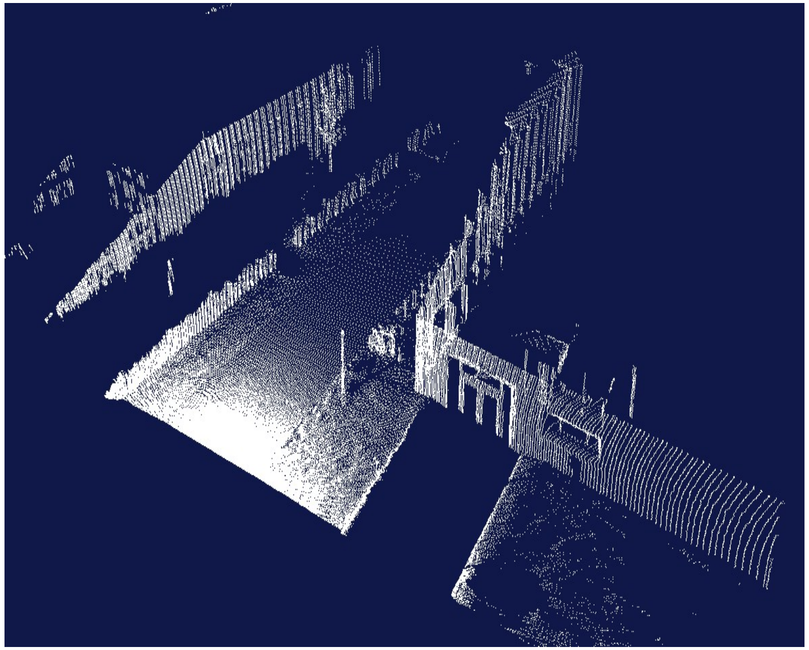
random sampling



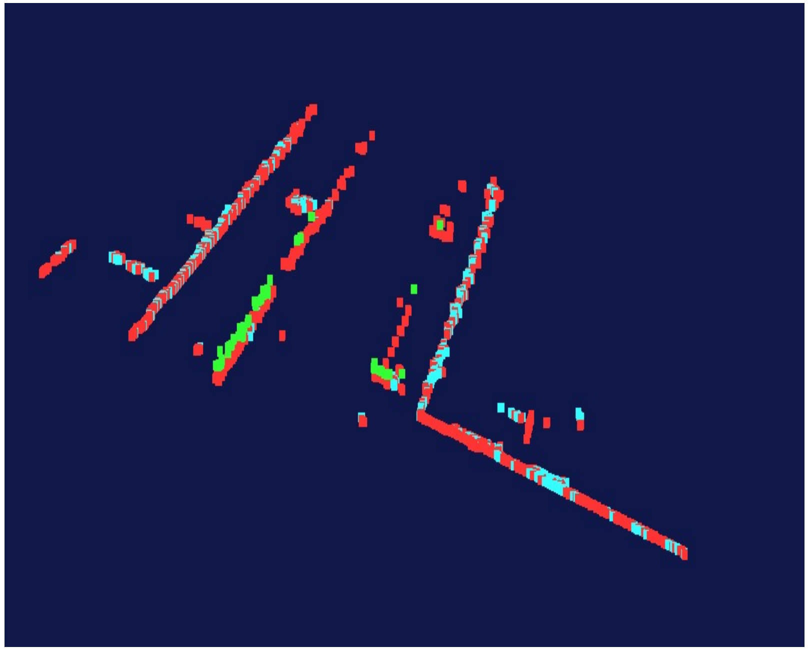
normal-space sampling

Feature-Based Sampling

- Try to work only with highly distinct points
- Simplifies the search for correspondences
- Higher efficiency and sometimes better accuracy
- Requires preprocessing



Full 3D scan (~200.000 points)



Extracted features (~5.000 points)

ICP Variants

Variants on the following stages of ICP have been proposed:

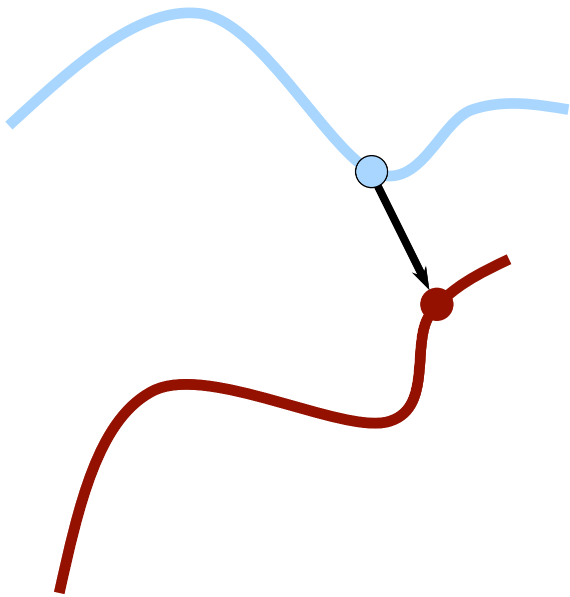
1. Consider point subsets
2. Different data association strategies
3. Weight the correspondences
4. Reject potential outlier point pairs

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 Matching

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



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

No Initial Guess?

