Target-less registration of point cloud: A Review

Yue Pan

Outline

- Introduction

- Determine Correspondences
  - Iterative Closest Points (ICP)
  - Registration based on (man-crafted and learned) feature matching
  - Randomized hypothesize-and-verify

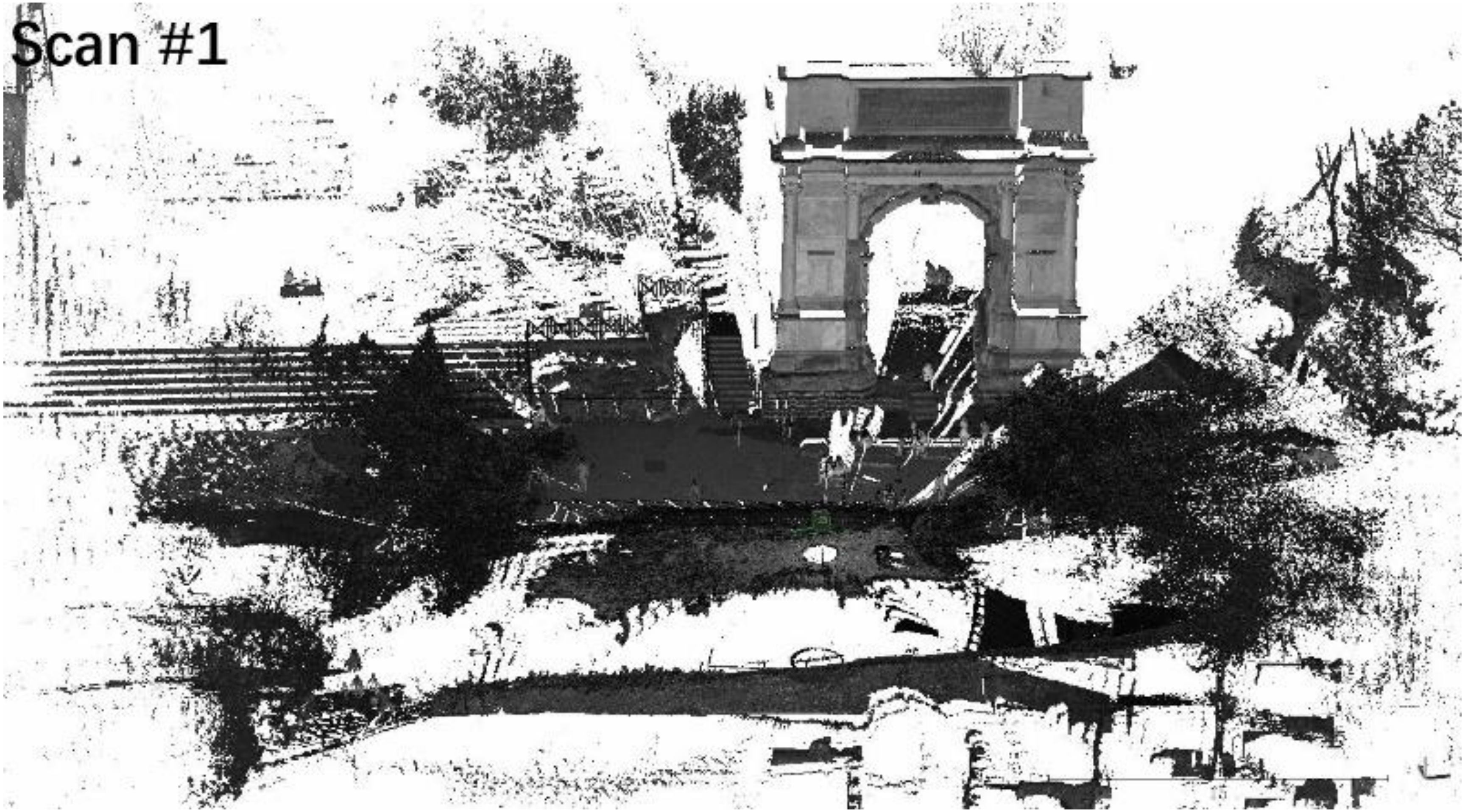
- Estimate Transformation
  - SVD (point correspondences)
  - 3 Planes + 1 Intersection (plane correspondences)

- Summary

Slides and reference links:
https://github.com/YuePanEdward/point-cloud-registration-review

Introduction

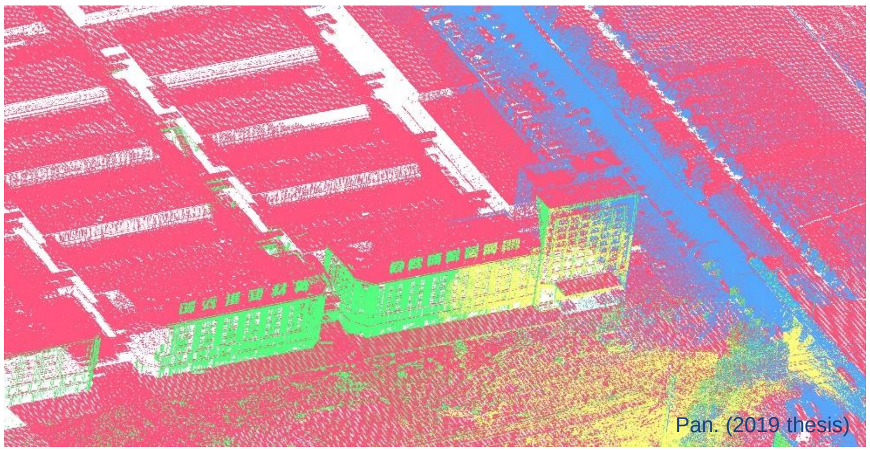
What is point cloud registration?



Application of registration

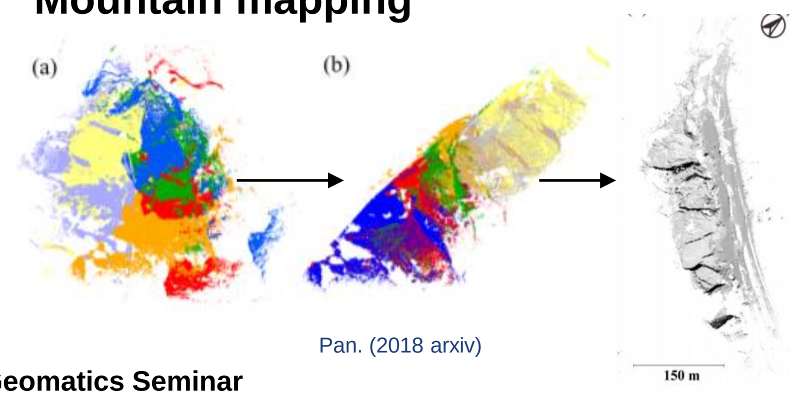
- Geomatics: Unify coordinate system of scans
- Simultaneous Localization and Mapping (SLAM)

Multi-source point cloud fusion



Pan. (2019 thesis)

