

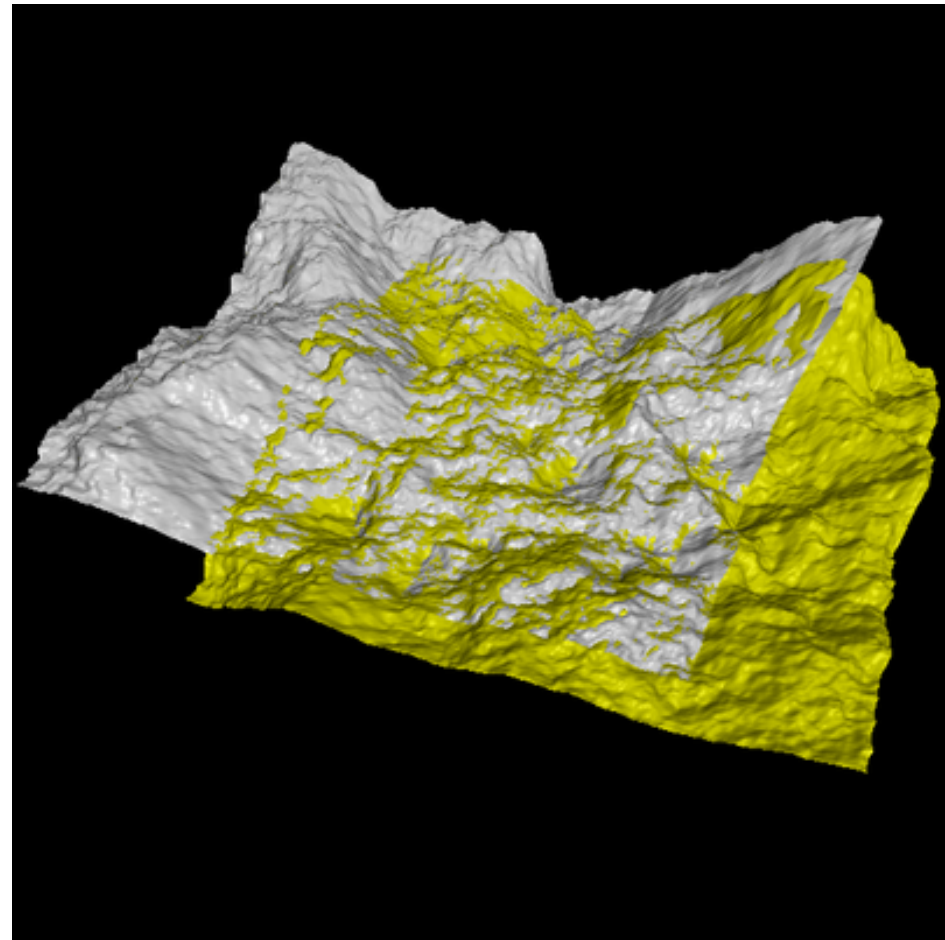
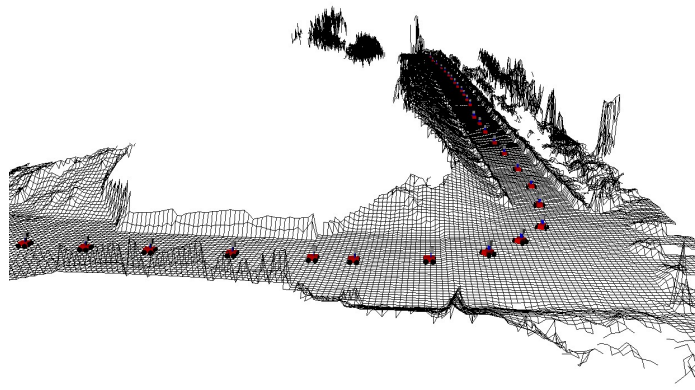
Introduction to Mobile Robotics

Iterative Closest Point Algorithm

Wolfram Burgard, Cyrill Stachniss, Maren
Bennewitz, Diego Tipaldi, Luciano Spinello



Motivation



Goal: Find local transformation to align points

The Problem

- Given two corresponding point sets:

$$X = \{x_1, \dots, x_{N_x}\}$$

$$P = \{p_1, \dots, p_{N_p}\}$$

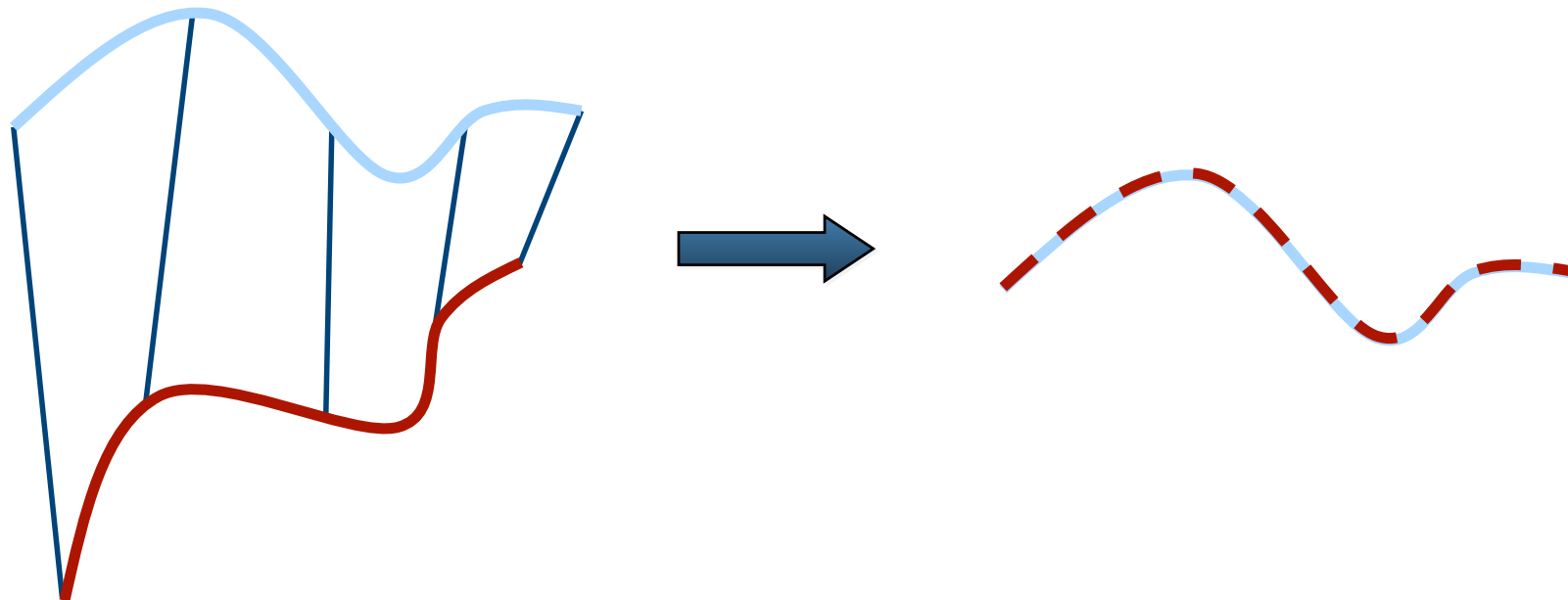
- Wanted: Translation t and rotation R that minimize the sum of the squared errors:

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

Here, x_i and p_i are corresponding points

Key Idea

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



Center of Mass

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

are the centers of mass of the two point sets

Idea:

- Subtract the corresponding center of mass from every point in the two point sets before calculating the transformation
- The resulting point sets are:

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

Singular Value Decomposition

$$\text{Let } W = \sum_{i=1}^{N_p} x_i' p_i'^T$$

denote 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 $U, V \in \mathbb{R}^{3 \times 3}$ are unitary, and

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

SVD

Theorem (without proof):

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

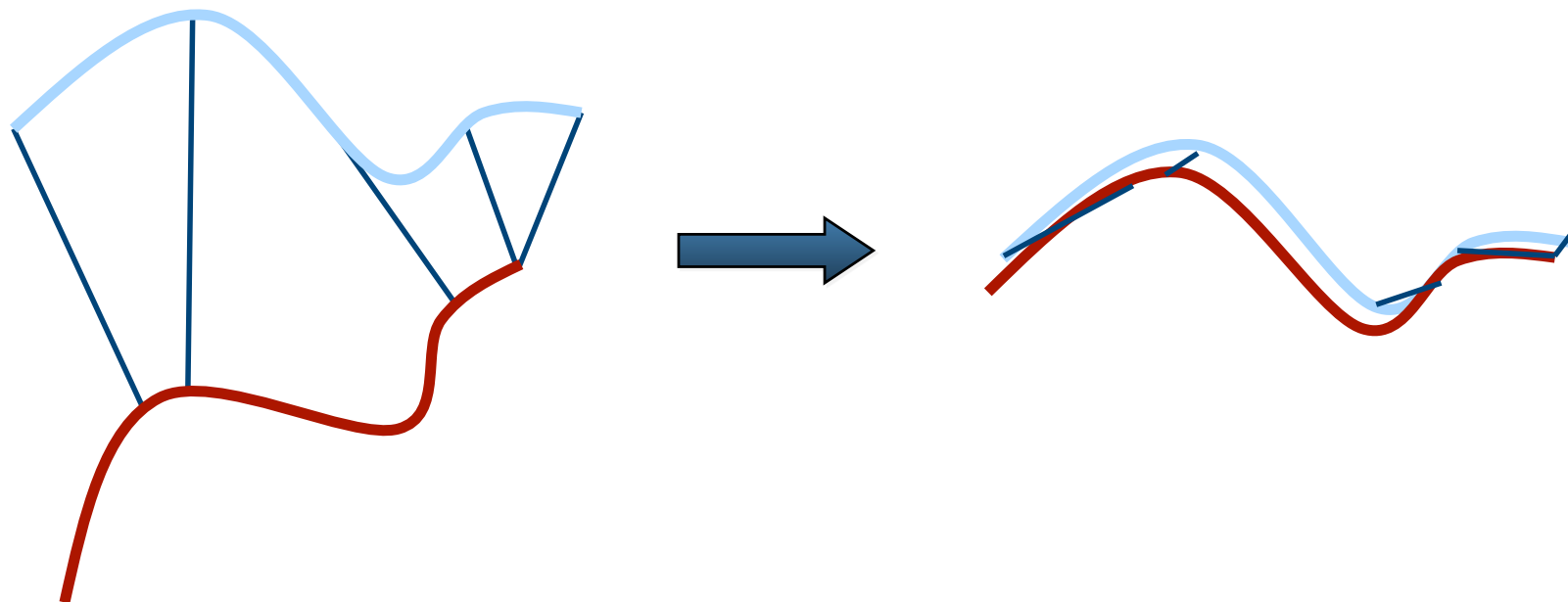
$$R = UV^T$$
$$t = \mu_x - R\mu_p$$

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

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

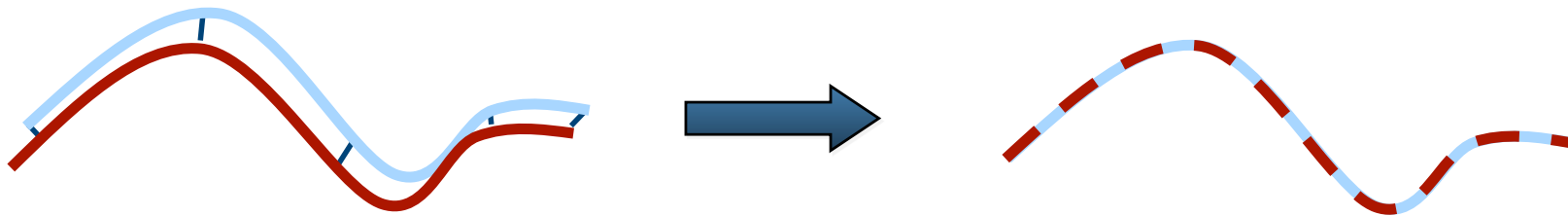
ICP with Unknown Data Association

- If the correct correspondences are **not known**, it is generally impossible to determine the optimal relative rotation and translation in one step