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

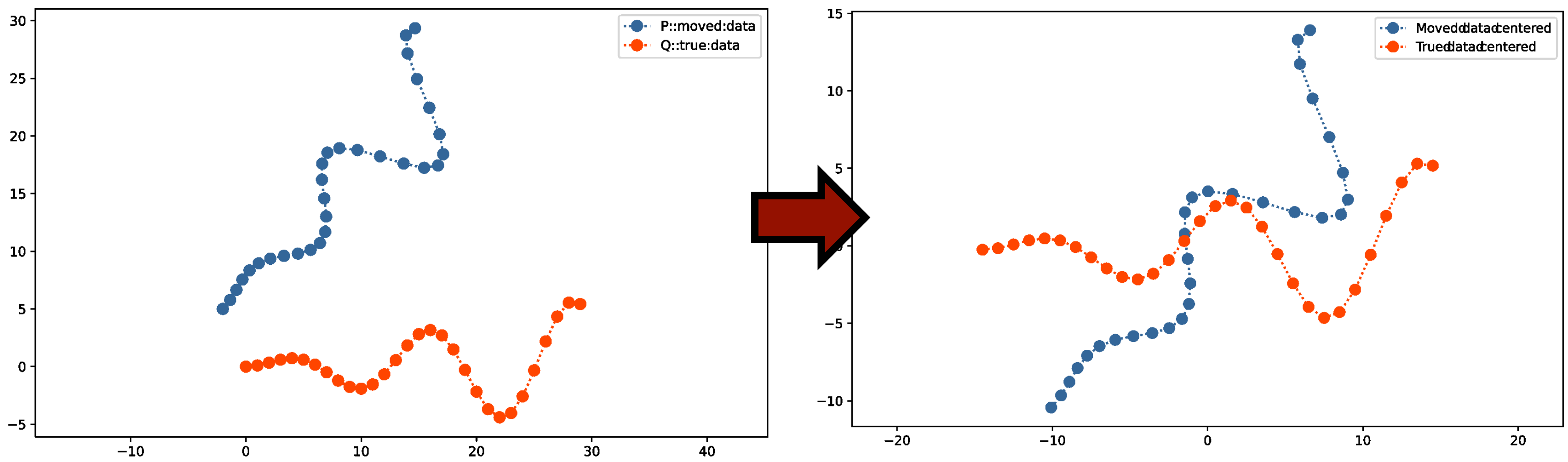


Image courtesy: Bogoslavskyi

Starting Configuration

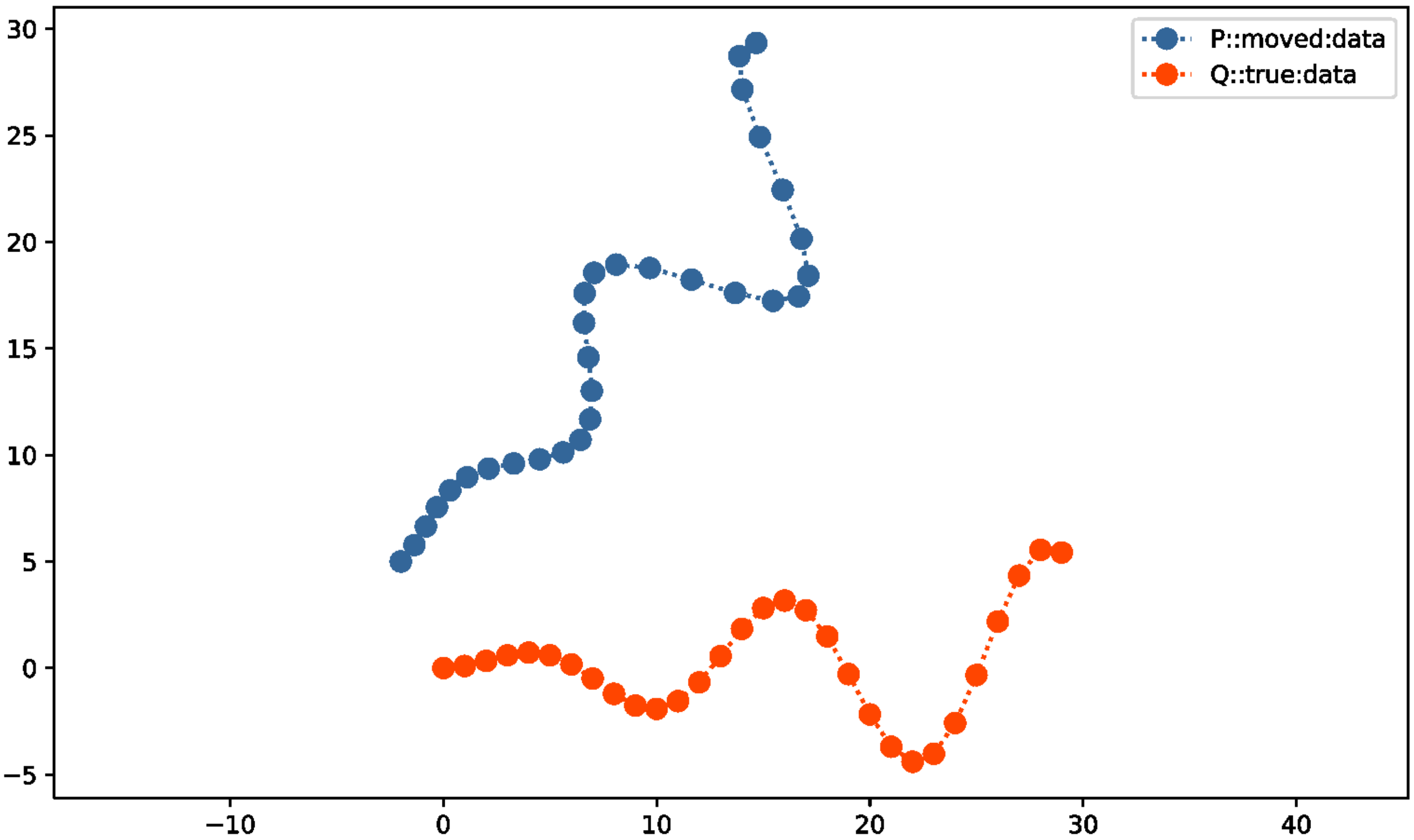


Image courtesy: Bogoslavskyi

Align Center of Masses by Shift

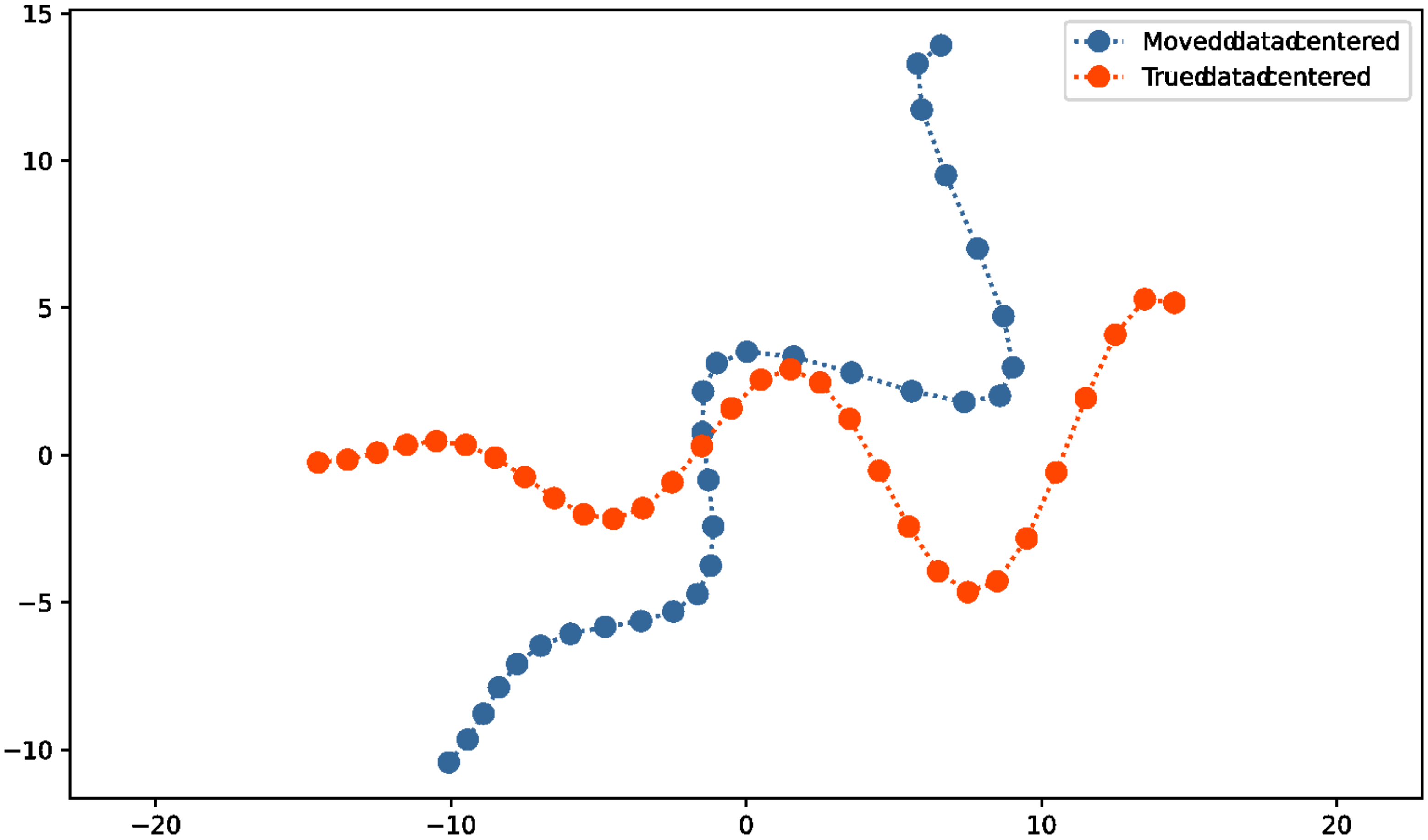


Image courtesy: Bogoslavskyi

Nearest Neighbor Assignment

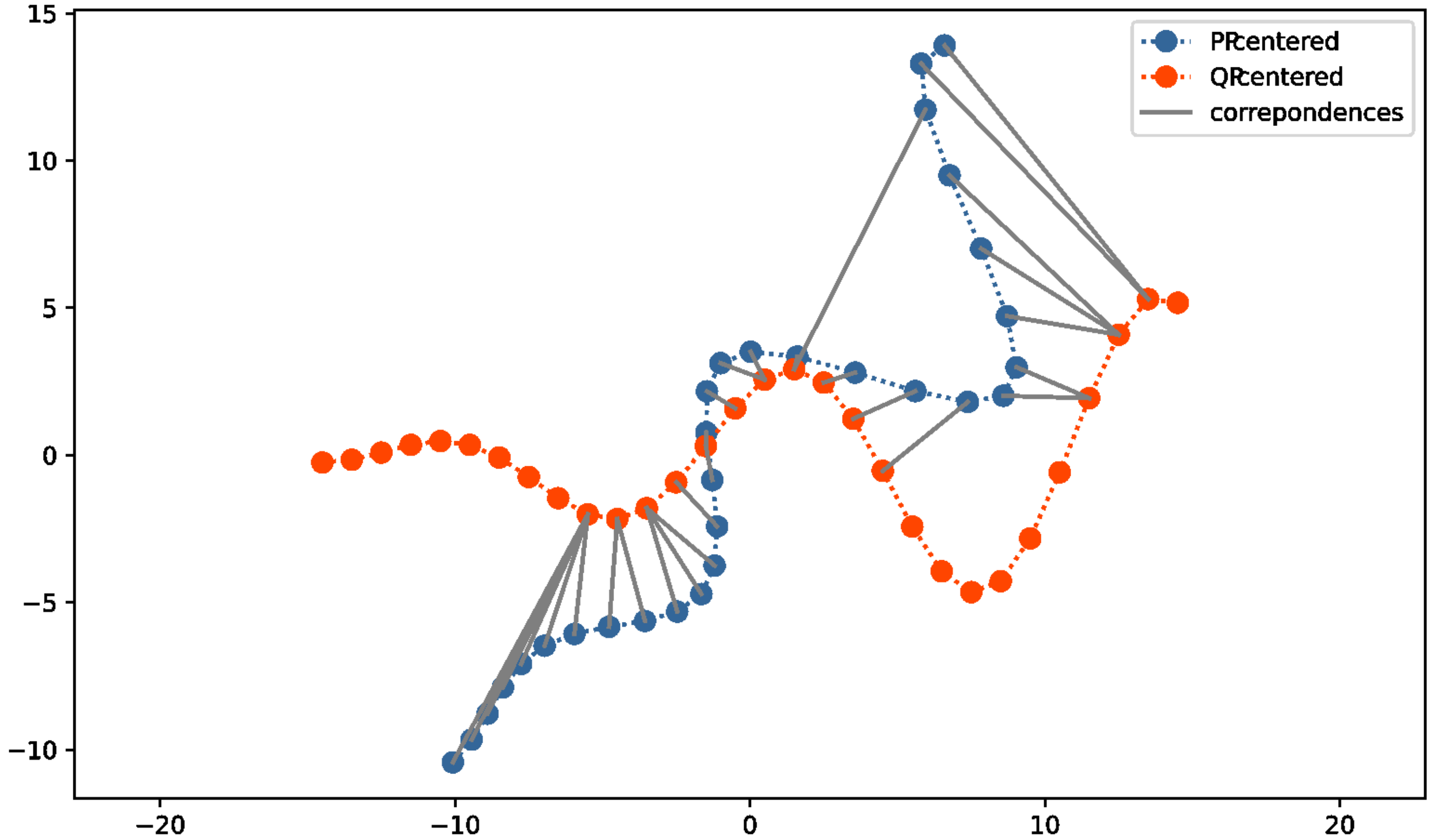


Image courtesy: Bogoslavskyi

Compute Transformation, Align

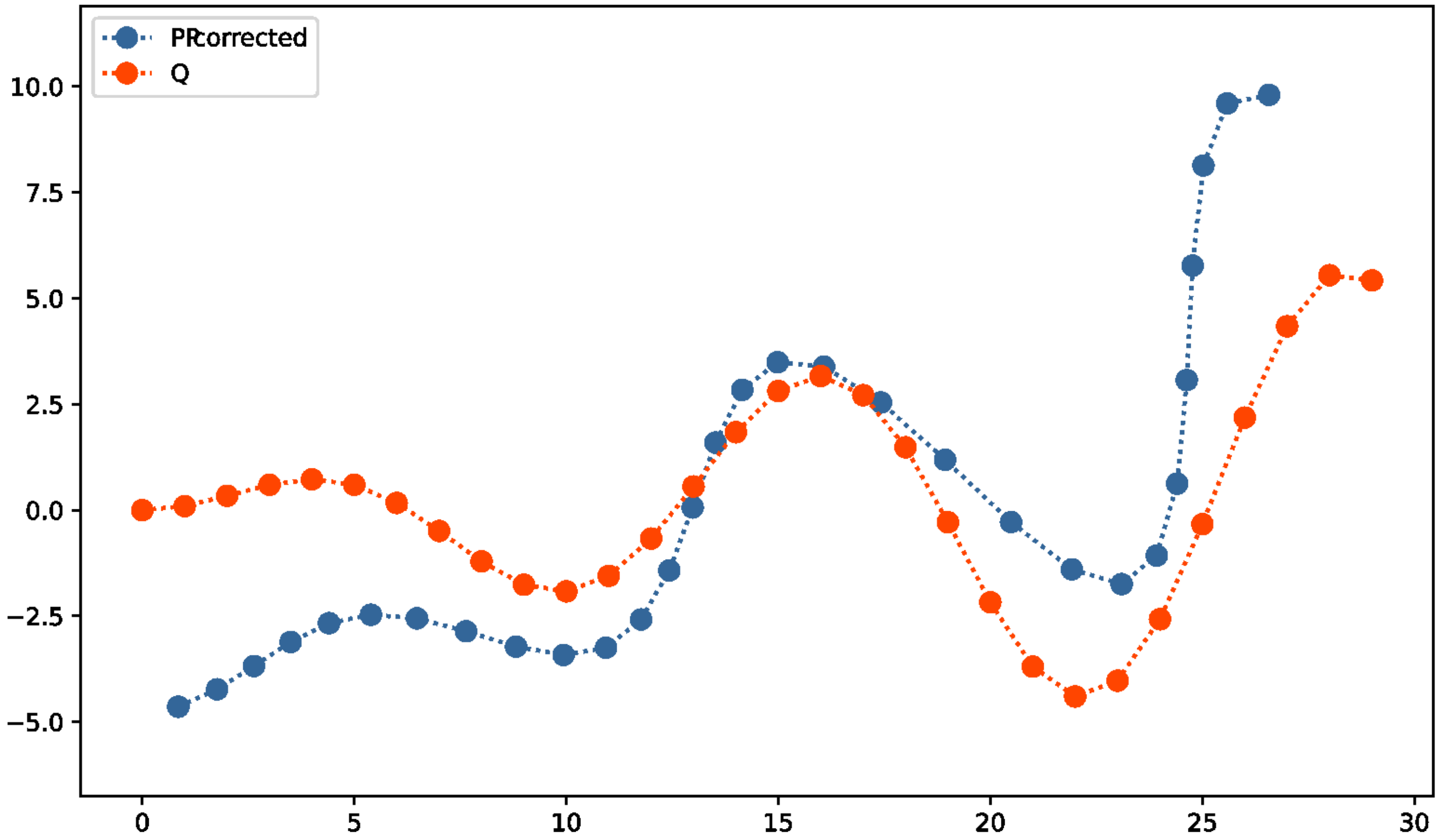


Image courtesy: Bogoslavskyi

Iterate

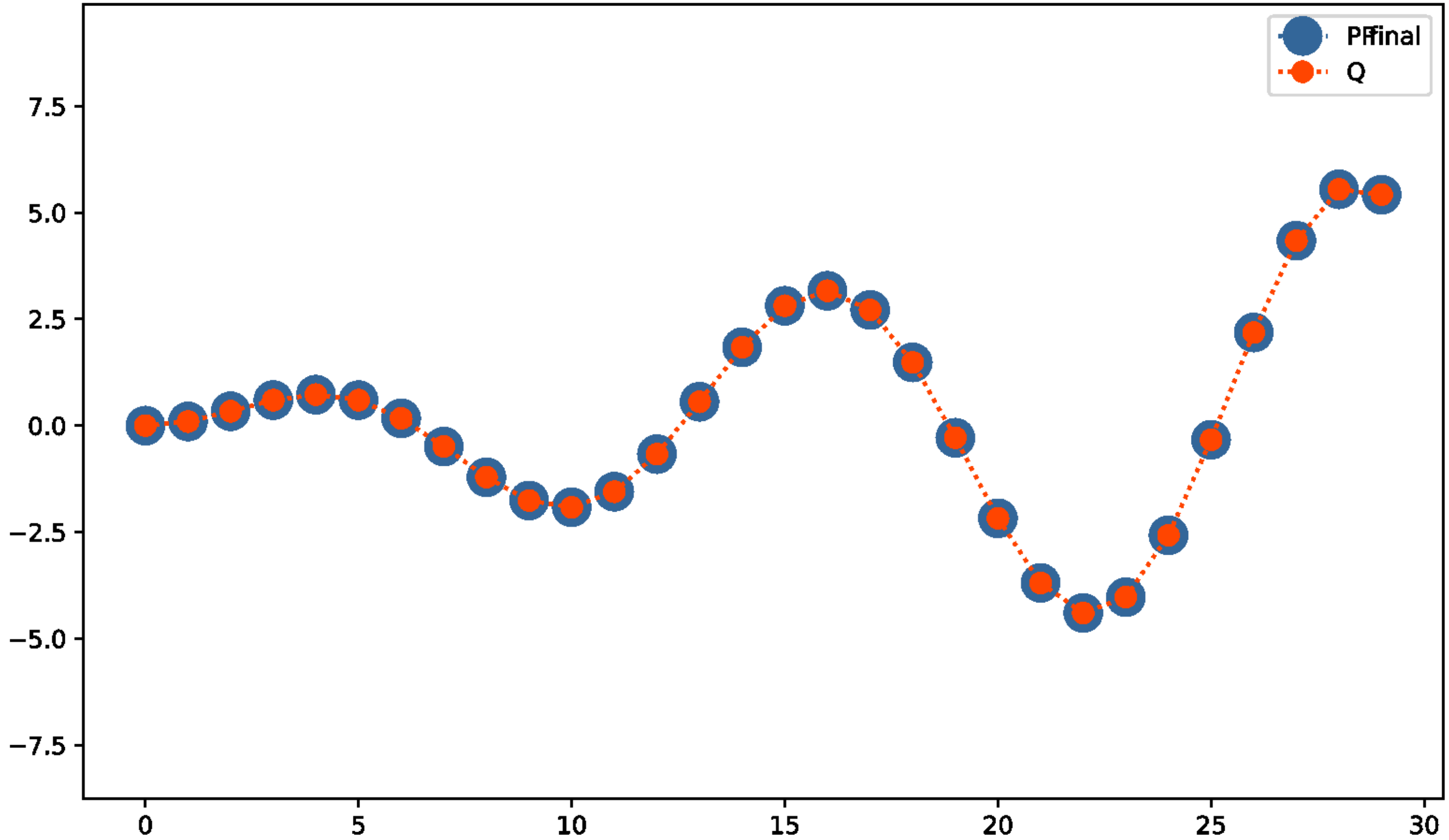


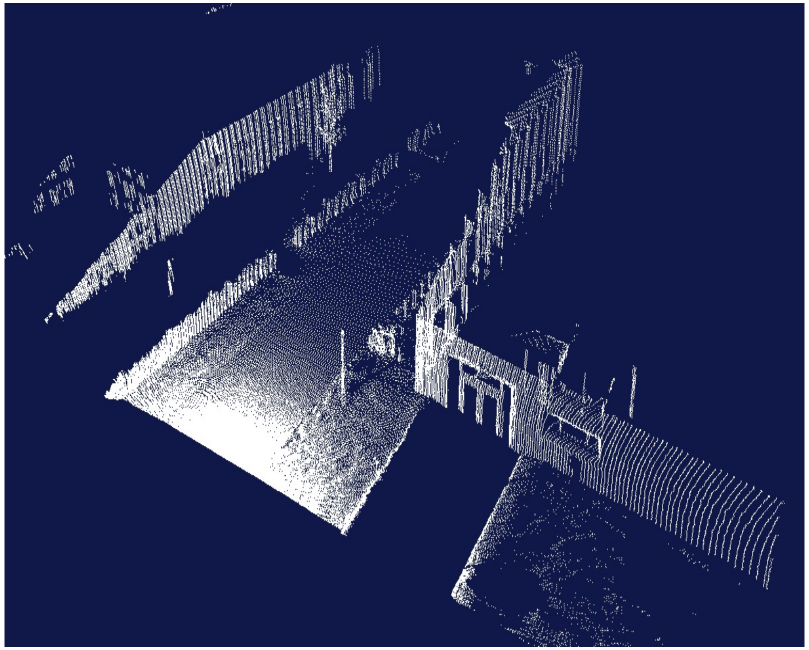
Image courtesy: Bogoslavskyi

Closest Compatible Point

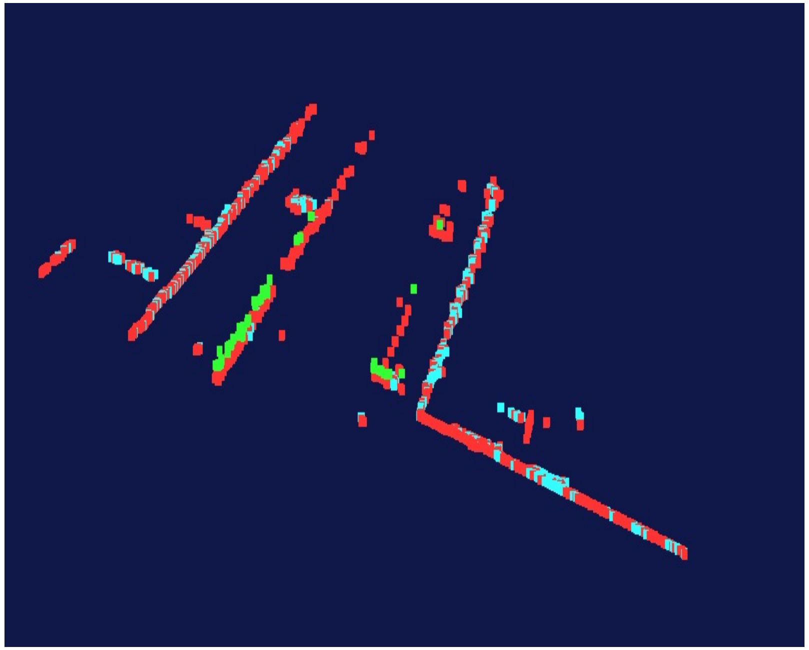
- Robustification by considering the **compatibility** of the points
- Only matches compatible points