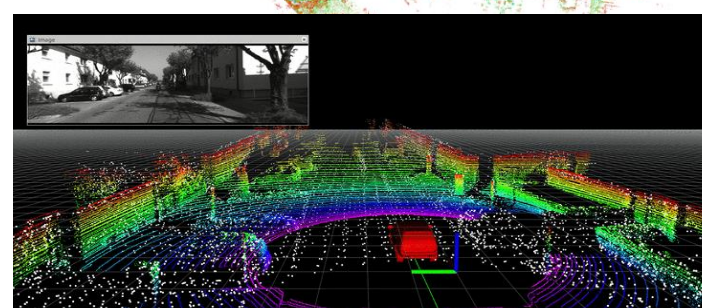
Mountain mapping



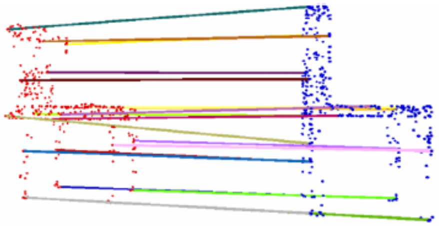
Pan. (2018 arxiv)

Lidar Odometry and Mapping

https://github.com/HKUST-Aerial-Robotics/A-LOAM



General Solution



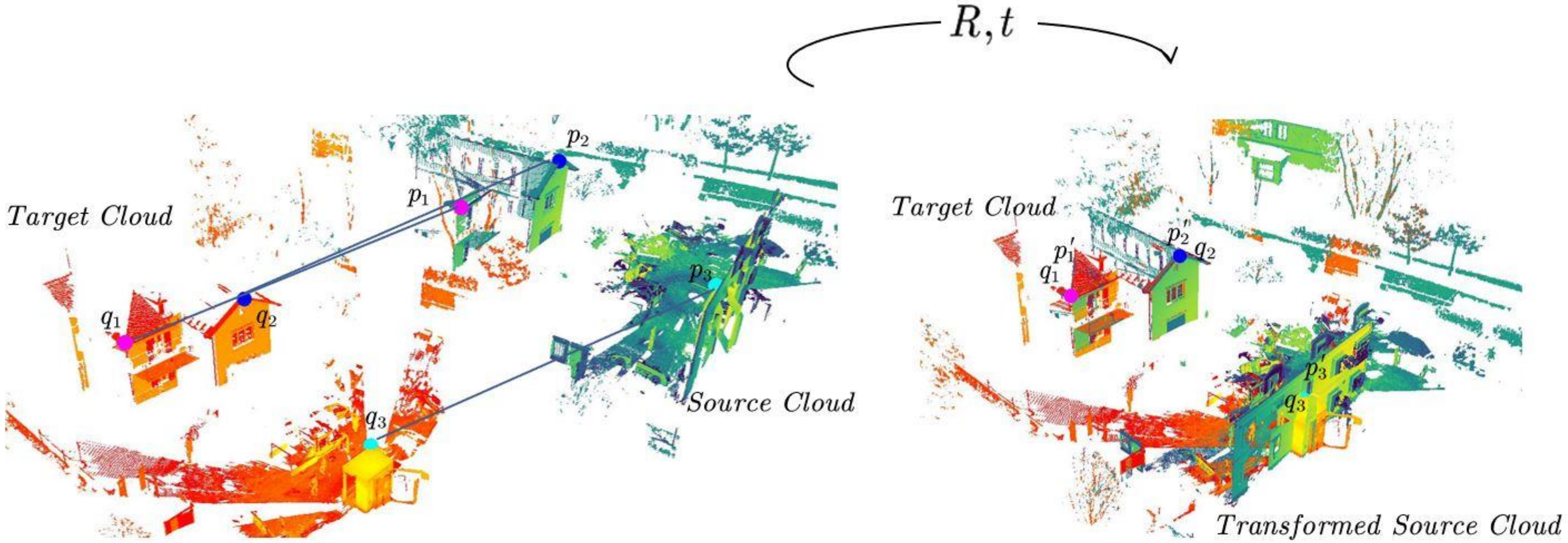
- **First Step**: Determine Correspondences
  - Corresponding points / lines / planes / objects…
  - Prepare: Detect key points, Fit lines / planes, Recognize objects

- **Second Step:** Estimate Transformation
  - Define target function
  - Minimize (optimize) target function
  - By SVD, Linear least square, Non-linear optimization (LM, GN, GA…)

- If two steps are both solved accurately
- $\longrightarrow$ successful registration

Transformation Estimation
for Point Correspondences

- Given point correspondences $\{p_i, q_i\}$ (by manual target or Step 1)
- Target function: $\{R^*, t^*\} = \underset{\{R,t\}}{\operatorname{argmin}} \left(\sum_i \| Rp_i + t - q_i \|^2 \right)$
- How to solve?
  - **<u>SVD</u>**, <u>Linear Least Square</u>, Non-linear Optimization



ETH zürich

Transformation Estimation

SVD

Horn (1987 JOSA)

Correspondences known
$\{p_i, q_i\}$

$$\{R^*, t^*\} = \underset{\{R,t\}}{\operatorname{argmin}} \left(\sum_i \| Rp_i + t - q_i \|^2 \right)$$

1. Calculate the centroids of two point clouds

$$\overline{p} = \frac{1}{N} \sum_{i=1}^{N} p_i, \overline{q} = \frac{1}{N} \sum_{i=1}^{N} q_i$$