Iterative Closest Point (ICP) Algorithm

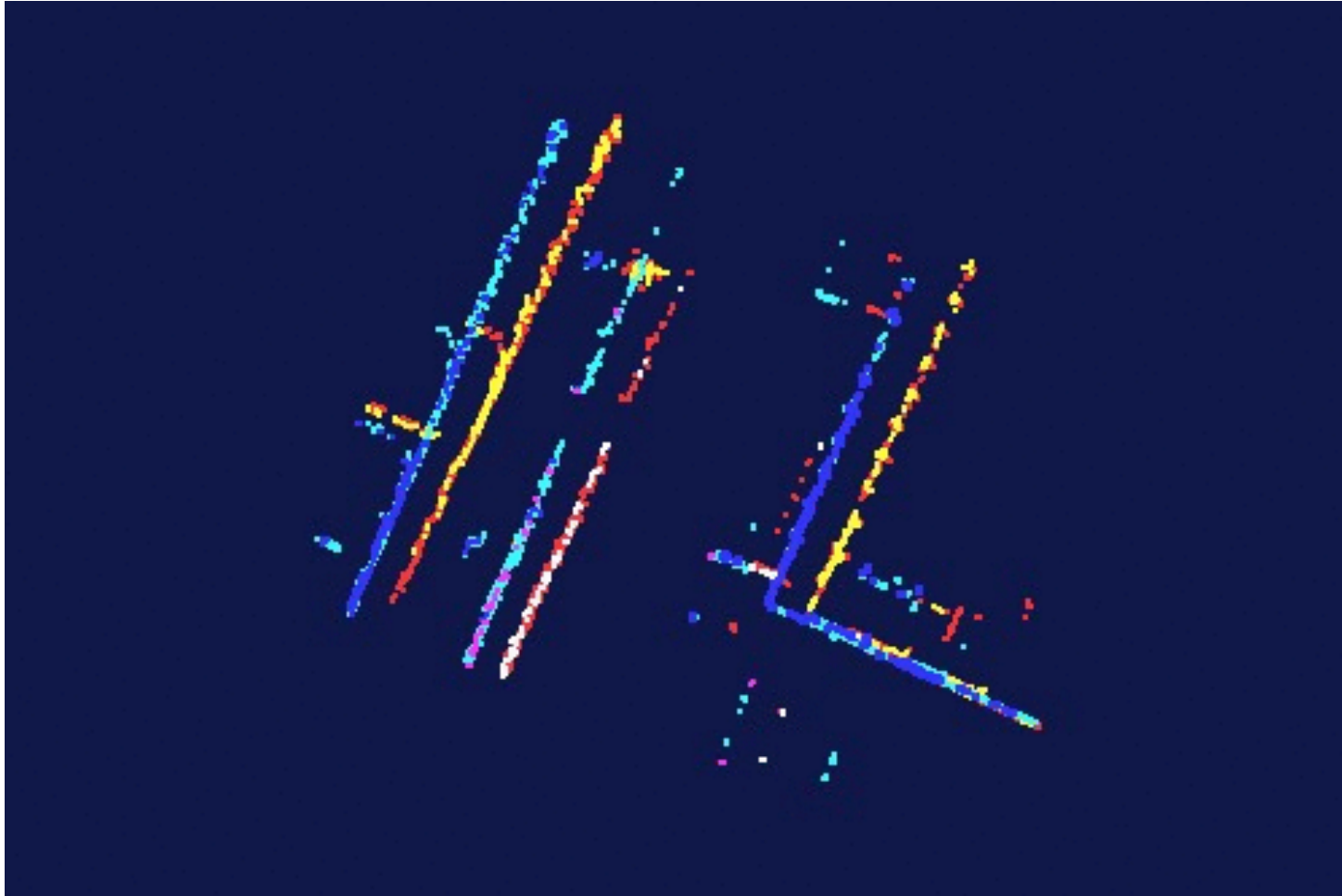
- Idea: Iterate to find alignment
- Iterative Closest Points
[Besl & McKay 92]
- Converges if starting positions are “close enough”



Basic ICP Algorithm

- Determine corresponding points
- 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

ICP Example



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
4. Rejecting certain (outlier) point pairs

Performance of Variants

- Various aspects of performance:
 - Speed
 - Stability (local minima)
 - Tolerance wrt. noise and outliers
 - Basin of convergence
(maximum initial misalignment)

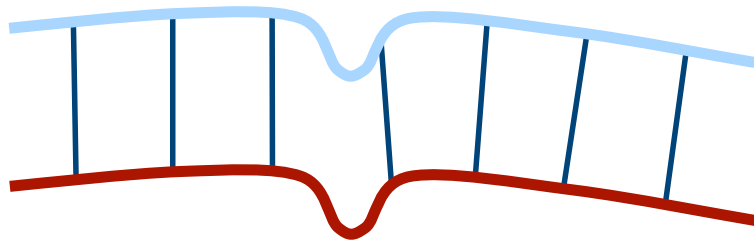
ICP Variants

- 
1. Point subsets (from one or both point sets)
 2. Weighting the correspondences
 3. Data association
 4. Rejecting certain (outlier) point pairs