- Compatibility can be based on
  - Normals
  - Colors
  - Curvature
  - Higher-order derivatives
  - Other local features

Feature Compatibility

Match only points that have compatibly features



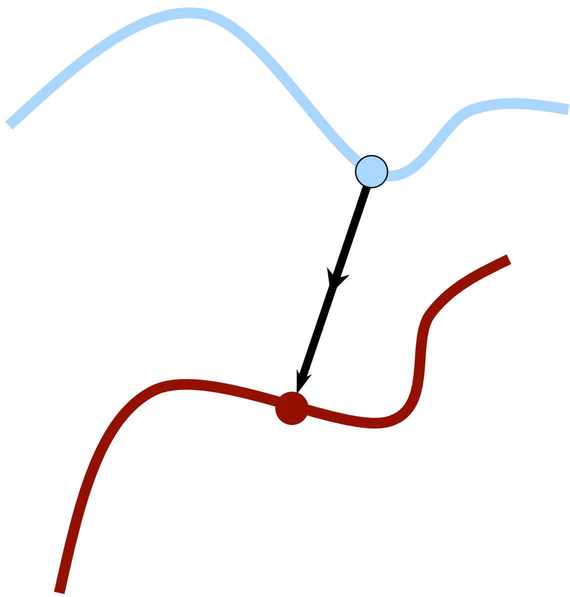
Full 3D scan (~200.000 points)



Extracted features (~5.000 points)

Normal Shooting

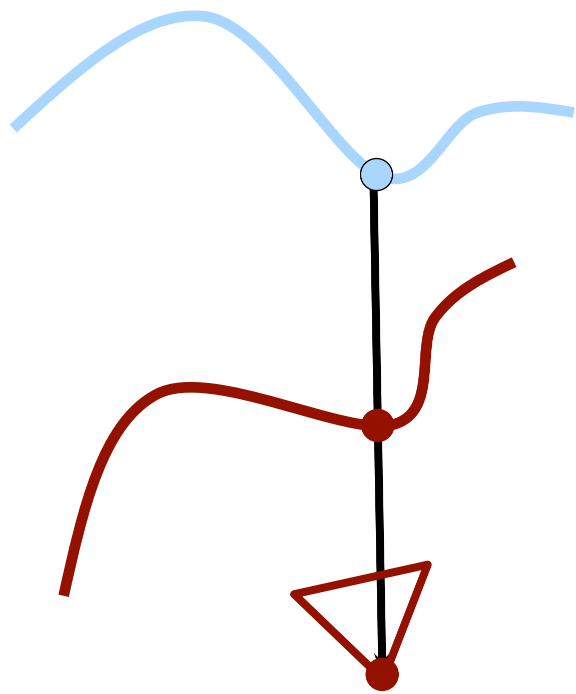
Project along normal, intersect other point set to find a correspondence



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

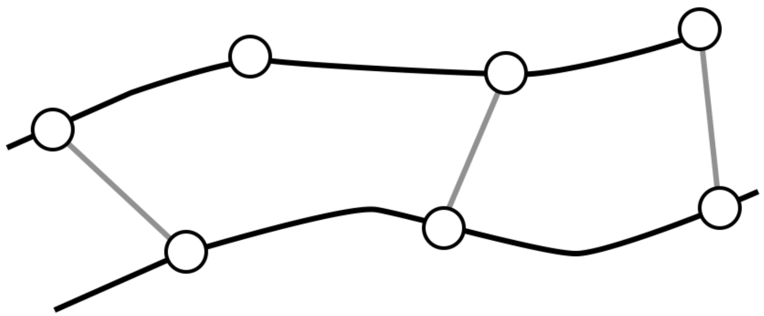
Projective Data Association

Searches for correspondences by projecting a point towards the sensor viewpoint

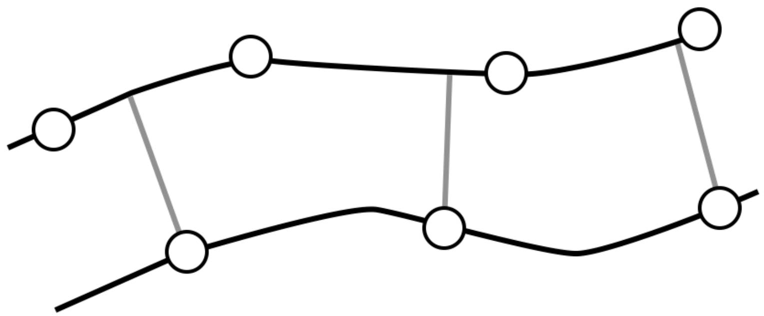


Point-to-Plane Metric

- Idea: still find the closest points
- Error = project point-to-point onto the direction of the normal, shot from the found point