2. Calculate the decentralized coordinate

$$p_i' = p_i - \overline{p}, \; q_i' = q_i - \overline{q}$$

3. Apply Singular Value Decomposition (SVD)

$$U \Sigma V^T = \sum_{i=1}^{N} p_i' q_i'^T$$

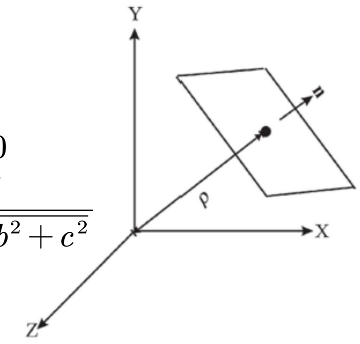
Deducing Details

4. Get rotation matrix R and translation vector t

$$R^* = VU^T, \; t^* = \overline{q} - R^* \overline{p}$$

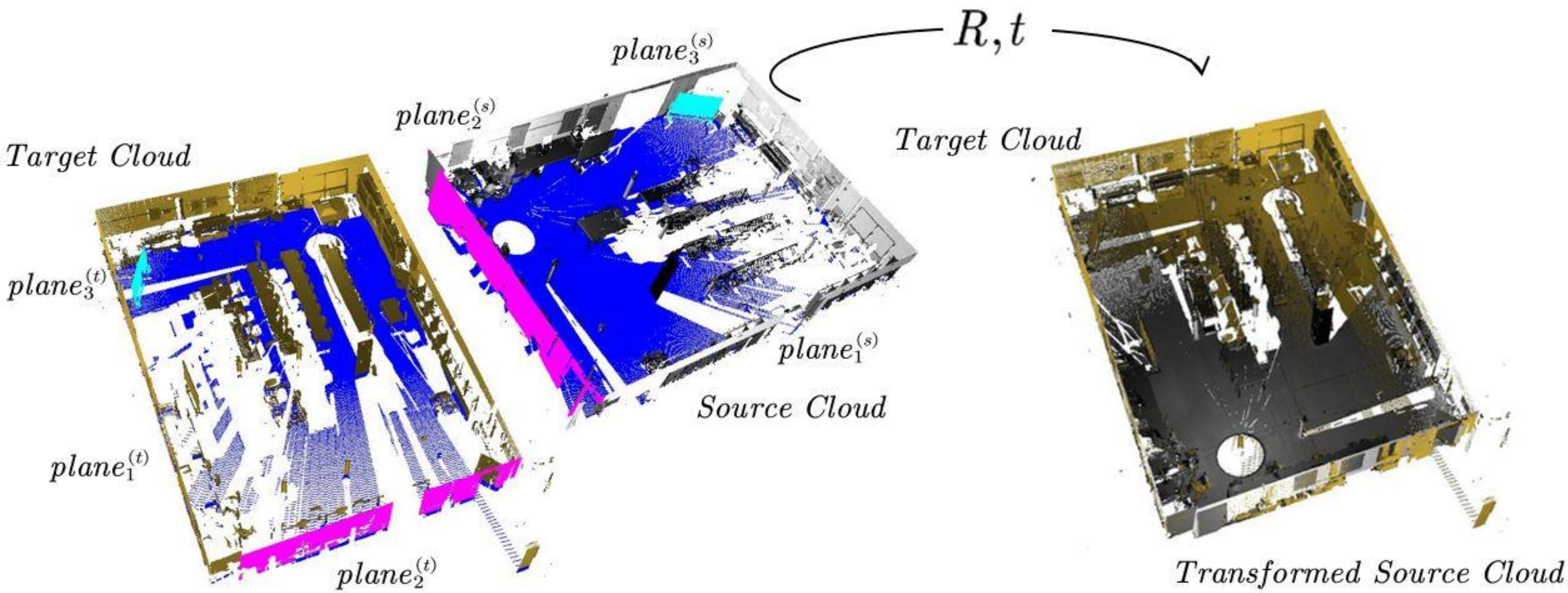
Transformation Estimation
for Plane Correspondences

$$ax + by + cz + d = 0$$
$$n = (a, b, c),\ \rho = \frac{d}{\sqrt{a^2 + b^2 + c^2}}$$



- Plane Correspondences
- Prepare: plane fitting by RANSAC
- Target function: $\{R^*, t^*\} = \underset{\{R,t\}}{\operatorname{argmin}} \left(\sum_i \left\| \begin{bmatrix} Rn_i^{(s)} - n_i^{(t)} \\ \rho_i^{(s)} - \rho_i^{(t)} + (Rn_i^{(s)})^T t \end{bmatrix} \right\|^2 \right)$

Previtali et.al. (2014 ISPRSA)



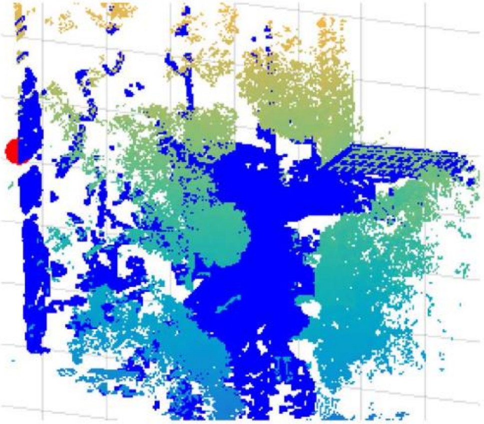
Transformation Estimation

for Plane Correspondences

$$ax + by + cz + d = 0$$

$$n = (a, b, c),\ \rho = \frac{d}{\sqrt{a^2 + b^2 + c^2}}$$

- 3 planes + 1 intersection point fine registration Kim et.al. (2017 JCCE)



2. Calculate translation vector from intersection points

$$x_{int}^{(t)} = \begin{bmatrix} a_1^{(t)} & b_1^{(t)} & c_1^{(t)} \\ a_2^{(t)} & b_2^{(t)} & c_2^{(t)} \\ a_3^{(t)} & b_3^{(t)} & c_3^{(t)} \end{bmatrix}^{-1} \begin{bmatrix} -d_1^{(t)} \\ -d_2^{(t)} \\ -d_3^{(t)} \end{bmatrix}$$

$$x_{int}^{(s)} = \begin{bmatrix} a_1^{(s)} & b_1^{(s)} & c_1^{(s)} \\ a_2^{(s)} & b_2^{(s)} & c_2^{(s)} \\ a_3^{(s)} & b_3^{(s)} & c_3^{(s)} \end{bmatrix}^{-1} \begin{bmatrix} -d_1^{(s)} \\ -d_2^{(s)} \\ -d_3^{(s)} \end{bmatrix}$$

$$t^* = x_{int}^{(t)} - x_{int}^{(s)}$$

1. Calculate rotation matrix from normal vectors

$$R = getRotation(v_1, v_2) = I + [v_1 \times v_2]_\times + [v_1 \times v_2]_\times^2 \frac{1 - v_1 \cdot v_2}{\|v_1 \times v_2\|}$$

$$R_1 = getRotation(n_1^{(s)}, n_1^{(t)})$$

$$R_2 = getRotation(R_1 n_2^{(s)}, n_2^{(t)})$$

$$R_3 = getRotation(R_2 R_1 n_3^{(s)}, n_3^{(t)})$$

$$R^* = R_3 R_2 R_1$$

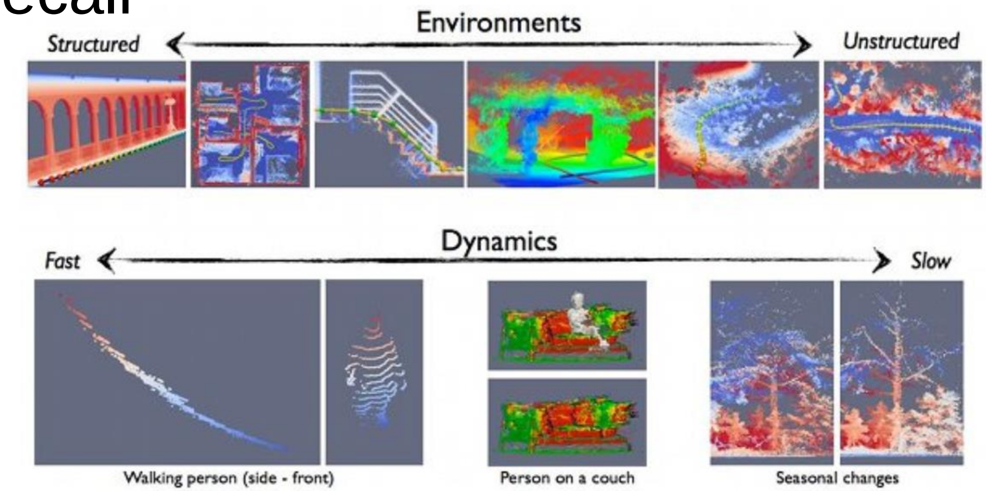
Plane correspondences are set according to the closest normal vector