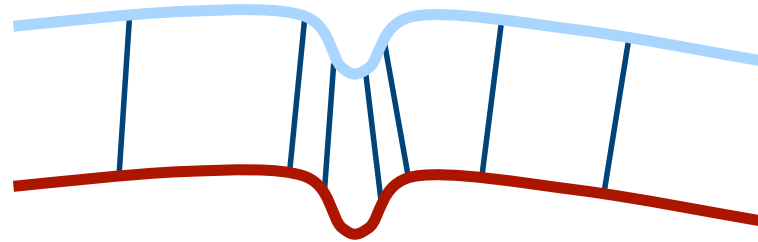
Selecting Source Points

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

Normal-Space Sampling



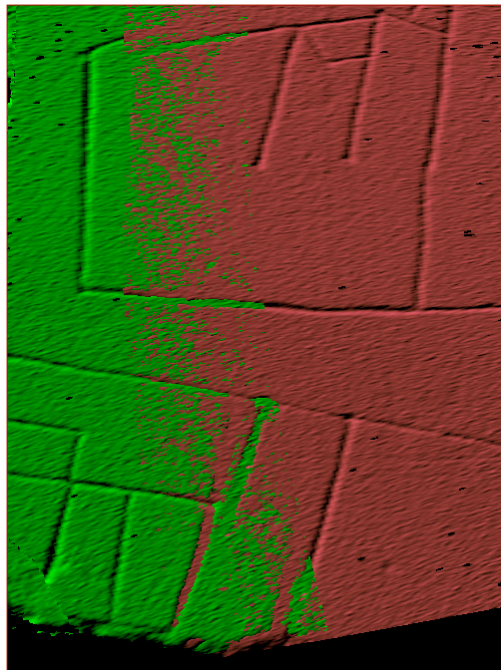
uniform sampling



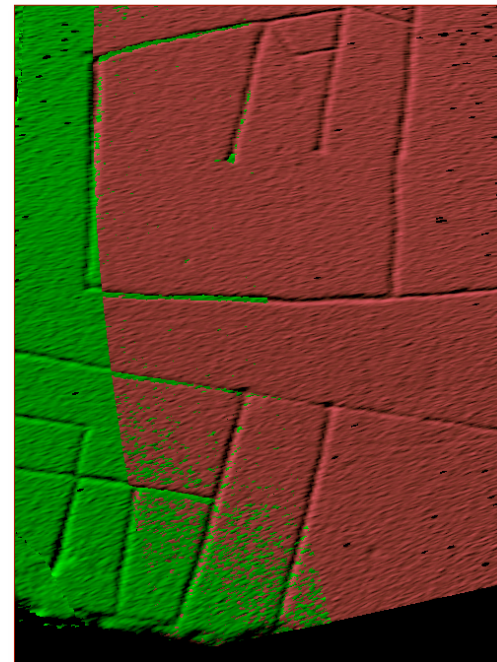
normal-space sampling

Comparison

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



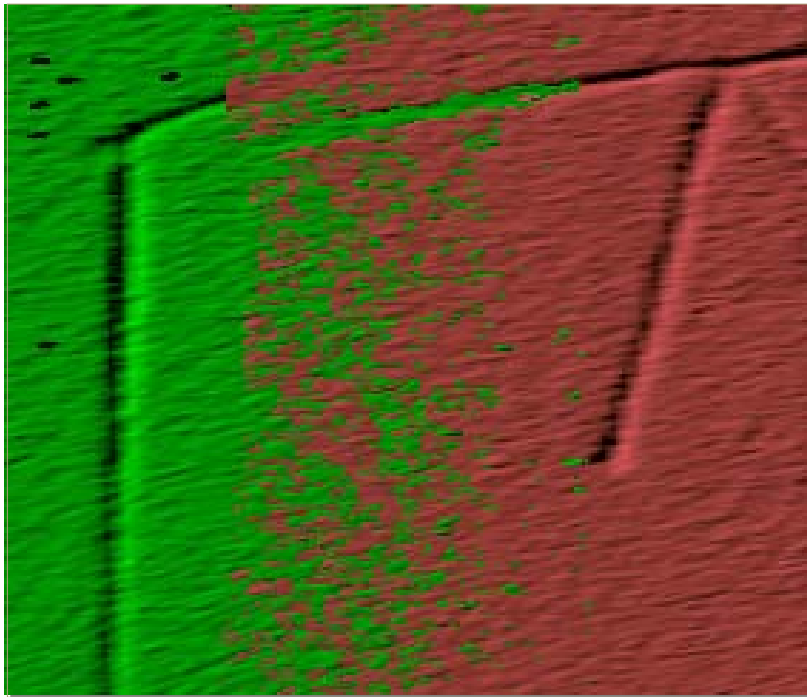
Random sampling