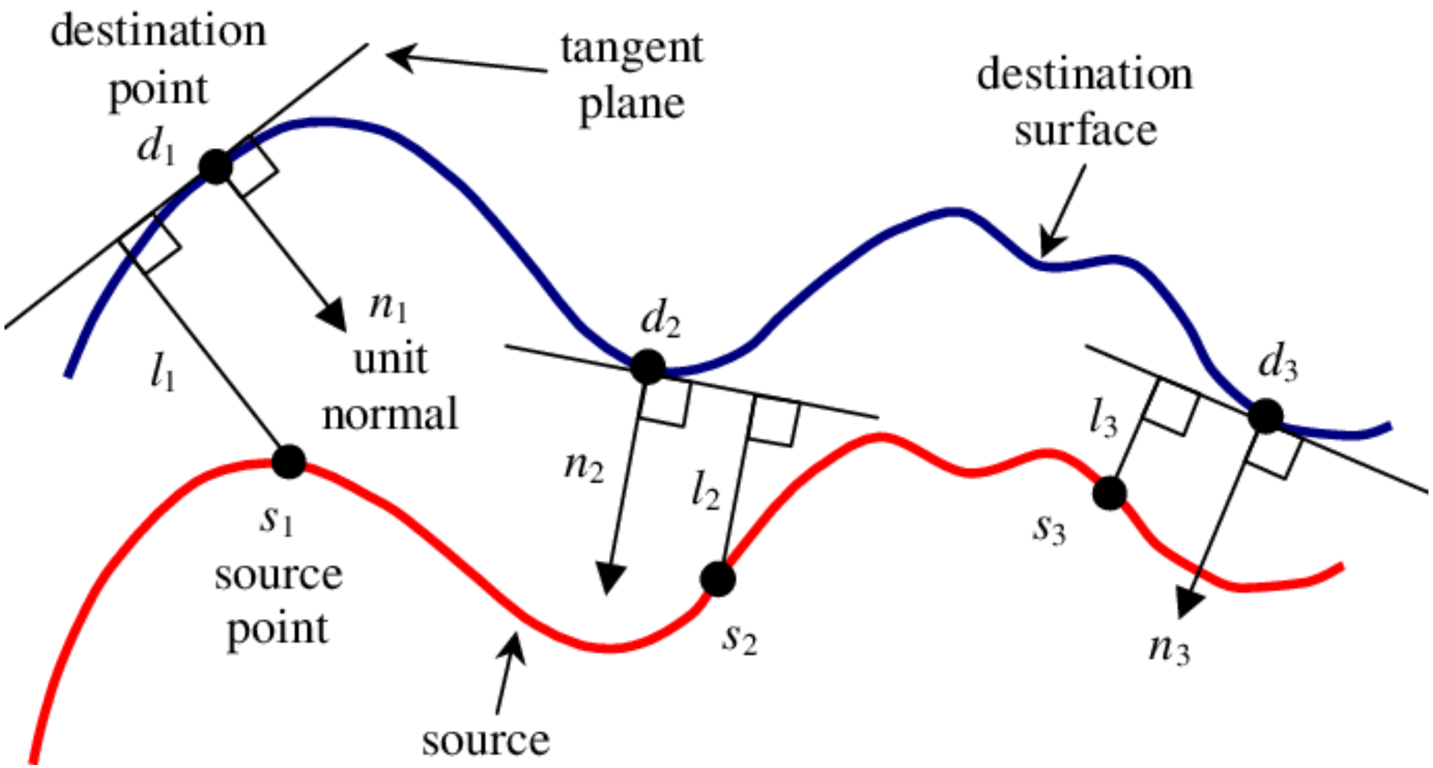
point-to-point



point-to-plane

Point-to-Plane Metric

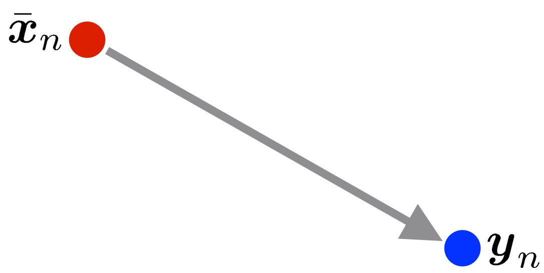
- Error = project point-to-point onto the direction of the normal, shot from the found point



[Image courtesy: Low]

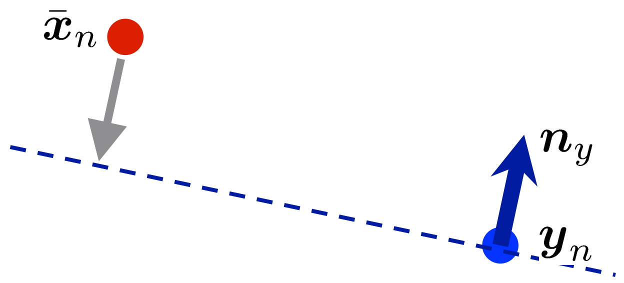
Point-to-Point vs Point-to-Plane

point-to-point



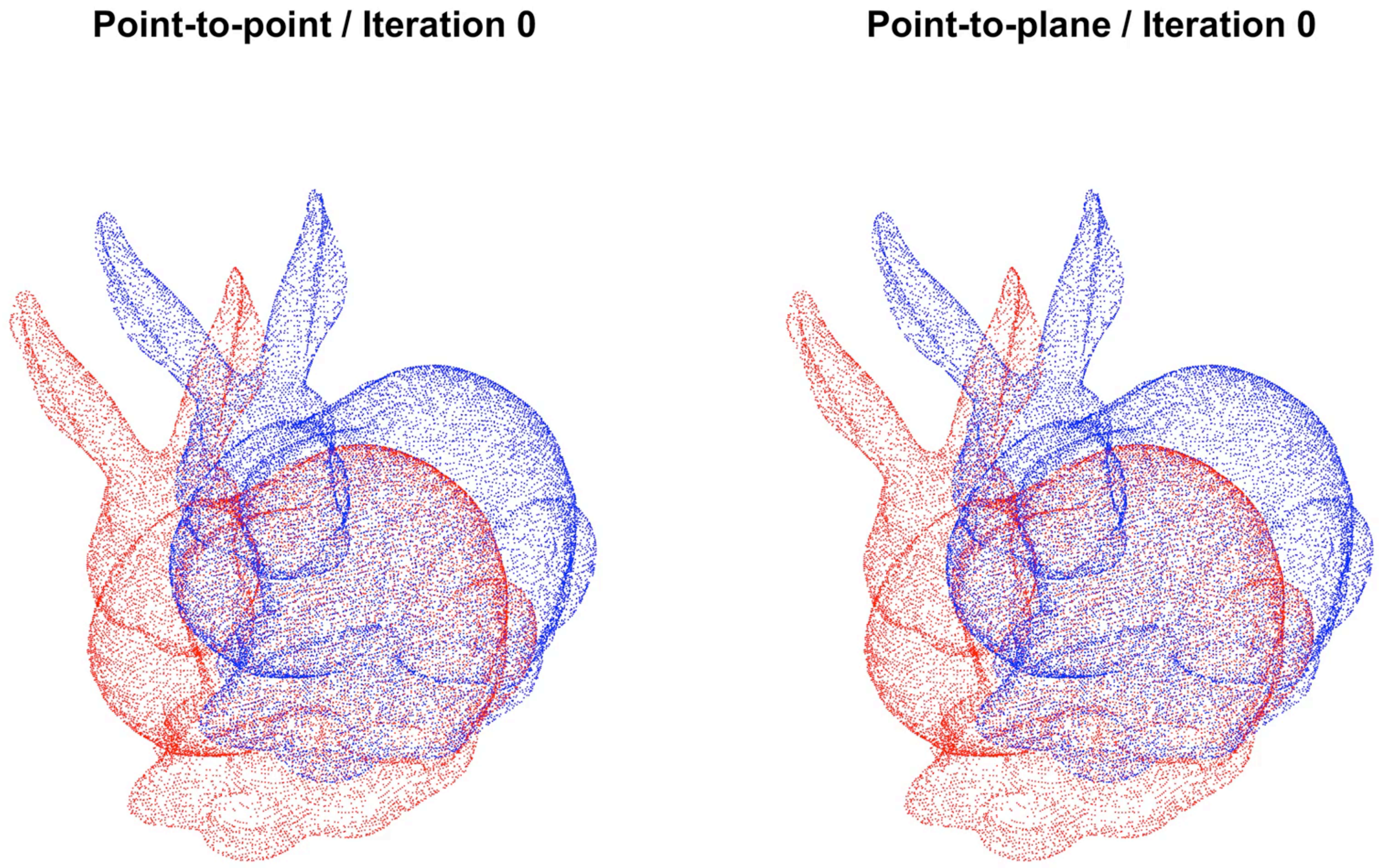
$$\min \sum ||\boldsymbol{y}_n - \bar{\boldsymbol{x}}_n||^2$$

point-to-plane



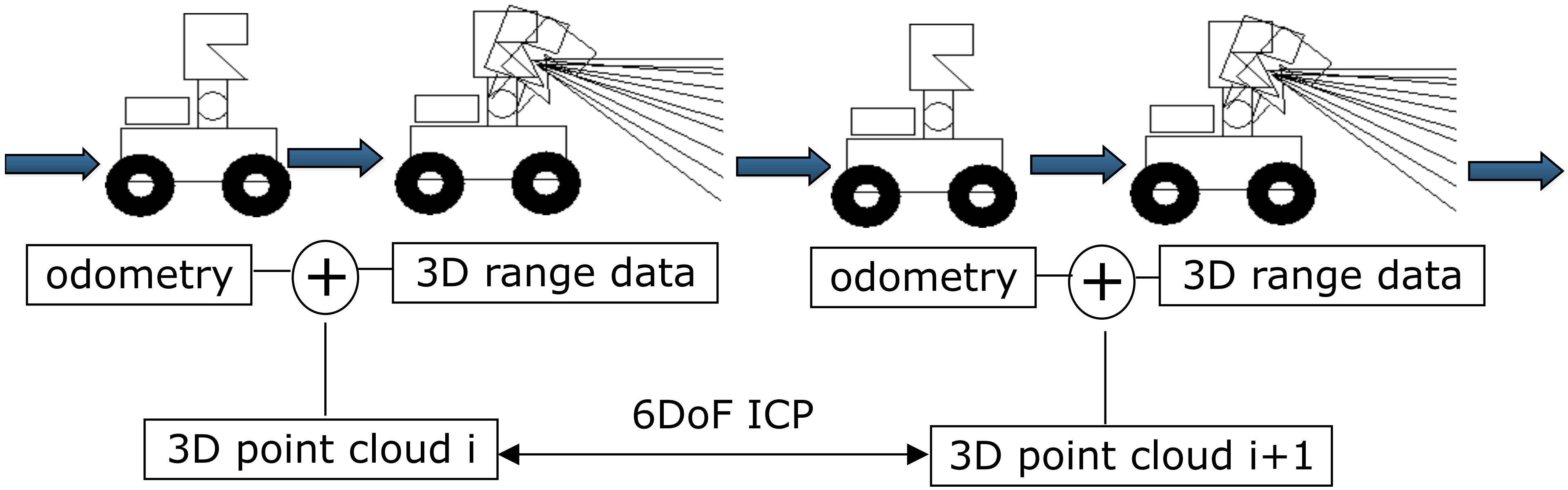
$$\min \sum \left((\boldsymbol{y}_n - \bar{\boldsymbol{x}}_n) \cdot \boldsymbol{n}_y\right)^2$$