Determine Correspondences

- Manual: Target based
- Automatic:
  - Geometric closest : ICP
  - Feature matching
  - Random sample

Bustos et.al. (2017 PAMI)

- Goal: High precision & recall
- Challenges
  - Low overlapping rate
  - Occlusion
  - Noise
  - Repetitive structure



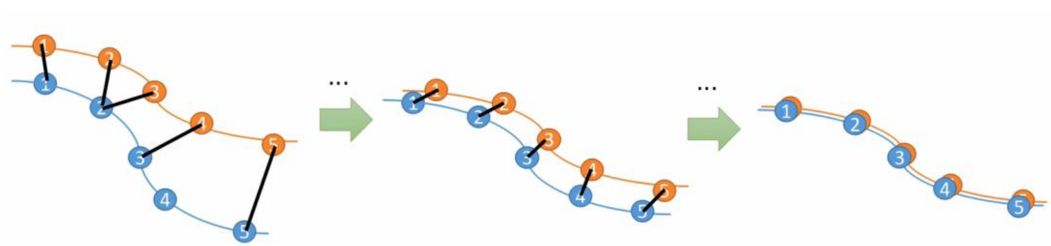
ETH ASL Registration Dataset

Iterative Closest Points (ICP) principle

Besl et.al. (1992 PAMI)

- Iteratively
  - Assume closest points of source and target point cloud as correspondences $M = \{p_i, q_i\}$
  - Estimate transformation T_{temp} from M
  - Transform source cloud with T_{temp}
  - Converged to local optima

1



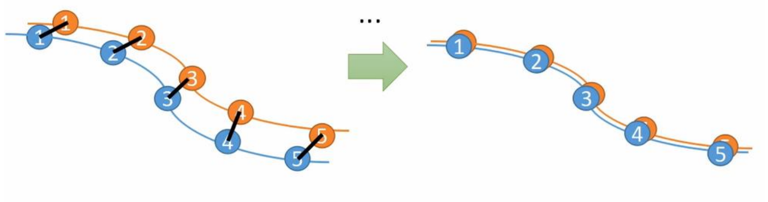
ICP Demo

ICP variants

Algorithm 1 Summary of ICP algorithm.

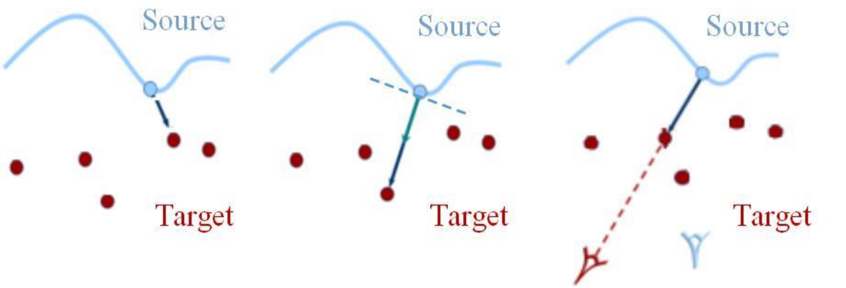
Require: ${}^{\mathbb{A}}\mathcal{P}$ — Source Point Cloud
Require: ${}^{\mathbb{B}}\mathcal{Q}$ — Target Point Cloud
Require: $\mathcal{T}_{init}$ — Transformation Initial Guess
$\quad {}^{\mathbb{A}}\mathcal{P}' \leftarrow \mathrm{datafilter}({}^{\mathbb{A}}\mathcal{P})$ — Filter Input Data
$\quad {}^{\mathbb{B}}\mathcal{Q}' \leftarrow \mathrm{datafilter}({}^{\mathbb{B}}\mathcal{Q})$
$\quad {}_{i-1}^{\ \ i}\mathcal{T} \leftarrow \mathcal{T}_{init}$
$\quad$ **repeat**
$\qquad {}^{i}\mathcal{P}' \leftarrow {}_{i-1}^{\ \ i}\mathcal{T}({}^{i-1}\mathcal{P}')$ — Update Source Point Cloud
$\qquad \mathcal{M}_i \leftarrow \mathrm{match}({}^{i}\mathcal{P}', \mathcal{Q}')$ — Determine correspondences
$\qquad \mathcal{W}_i \leftarrow \mathrm{outlier}(\mathcal{M}_i)$ — Filter outliers
$\qquad {}_{i}^{i+1}\mathcal{T} \leftarrow \underset{\mathcal{T}}{\arg\min}\left(\mathrm{error}\left(\mathcal{T}\left({}^{i}\mathcal{P}'\right), \mathcal{Q}'\right)\right)$ — Estimate Transformation (minimize error)
$\quad$ **until** convergence
Ensure: ${}_{\mathbb{A}}^{\mathbb{B}}\hat{\mathcal{T}} = \left(\bigcirc_{i}\, {}_{i-1}^{\ \ i}\mathcal{T}\right) \circ \mathcal{T}_{init}$ — Cumulate Final Transformation

François Pomerleau et.al. (2015 FTR)



- Filter input data:
  - Downsampling (efficiency)
  - Extract keypoints (efficiency and saliency)

- Determine correspondences:
  - Closest point (classic)
  - Normal shooting (smooth structure)
  - Viewpoint projection (efficiency)



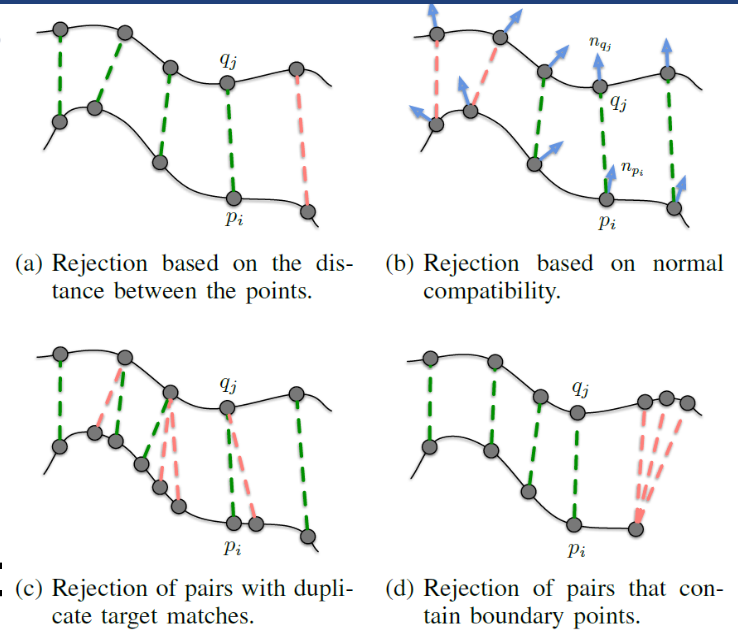
(a) closest point (b) normal shooting (c) viewpoint projection

Wolfram et.al. Introduction to Mobile Robotics

ICP variants

Holz et.al. (2015 RAM)

- Filter outlier:
  - Correspondence distance threshold
  - Trimmed-ICP: estimate distance threshold using overlapping ratio Chetverkiov et.al. (2002 ICPR)
  - Normal compatibility, Unique…



(a) Rejection based on the distance between the points. (b) Rejection based on normal compatibility. (c) Rejection of pairs with duplicate target matches. (d) Rejection of pairs that contain boundary points.