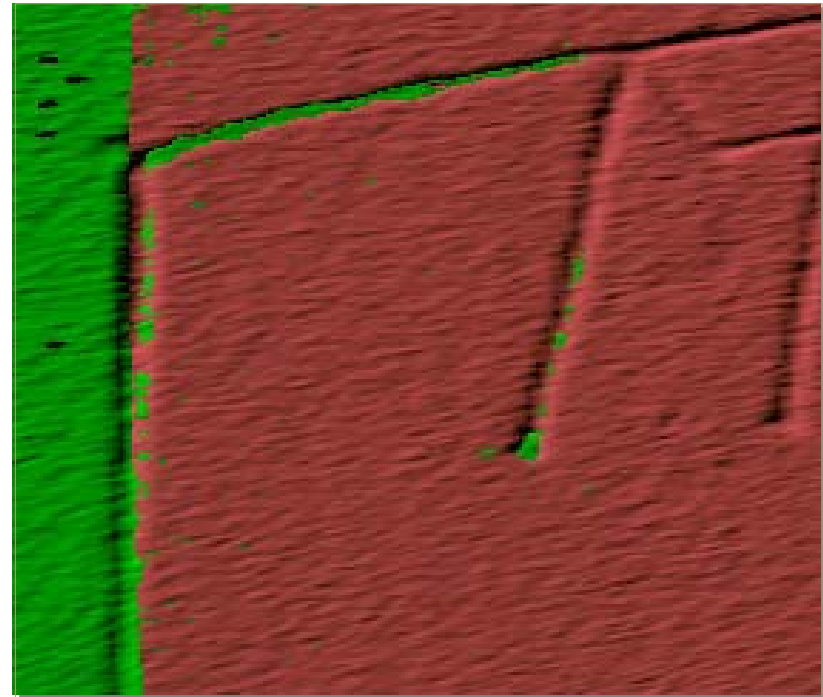
Normal-space sampling

Comparison

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



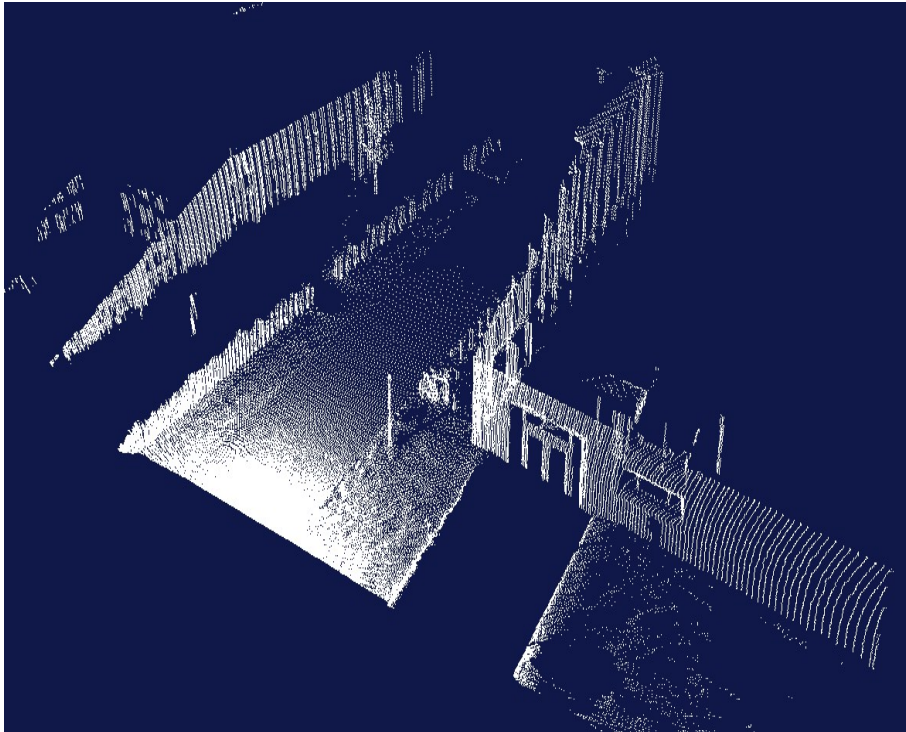
Random sampling



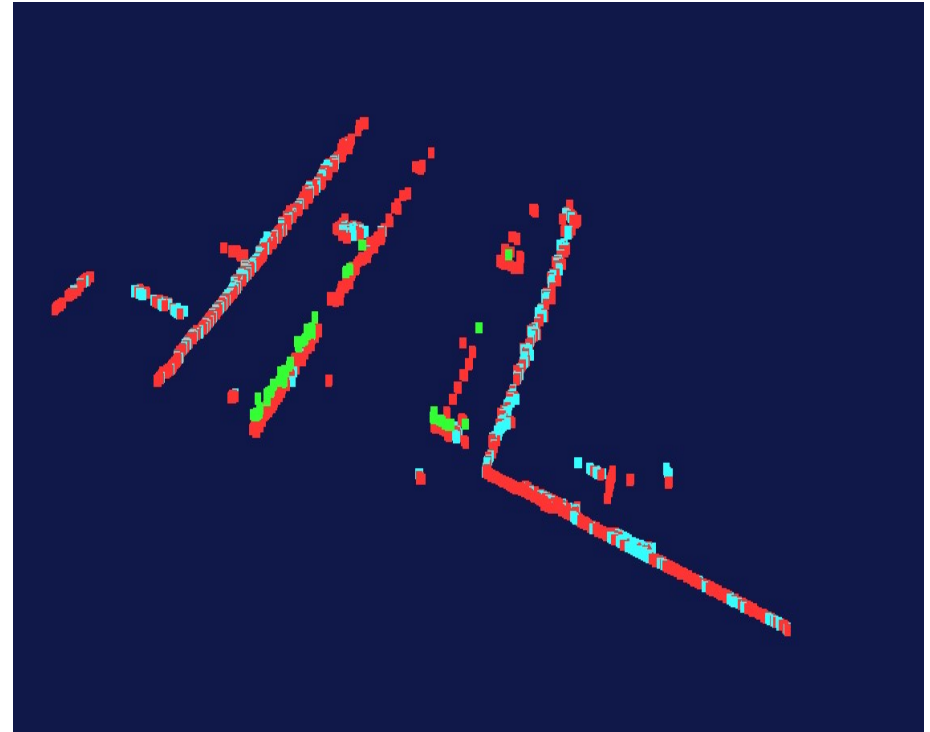
Normal-space sampling

Feature-Based Sampling

- Try to find “important” points
- Decreases the number of correspondences to find
- Higher efficiency and higher accuracy
- Requires preprocessing