Point-to-Point vs Point-to-Plane

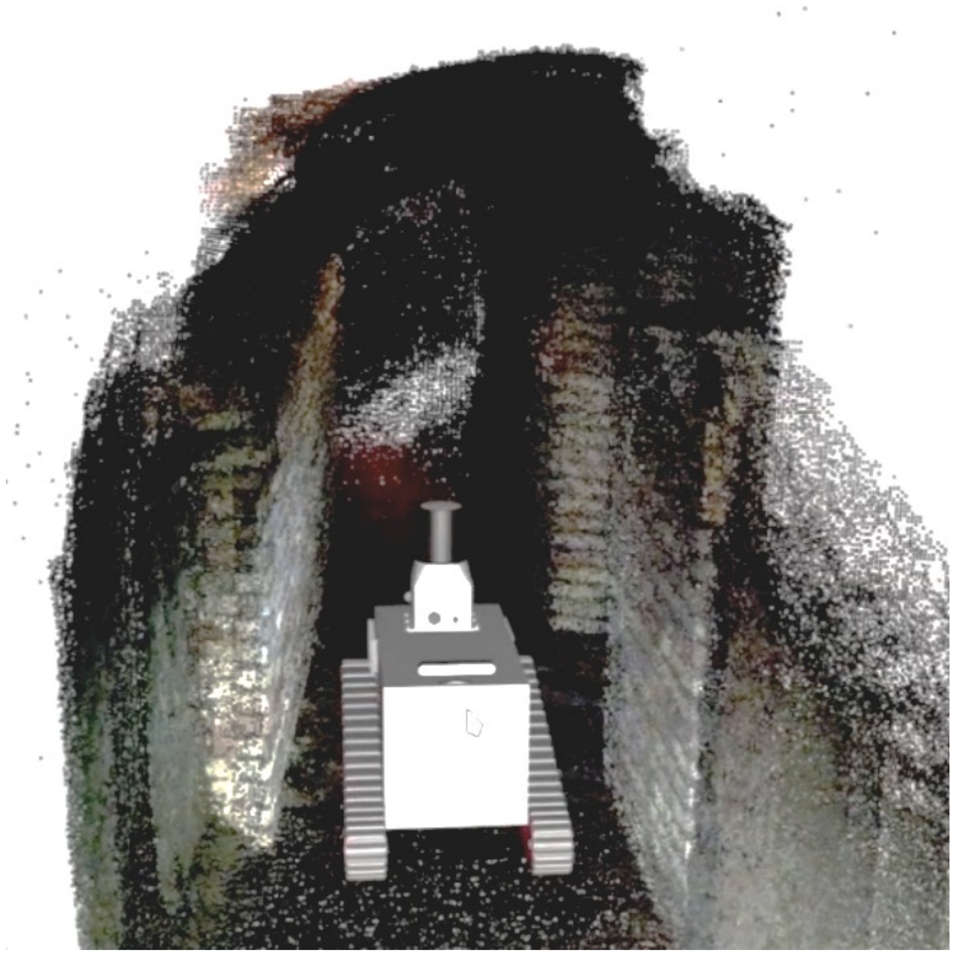


[Video courtesy: P. Glira]

ICP Example for Mapping



Kinect-Based Mapping



Projective Frame-to-Model Data Association

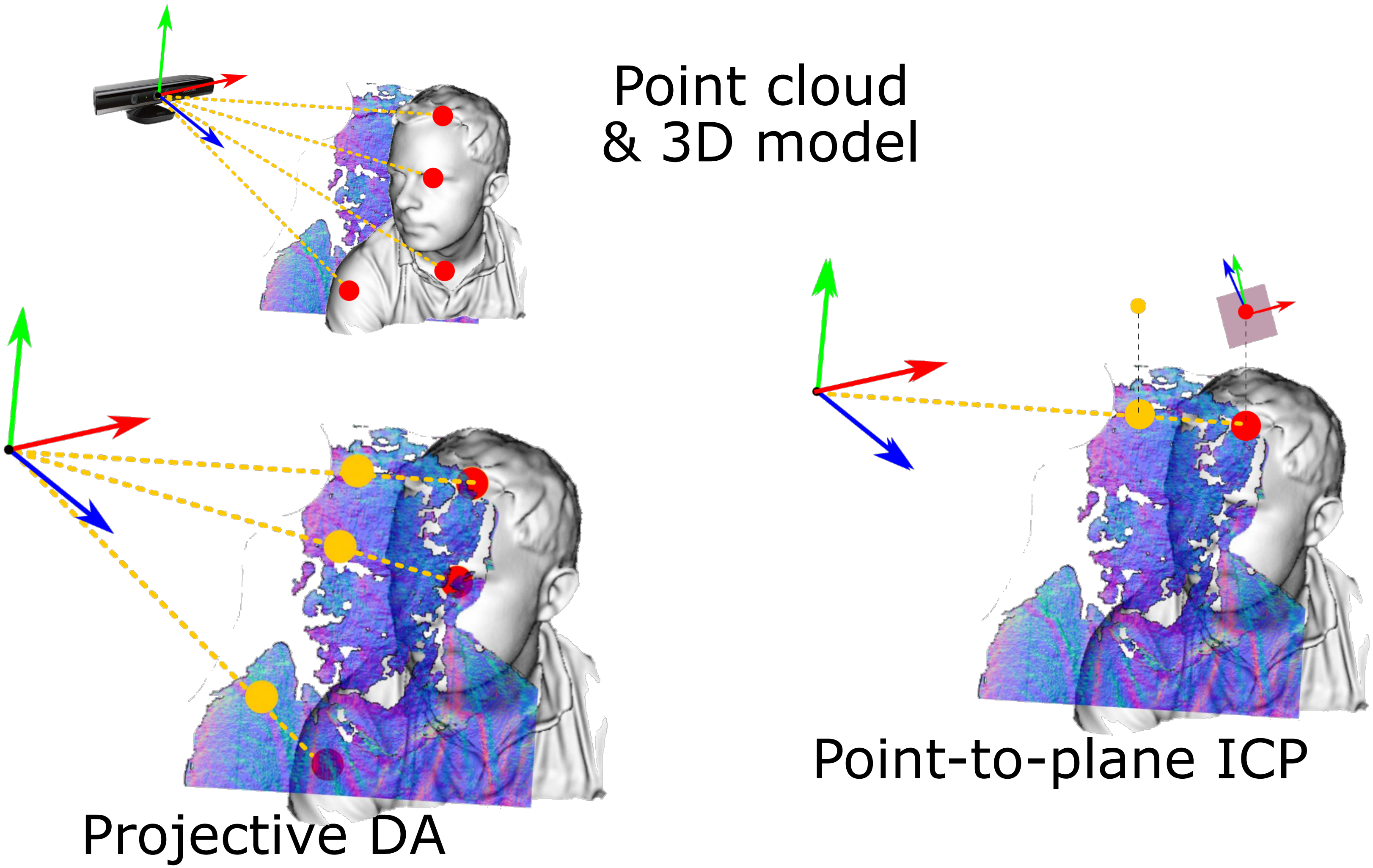
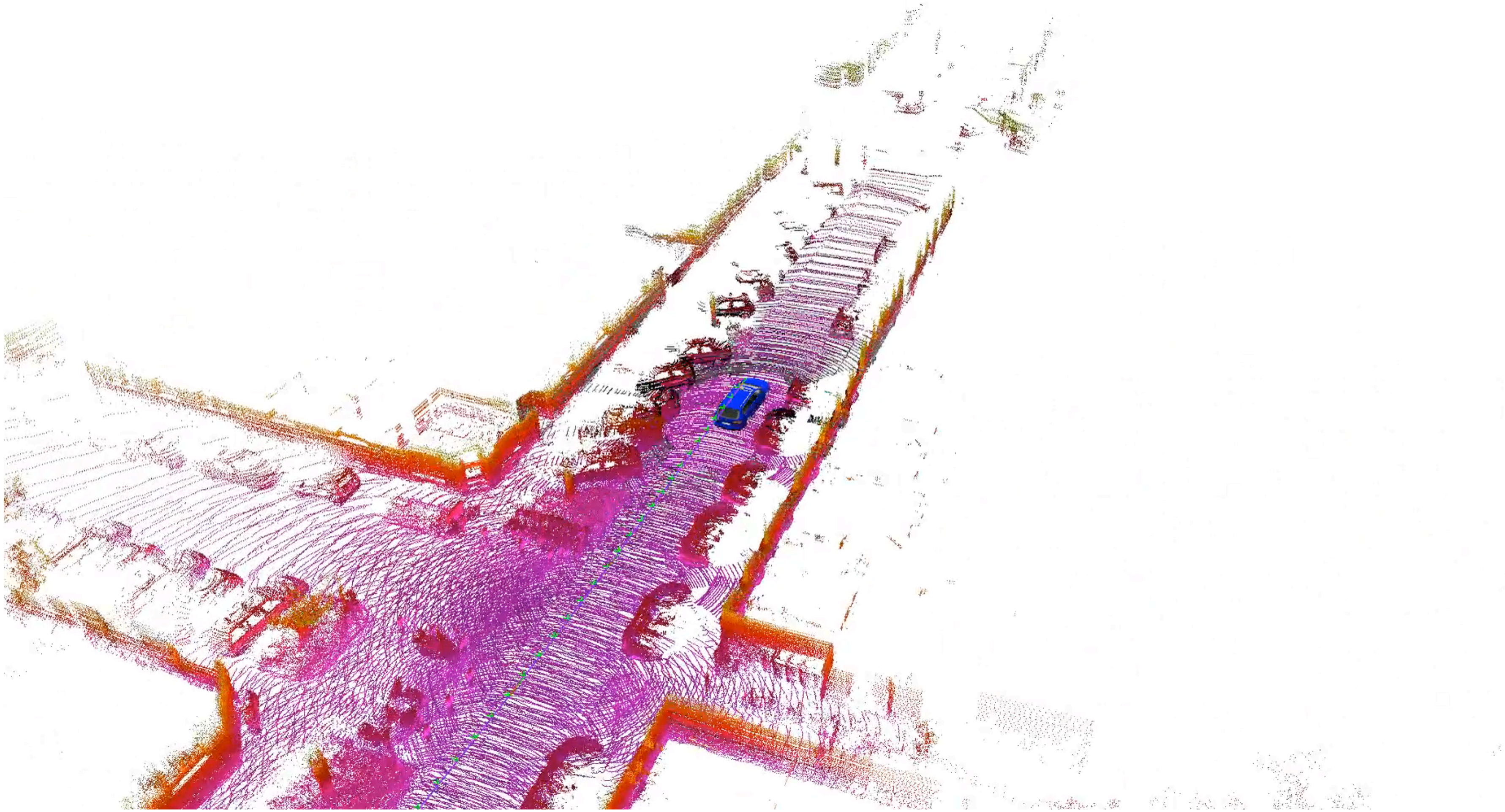


Image courtesy: Newcombe et al.