- Distance metrics for minimization:
  - Point-to-Point (classic)

$$\{R^*, t^*\} = \underset{\{R,t\}}{\operatorname{argmin}} \left(\sum_i \| Rp_i + t - q_i \|^2 \right)$$

  - Point-to-Plane (façade, ground) Chen et.al. (1992 IVC)

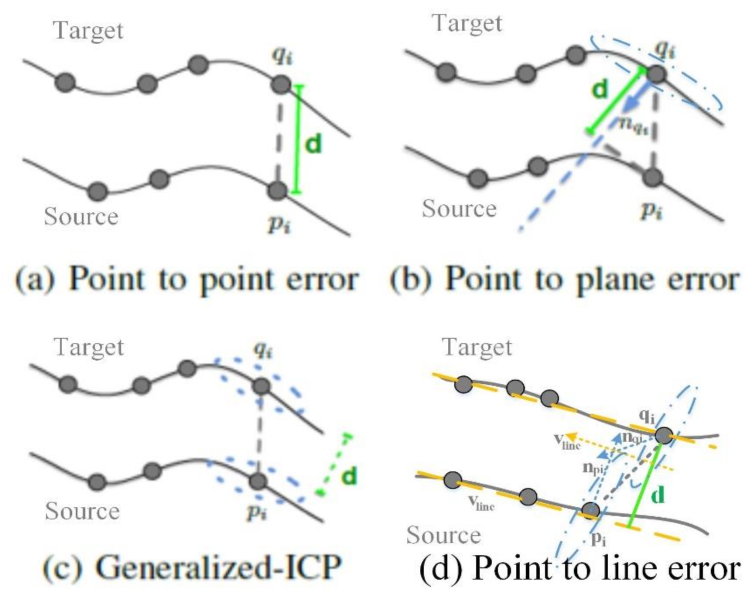
$$\{R^*, t^*\} = \underset{\{R,t\}}{\operatorname{argmin}} \left(\sum_i \| (Rp_i + t - q_i) \cdot n_i^{(q)} \|^2 \right)$$

  - Point-to-Line (trunk, pillar) Censi (2008 ICRA)

$$\{R^*, t^*\} = \underset{\{R,t\}}{\operatorname{argmin}} \left(\sum_i \left\| (Rp_i + t - q_i) \times \widehat{(n_i^{(q)} \times n_i^{(p)})} \right\|^2 \right)$$

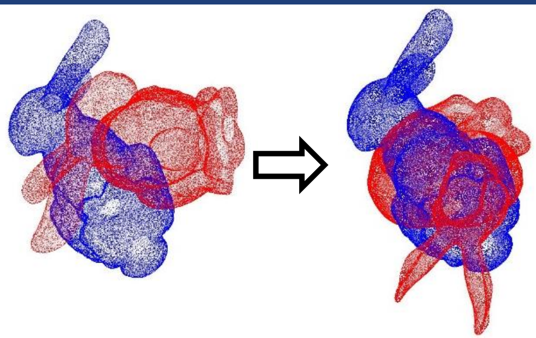
  - General ICP (neighborhood covariance)

Segal et.al. (2009 RSS)

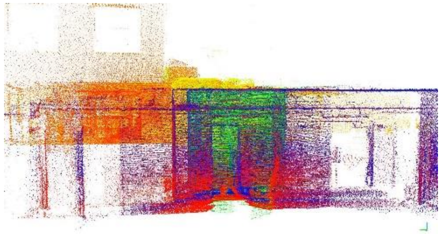


(a) Point to point error (b) Point to plane error (c) Generalized-ICP (d) Point to line error

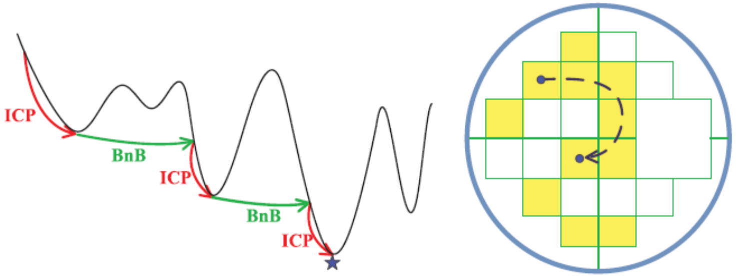
ICP: Pros and Cons



- Pros:
  - Simple structure, easy to implement
  - Somewhat efficient
  - High accuracy when reaching global optima (**Fine registration**) [popular in SLAM applications]



- Cons:
  - Converge to local optima when without good initial guess (**Local registration**)
  - Not robust to …
- Improvement:
  - Go-ICP (arbitrary initial guess)
  - S-ICP (larger convergence domain)



Yang et.al. (2017 PAMI)

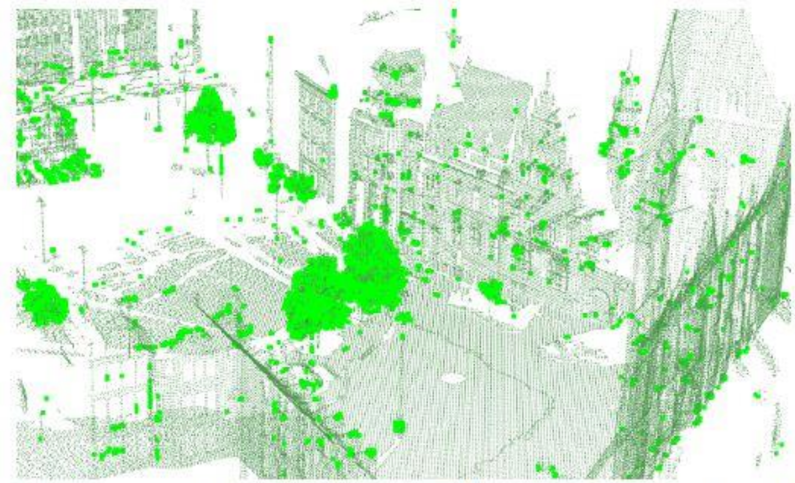
Feature matching

- Determine Correspondences: Feature matching

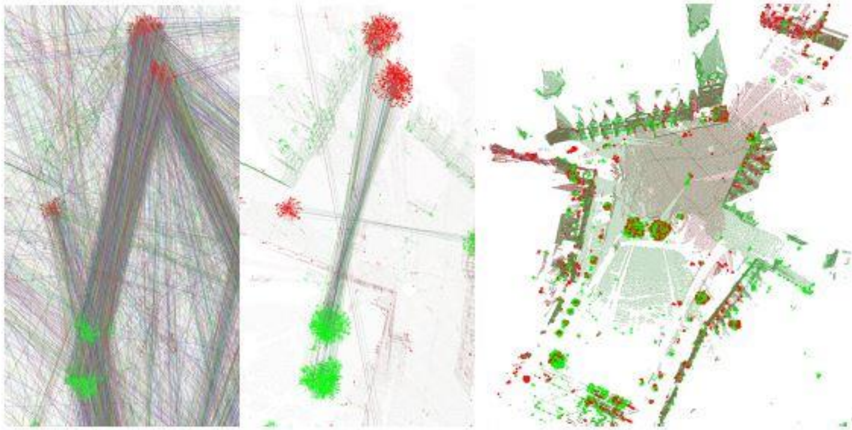
- Workflow:
  - 1.Detect keypoints
  - 2.Extract feature
  - 3.Initial feature matching
  - 4.Filter outliers