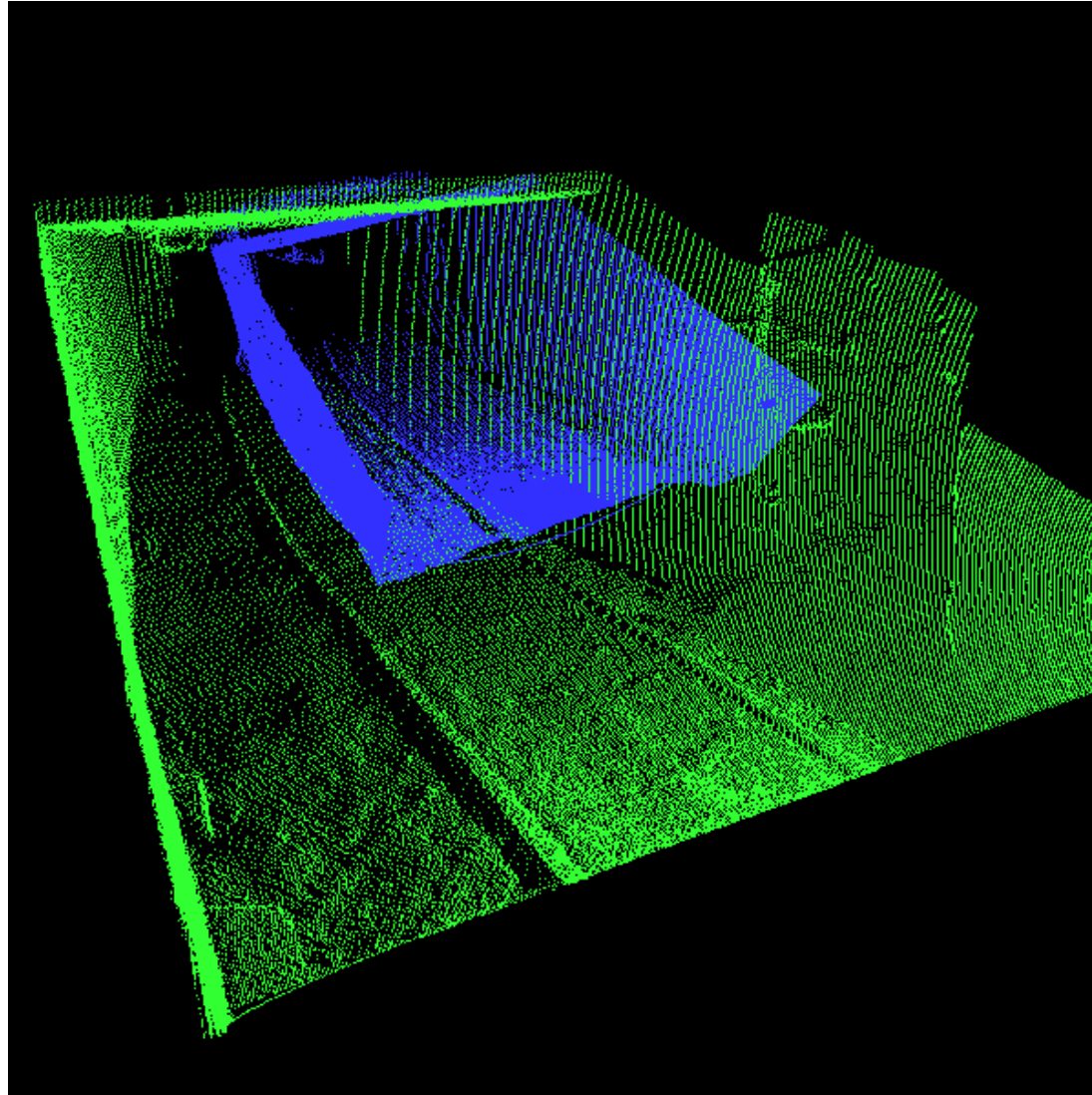


3D Scan (~200.000 Points)




Extracted Features (~5.000 Points)

ICP Application (With Uniform Sampling)



[Nuechter et al., 04]

ICP Variants

1. Point subsets (from one or both point sets)
-  2. **Weighting the correspondences**
3. Data association
4. Rejecting certain (outlier) point pairs


Weighting

- Select a set of points for each set
- Match the selected points of the two sets
- **Weight the corresponding pairs**
- E.g., assign lower weights for points with higher point-point distances
- Determine transformation that minimizes the error function

Selection vs. Weighting

- Could achieve same effect with weighting
- Hard to guarantee that enough samples of important features except at high sampling rates
- Weighting strategies turned out to be dependent on the data
- Preprocessing / run-time cost tradeoff (how to find the correct weights?)

ICP Variants

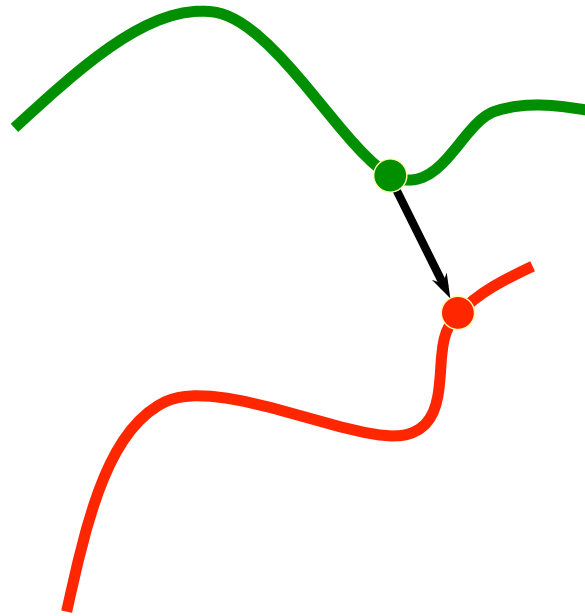
1. Point subsets (from one or both point sets)
2. Weighting the correspondences
-  3. **Data association**
4. Rejecting certain (outlier) point pairs