LiDAR Projective ICP in SLAM



Behley, Stachniss: “Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments”, RSS 2018

Data Association

- There a various different ways to find correspondences
- Investing into a good data association is key obtaining good results
- Exploit any initial guess
- Normal-based metrics often better than standard point-to-point metric

ICP Variants

Variants on the following stages of ICP have been proposed:

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

Weighting Correspondences

- **Weight** the corresponding point pairs
- **Noise:** Weighting based on sensor uncertainty
- Weights are easy to incorporate into the transformation computation

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

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

Weighting Correspondences

- **Noise:** Weighting based on sensor uncertainty
- Weights are especially relevant if measurement noise is varying

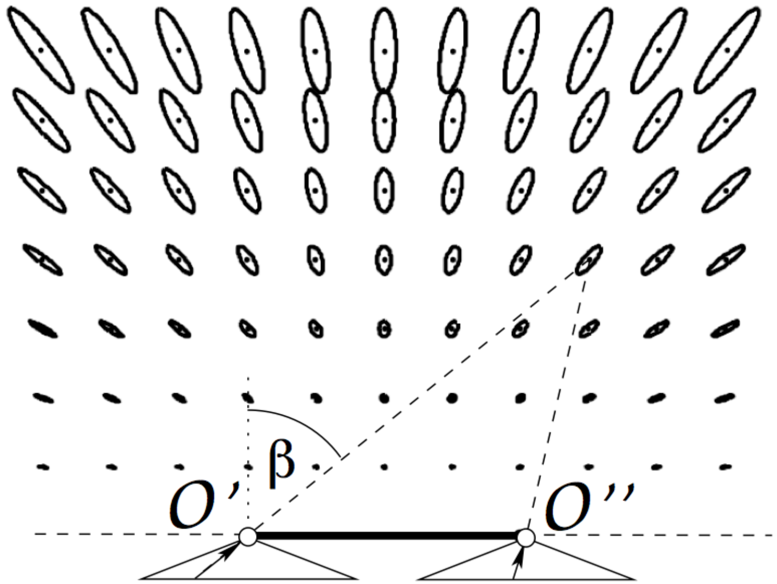


Image courtesy: Förstner and Wrobel

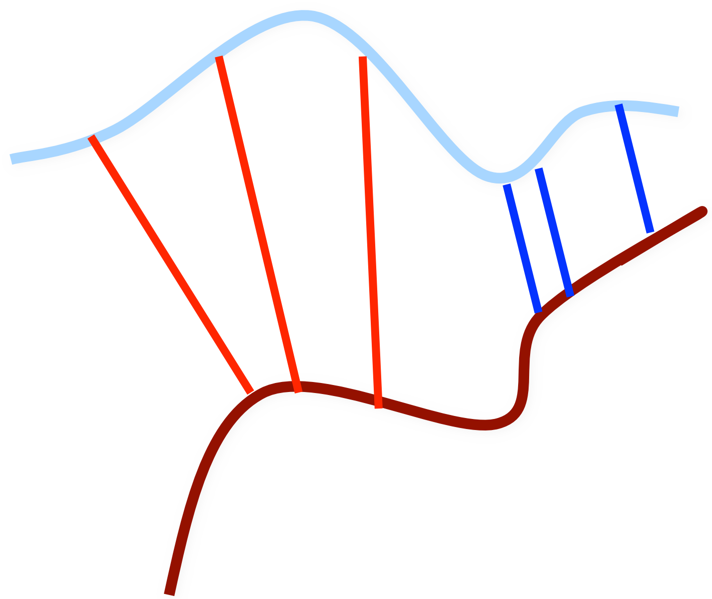
ICP Variants

Variants on the following stages of ICP have been proposed:

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

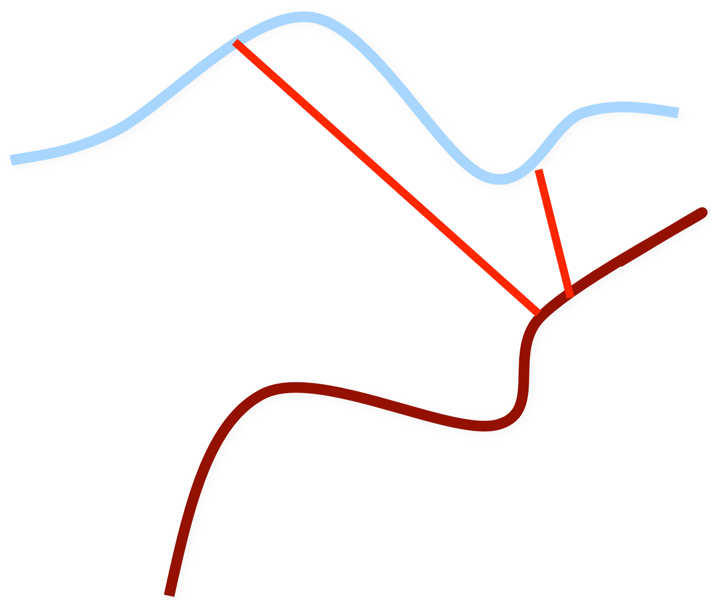
Rejecting Potential Outlier Pairs

- Point-to-point distance larger than a given threshold



Rejecting Potential Outlier Pairs

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



Rejecting Potential Outlier Pairs

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

Example: Mapping in Dynamic Environments

Palazzolo, Behley, Lottes, Giguère, Stachniss, “ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals”, IROS 2019