- Keypoints detection:
  - local curvature extremum

(neighborhood eigen value calculation)
  - 3D Harris …



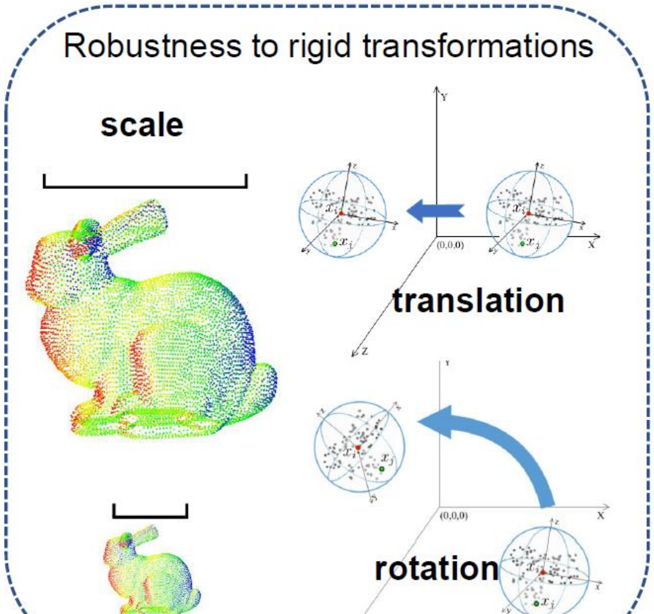
(a) Target point cloud with estimated keypoints (visualized with larger radii)



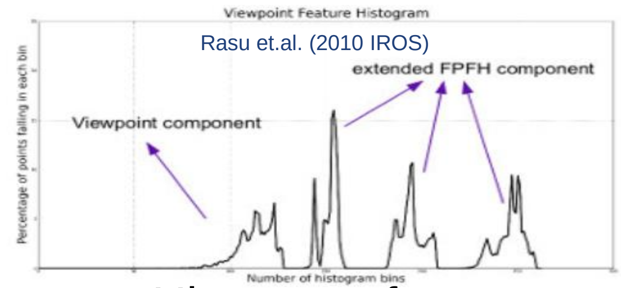
(b) Estimated correspondences (c) Filtered correspondences (d) Aligned point clouds

Holz et.al. (2015 RAM)

Handcrafted features

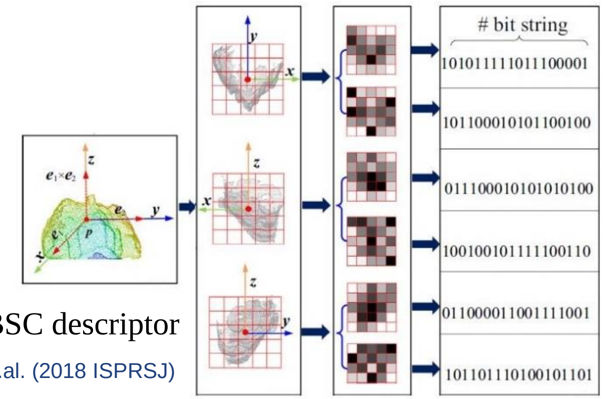


Liu (2019 CVPR)



Rasu et.al. (2010 IROS)

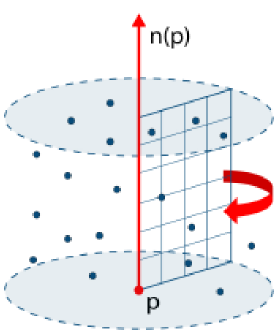
Histogram feature



(e) BSC descriptor

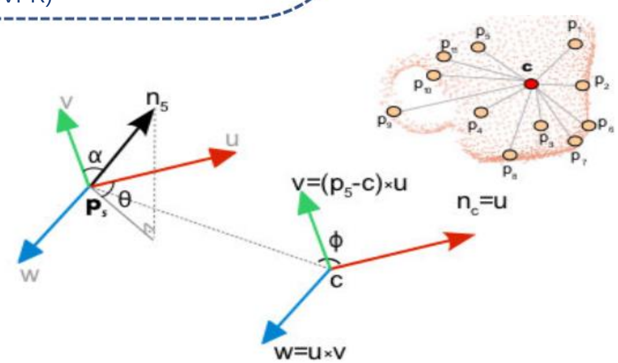
Dong et.al. (2018 ISPRSJ)

Binary feature



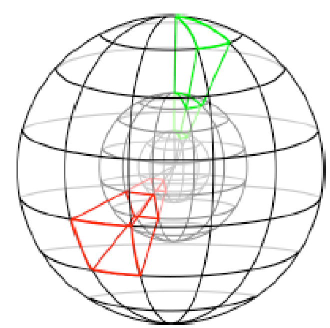
(a) Spin Images

Andrew (1997 thesis)



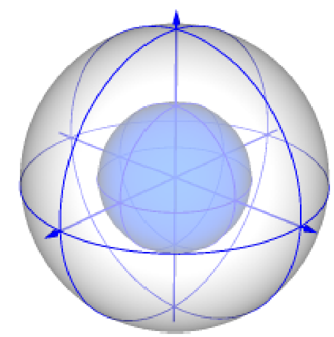
(b) FPFH Descriptor

Rasu et.al. (2009 ICRA)



(c) 3DSC Descriptor

Frome et.al. (2014 ECCV)



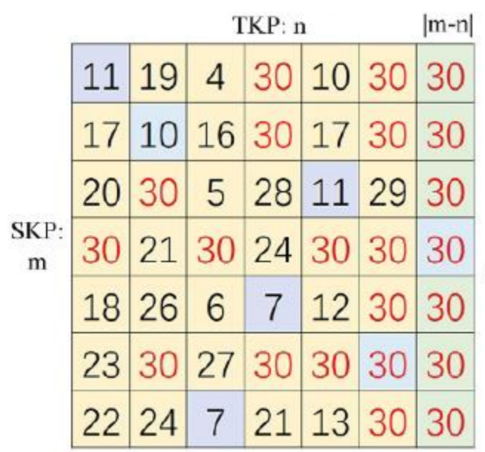
(d) SHOT Descriptor

Tombari et.al. (2010 ECCV)