Data Association

- Has greatest effect on convergence and speed
- Matching methods:
 - Closest point
 - Normal shooting
 - Closest compatible point
 - Projection-based

Closest-Point Matching

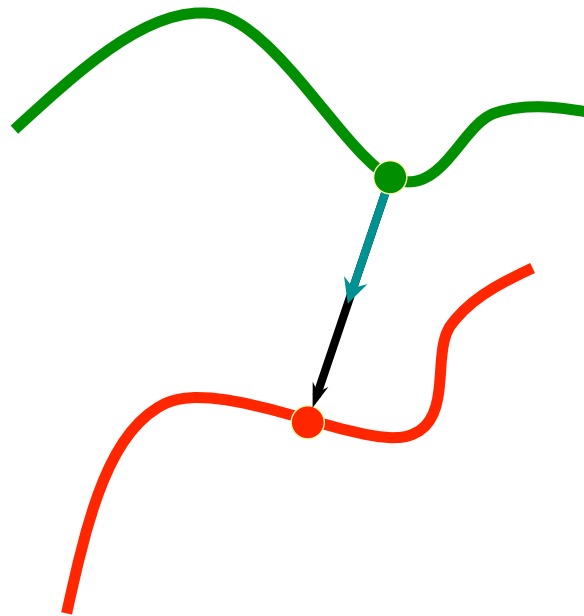
- Find closest point in other the point set (using kd-trees)



Generally stable, but slow convergence and requires preprocessing

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

Closest Compatible Point

- Improves the two previous variants by considering the **compatibility** of the points
- Only match compatible points
- Compatibility can be based on
 - Normals
 - Colors
 - Curvature
 - Higher-order derivatives
 - Other local features

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]

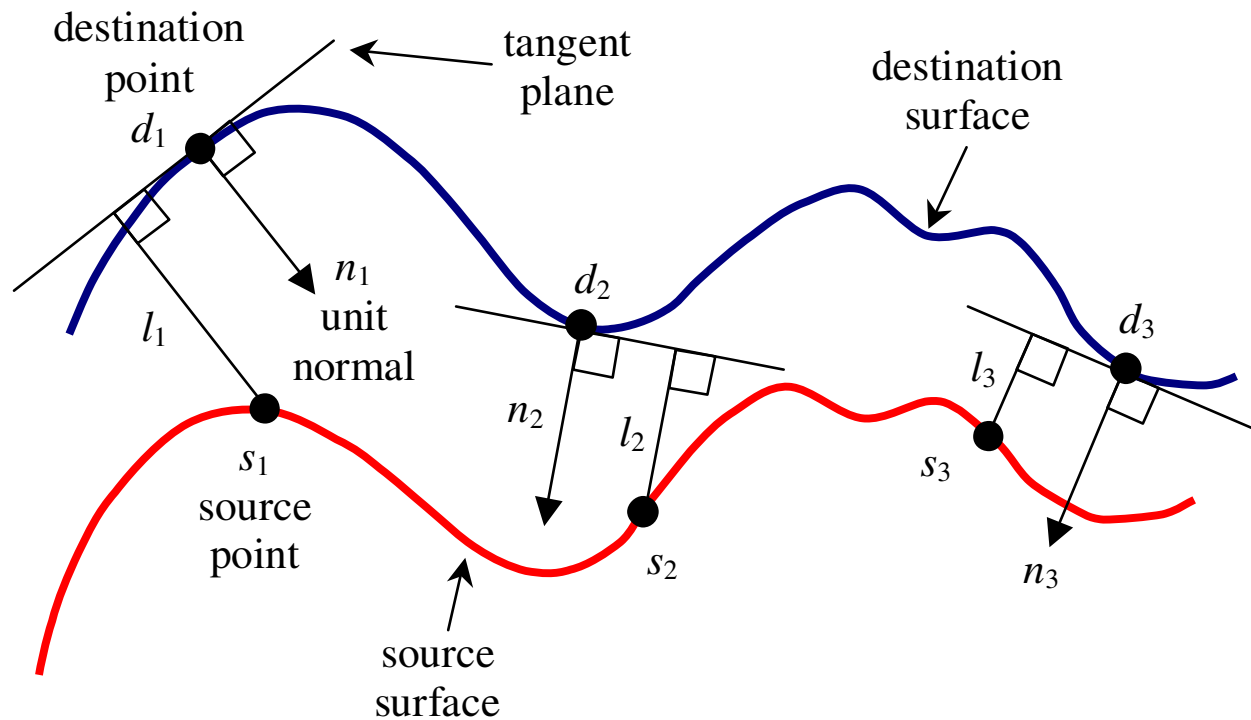


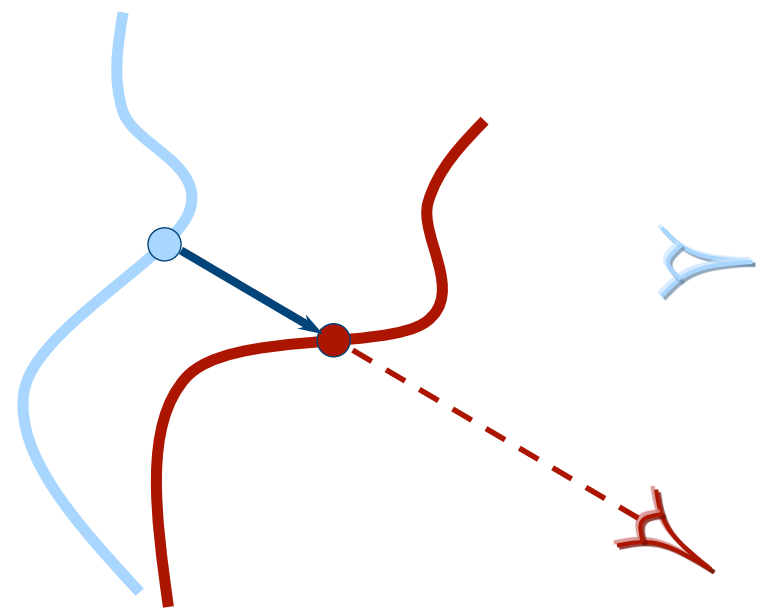
image from [Low 04]