Mapping Works in Static Scenes Fails in Dynamic Environments

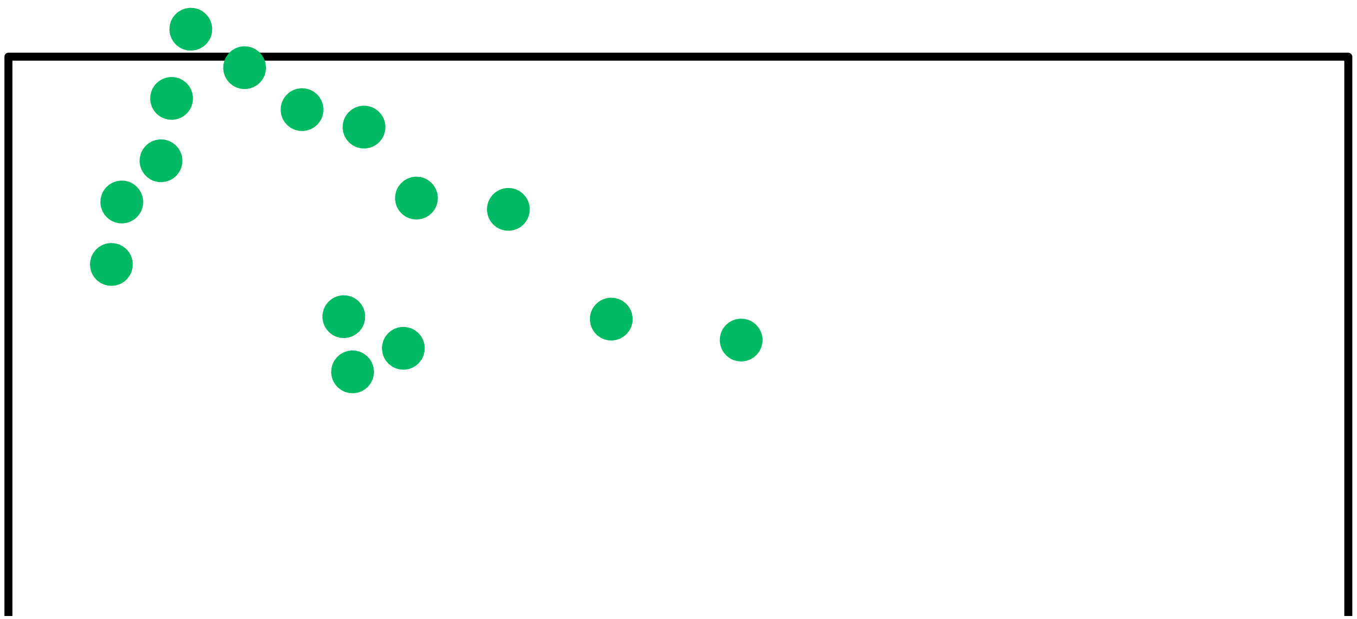
RGB camera frames

3D model re-projected onto the camera frames

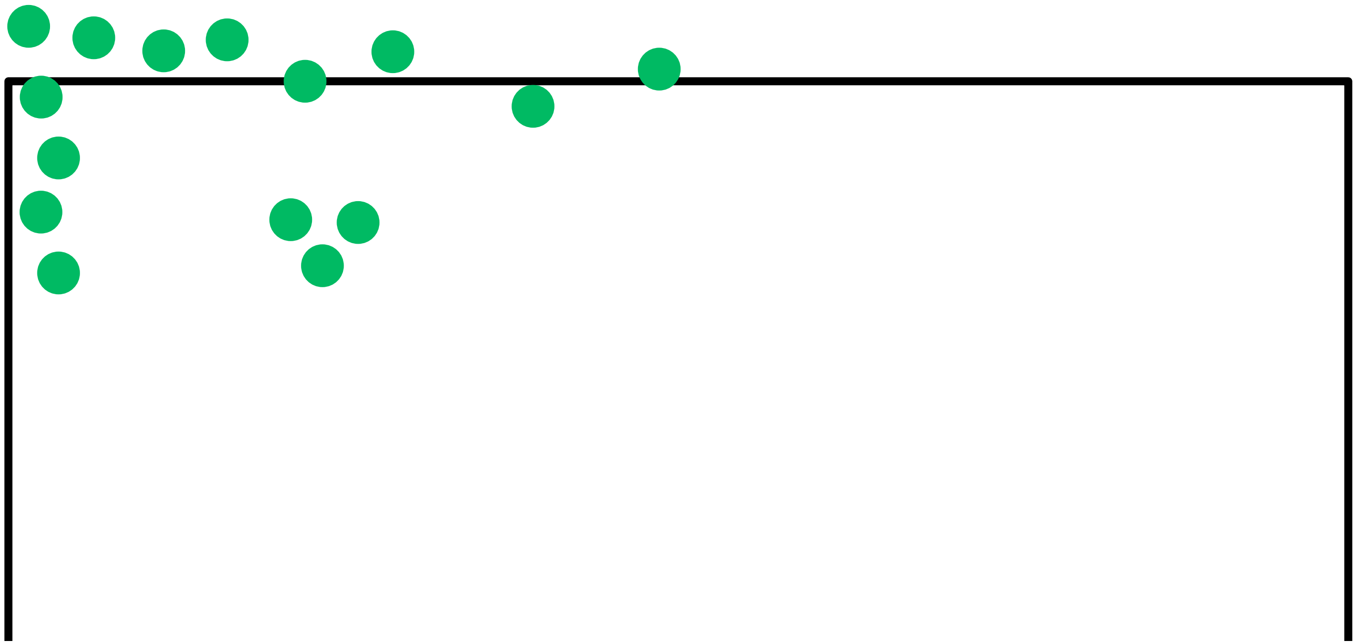
Fails in Dynamic Environments

Detection of Dynamic Elements Exploiting Residuals

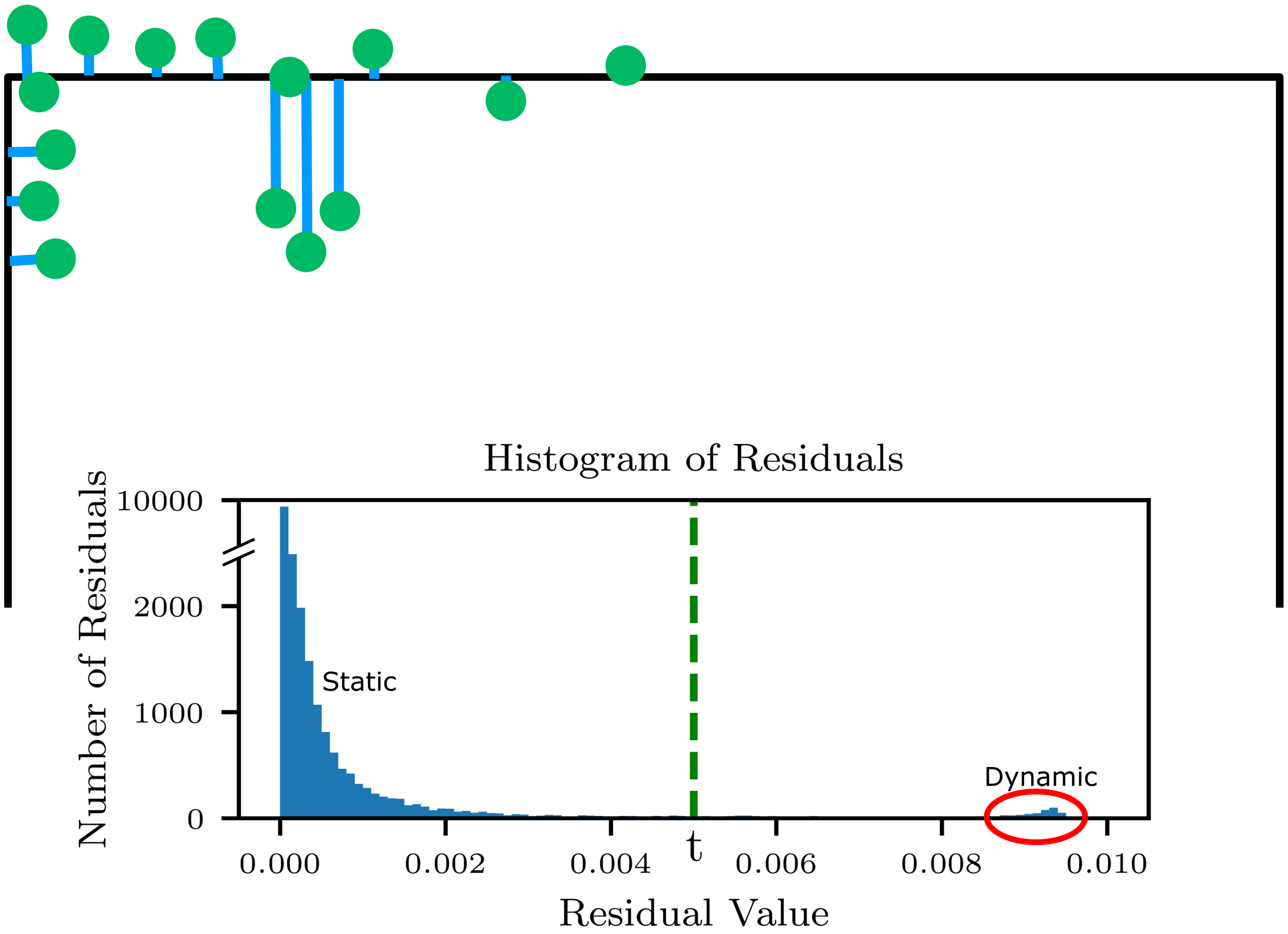
Detection of Dynamic Elements Exploiting Residuals



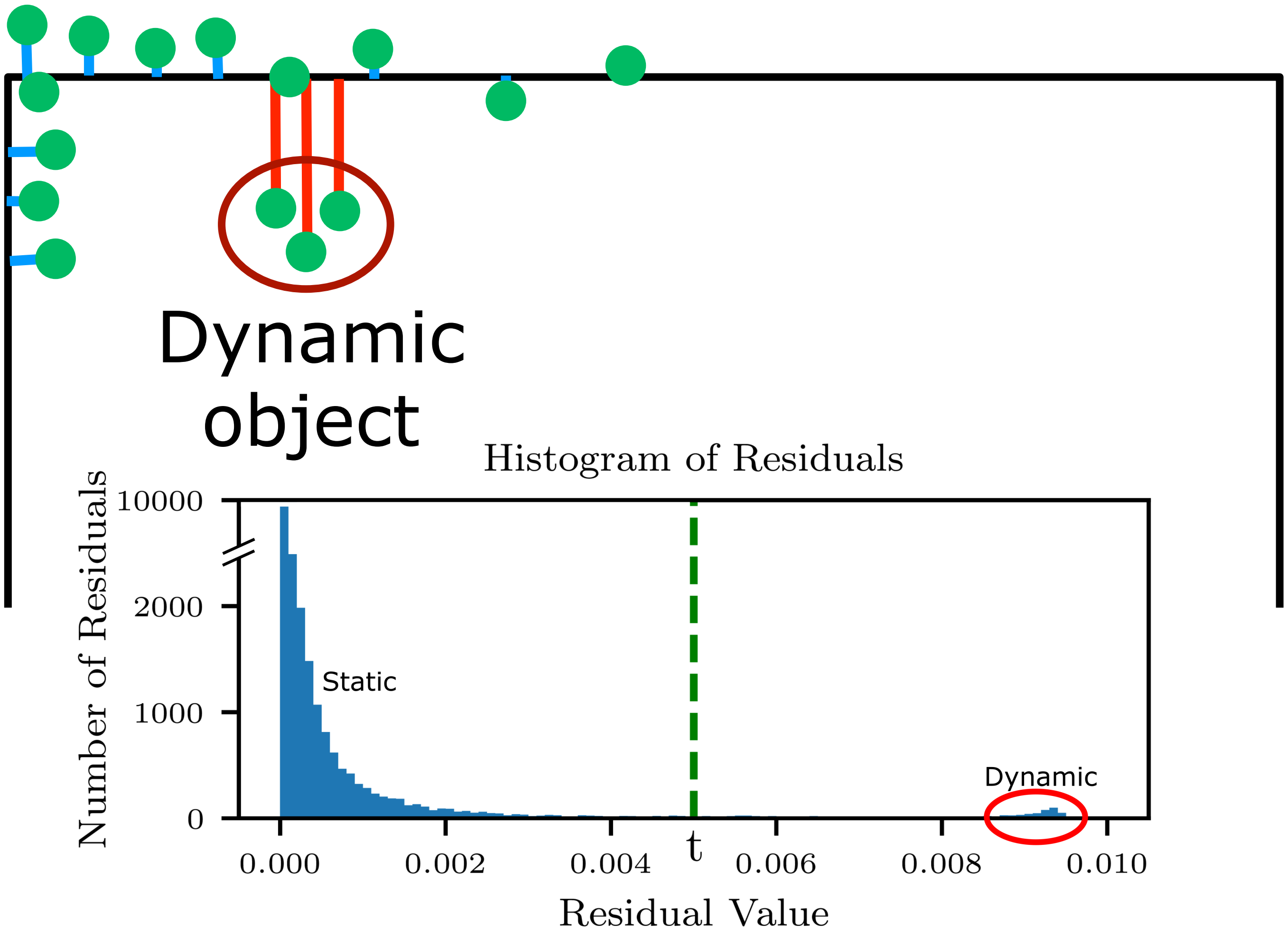
Detection of Dynamic Elements Exploiting Residuals