Matching strategy

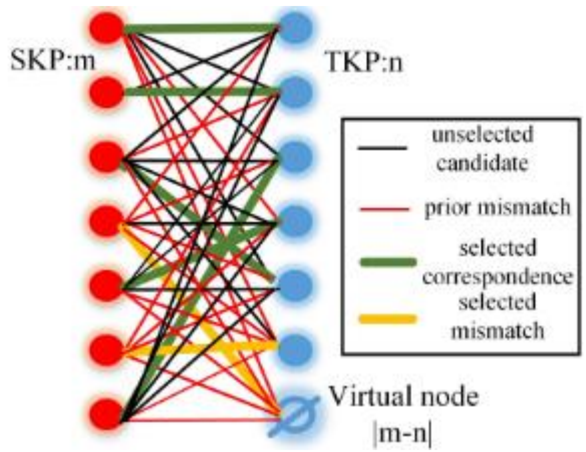
- Similarity / Distance calculation
  - SAD (L1), SSD (L2), Hamming (Binary)
  - Cosine similarity, Correlation

- Matching
  - One way / Reciprocal Nearest Neighbor
  - NN-SNN Ratio test
  - Bipartite graph



11	19	4	30	10	30	30
17	10	16	30	17	30	30
20	30	5	28	11	29	30
30	21	30	24	30	30	30
18	26	6	7	12	30	30
23	30	27	30	30	30	30
22	24	7	21	13	30	30

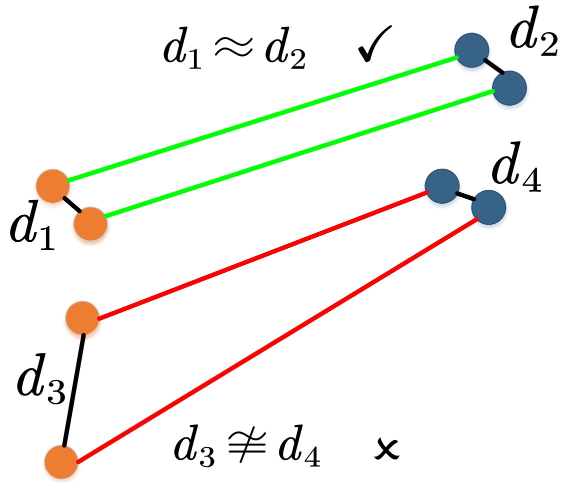
$$\min \sum (selected\ edge\ weight)$$

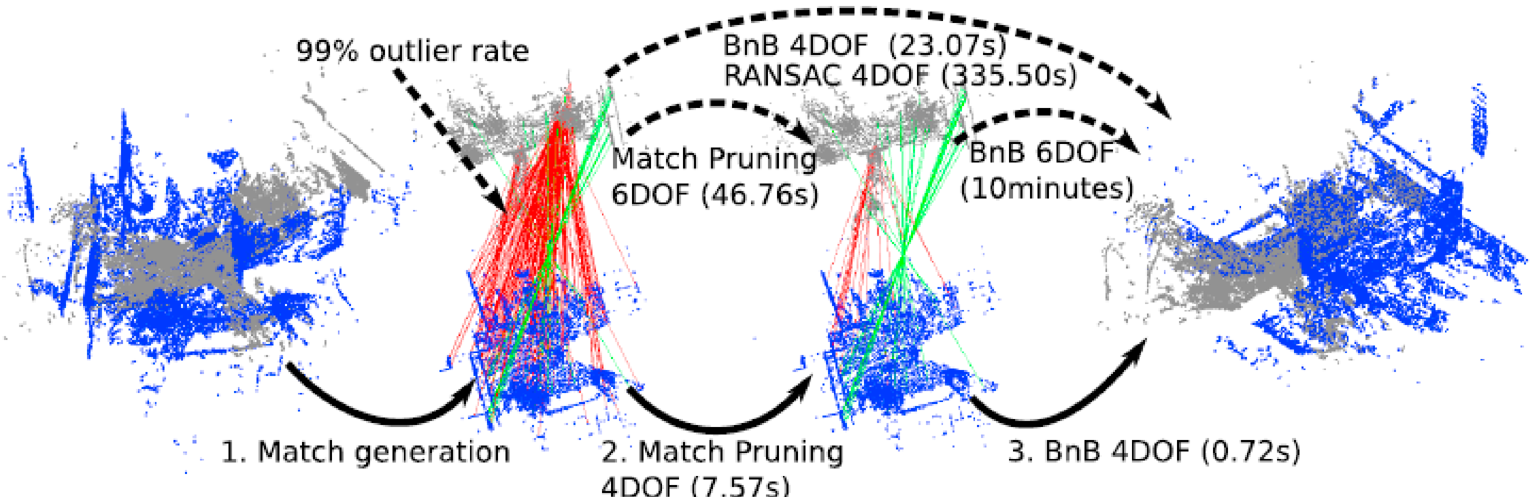


Pan et.al. (2018 3DV)

Robust outlier filtering

- RANSAC based on correspondence
  - Find 3 outlier-free correspondences
- Geometric consistency
- Guaranteed outlier removal (GORE)