Point-to-Plane Error Metric

- Solved using standard nonlinear least squares methods (e.g., Levenberg-Marquardt method [Press92]).
- 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]

Projection

- Finding the closest point is the most expensive stage of the ICP algorithm
- Idea: Simplified nearest neighbor search
- For range images, one can project the points according to the view-point [Blais 95]



Projection-Based Matching

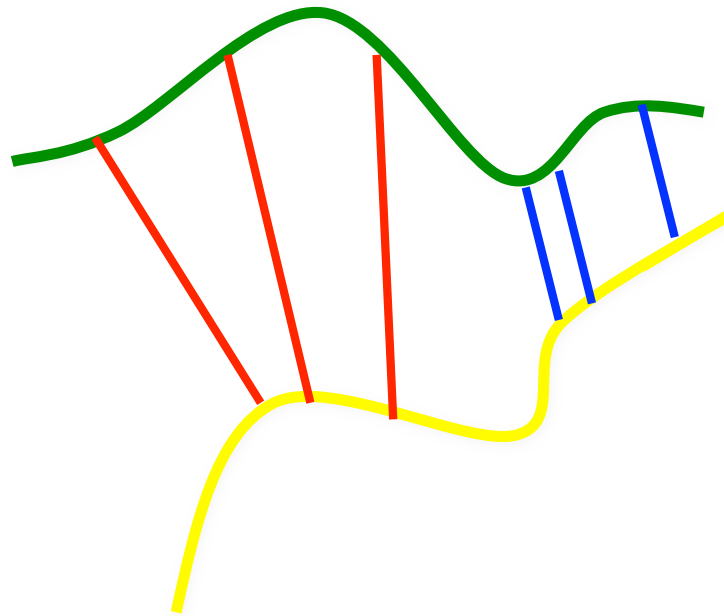
- Constant time
- Does not require pre-computing a special data structure
- Requires point-to-plane error metric
- Slightly worse alignments per iteration

ICP Variants

1. Point subsets (from one or both point sets)
2. Weighting the correspondences
3. Data association
- ➔ 4. Rejecting certain (outlier) point pairs

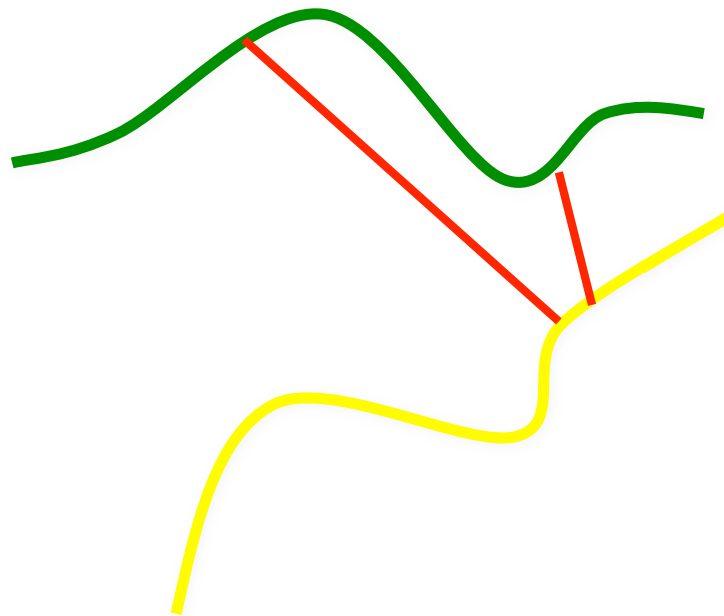
Rejecting (Outlier) Point Pairs

- Corresponding points with point to point distance higher than a given threshold



Rejecting (Outlier) Point Pairs

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



Rejecting (Outlier) Point Pairs

- Corresponding points with point to point distance higher than a given threshold
- Rejection of pairs that are not consistent with their neighboring pairs [Dorai 98]
- Sort all correspondences with respect to their error and delete the worst $t\%$,
Trimmed ICP (TrICP) [Chetverikov et al. 02]
 - t is used to estimate the overlap
 - Problem: Knowledge about the overlap is necessary or has to be estimated

Rejecting (Outlier) Point Pairs

- Sort all correspondences with respect to their error and delete the worst $t\%$,
Trimmed ICP (TrICP) [Chetverikov et al. 2002]
- t is used to estimate the overlap



Problem: Knowledge about the overlap is necessary or has to be estimated

Summary: ICP Algorithm

- Potentially sample Points
- Determine corresponding points
- Potentially weight / reject pairs
- Compute rotation R , translation t (e.g. SVD)
- Apply R and t to all points of the set to be registered
- Compute the error $E(R,t)$
- If error decreased and error $>$ threshold
 - Repeat to determine correspondences etc.
 - Stop and output final alignment, otherwise

ICP Summary

- ICP is a powerful algorithm for calculating the displacement between scans
- The major problem is to determine the correct data associations
- Convergence speed depends on point matched points
- Given the correct data associations, the transformation can be computed efficiently using SVD
- ICP does not always converge