Detection of Dynamic Elements Exploiting Residuals

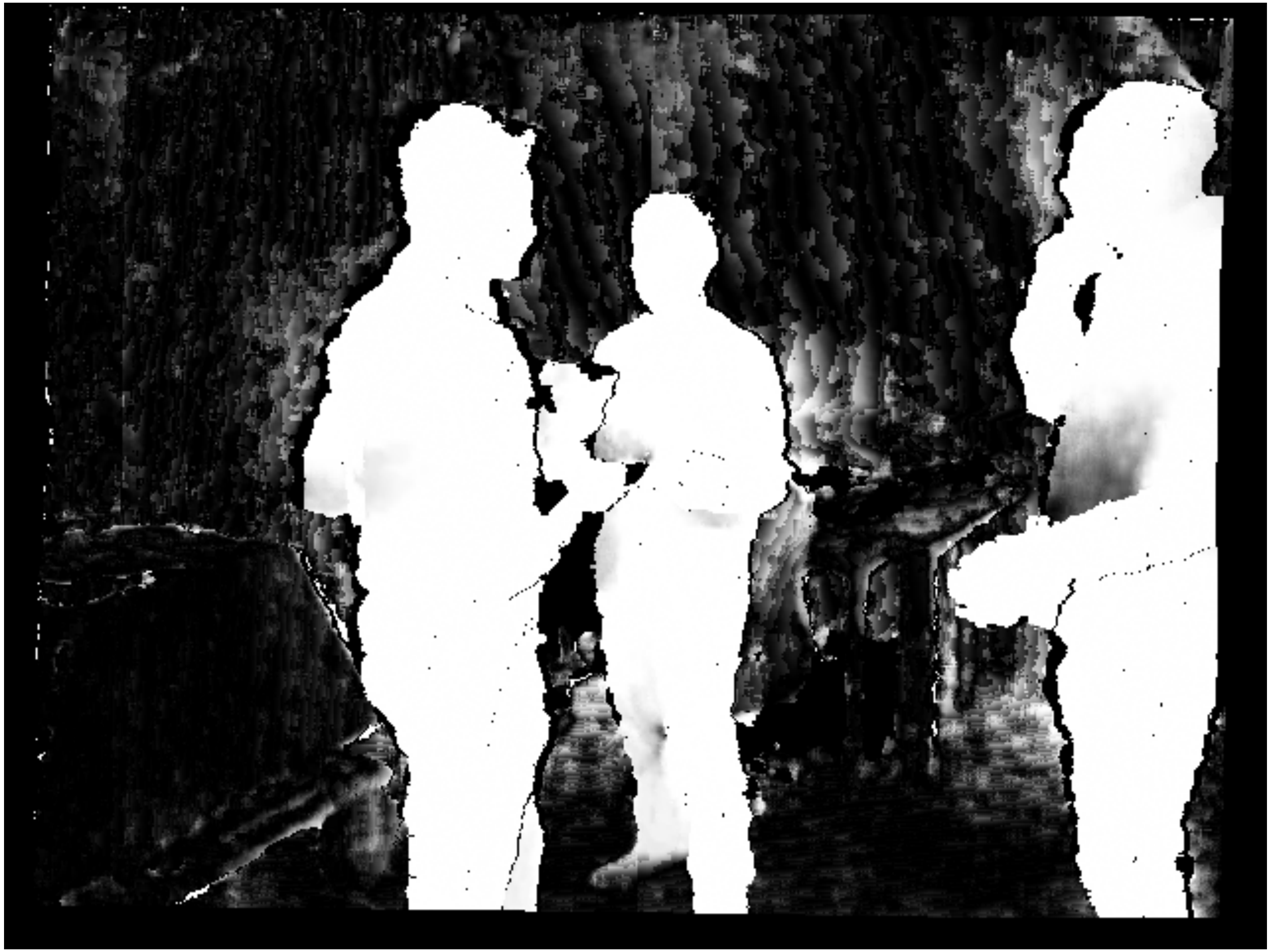


Detection of Dynamic Elements Exploiting Residuals

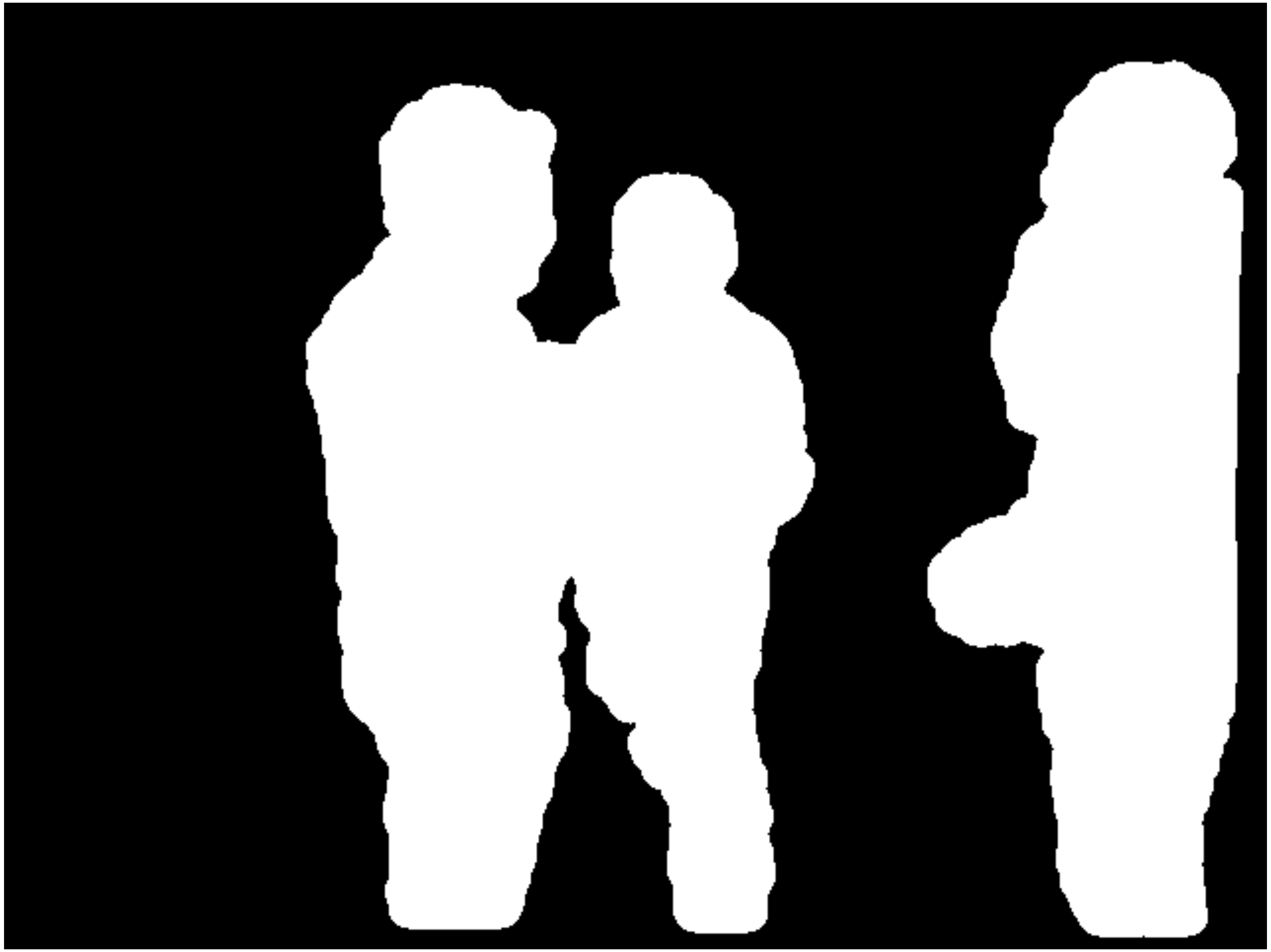


Detection of Dynamic Elements Exploiting Residuals

Detection of Dynamic Elements Exploiting Residuals



Detection of Dynamic Elements Exploiting Residuals



Works in Dynamic Environments

RGB camera frames

3D model re-projected onto the camera frames

Palazzolo, Behley, Lottes, Giguère, Stachniss, "ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals", IROS 2019

Works in Dynamic Environments

Palazzolo, Behley, Lottes, Giguère, Stachniss, “ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals”, IROS 2019

Works in Dynamic Environments

RGB camera frames

3D model re-projected onto the camera frames

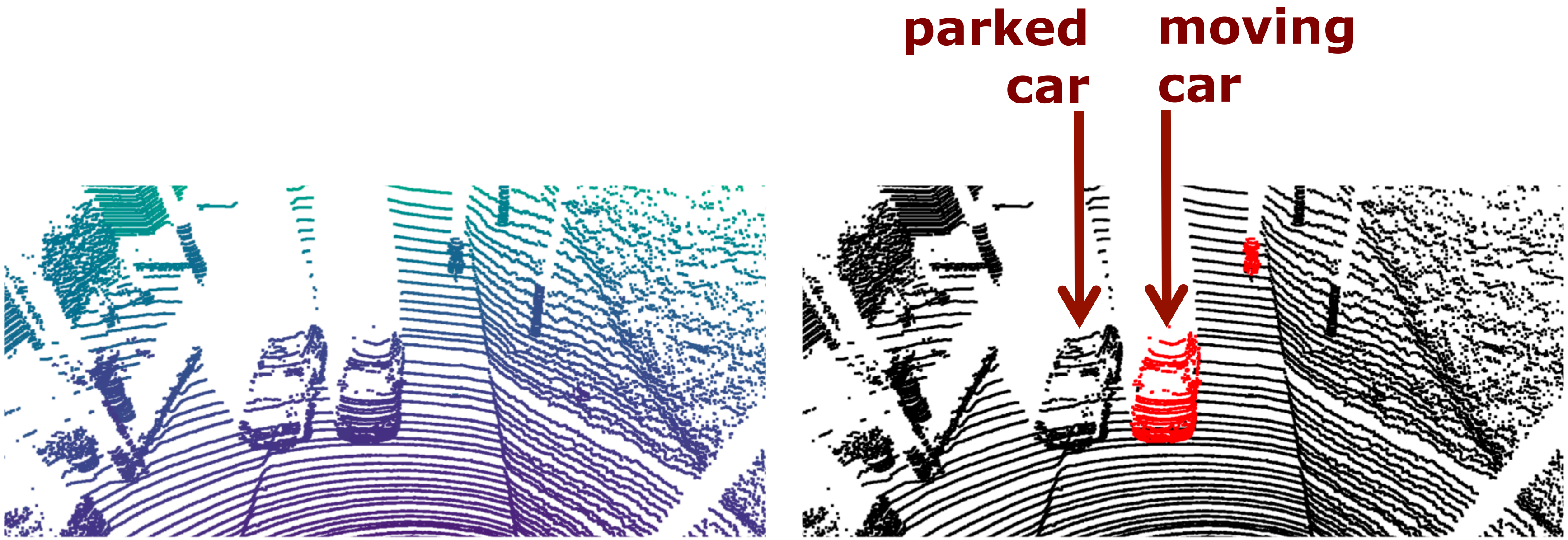
Palazzolo, Behley, Lottes, Giguère, Stachniss, "ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals", IROS 2019

Learning-based
Moving Object Detection

Chen, Li, Mersch, Wiesmann, Gall, Behley, Stachniss: "Moving Object Segmentation in 3D LiDAR Data: A Learning-based Approach Exploiting Sequential Data"

Sophisticated Outlier Rejection

Deep learning-based moving object segmentation in 3D LiDAR scans

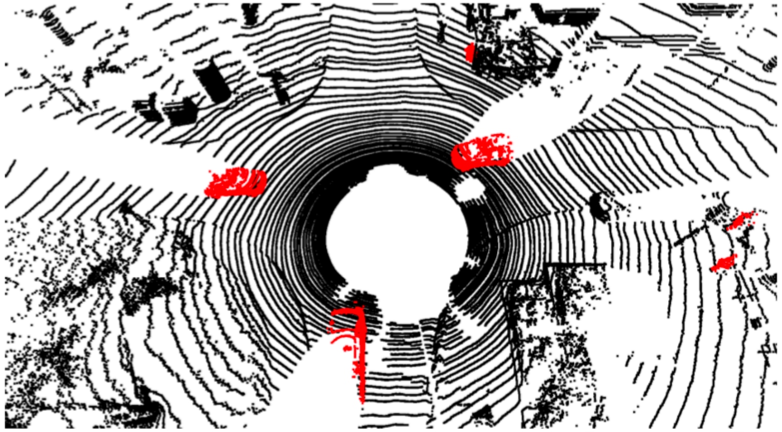


raw point cloud **moving object segmentation**

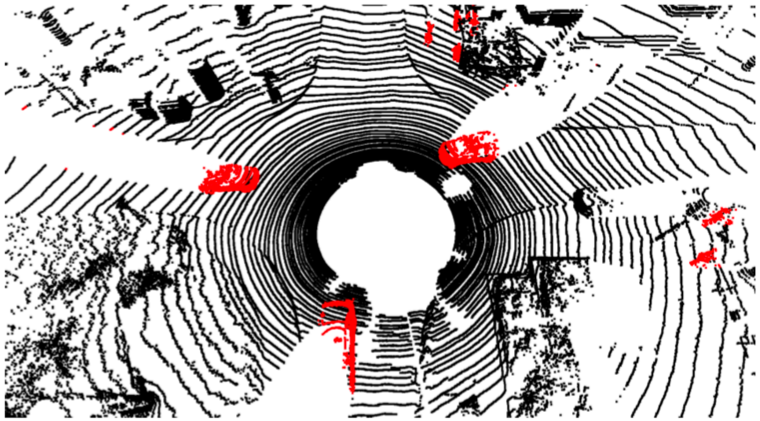
Chen, Li, Mersch, Wiesmann, Gall, Behley, Stachniss: "Moving Object Segmentation in 3D LiDAR Data: A Learning-based Approach Exploiting Sequential Data"

Sophisticated Outlier Rejection

raw point cloud



ground truth



estimated moving object segmentation

Chen, Li, Mersch, Wiesmann, Gall, Behley, Stachniss: "Moving Object Segmentation in 3D LiDAR Data: A Learning-based Approach Exploiting Sequential Data"

ICP Algorithm

- Potentially subsample point clouds
- Determine corresponding points
- Potentially weight or reject outlier pairs
- Compute rotation *R*, translation *t* (SVD)
- Apply *R* and *t* to all points of the set to be registered
- Compute the error $E(R, \boldsymbol{t})$
- While error decreased and error > threshold
  - Determine correspondences and weights
  - Compute and apply rigid body transformation
- Output final alignment

Outlook

Part 3:
Point Cloud Registration using Non-Linear Least Squares

Summary

- Registration of point clouds is an important task in perception
- ICP is the standard algorithm for point cloud alignment/scan matching
- Estimates translation and rotation between clouds/scans
- Given data associations between clouds, the transformation can be computed efficiently

Summary

- **The major problem is to determine the correct data associations**
- Iterative approach (DA & alignment)
- Several variants exist
- Initial guess is needed for robust data association
- **Often:** least squares approach with a plane-based metric, data association heuristics, and outlier rejection

5 Minute Summary...

https://www.youtube.com/watch?v=QWDM4cFdKrE