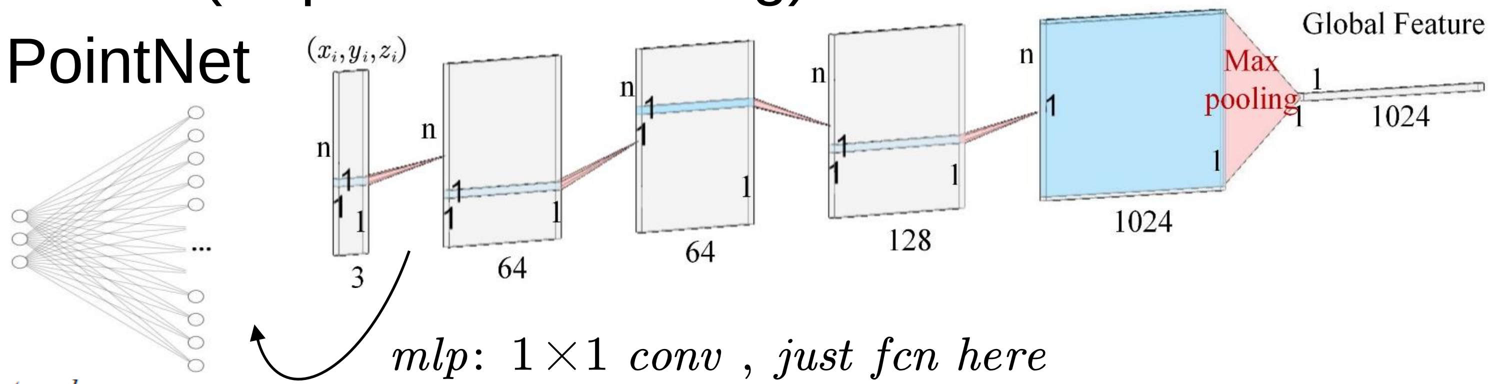


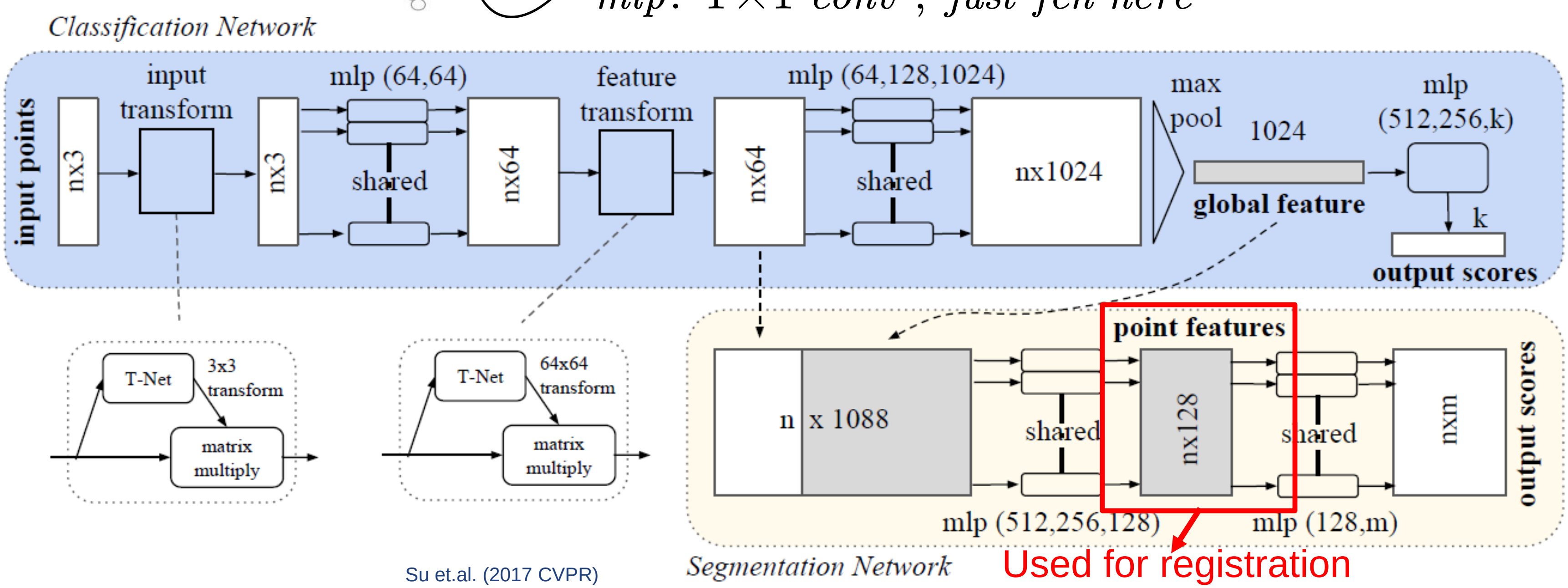


Cai et.al. (2019 ISPRSJ)

Registration based on learned feature

- Deep feature (Supervised learning)
- Basic: PointNet

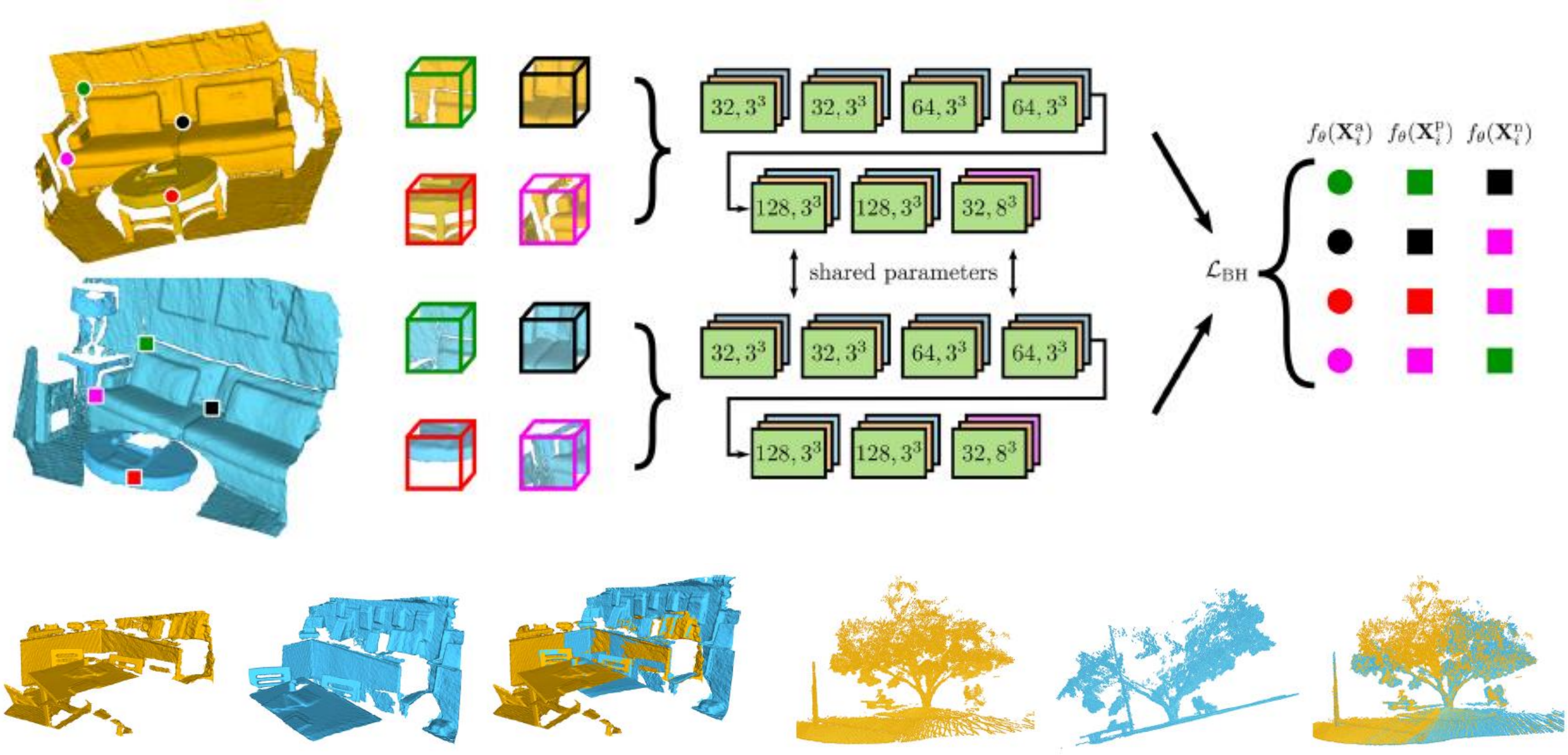




Su et.al. (2017 CVPR)

Registration based on learned feature

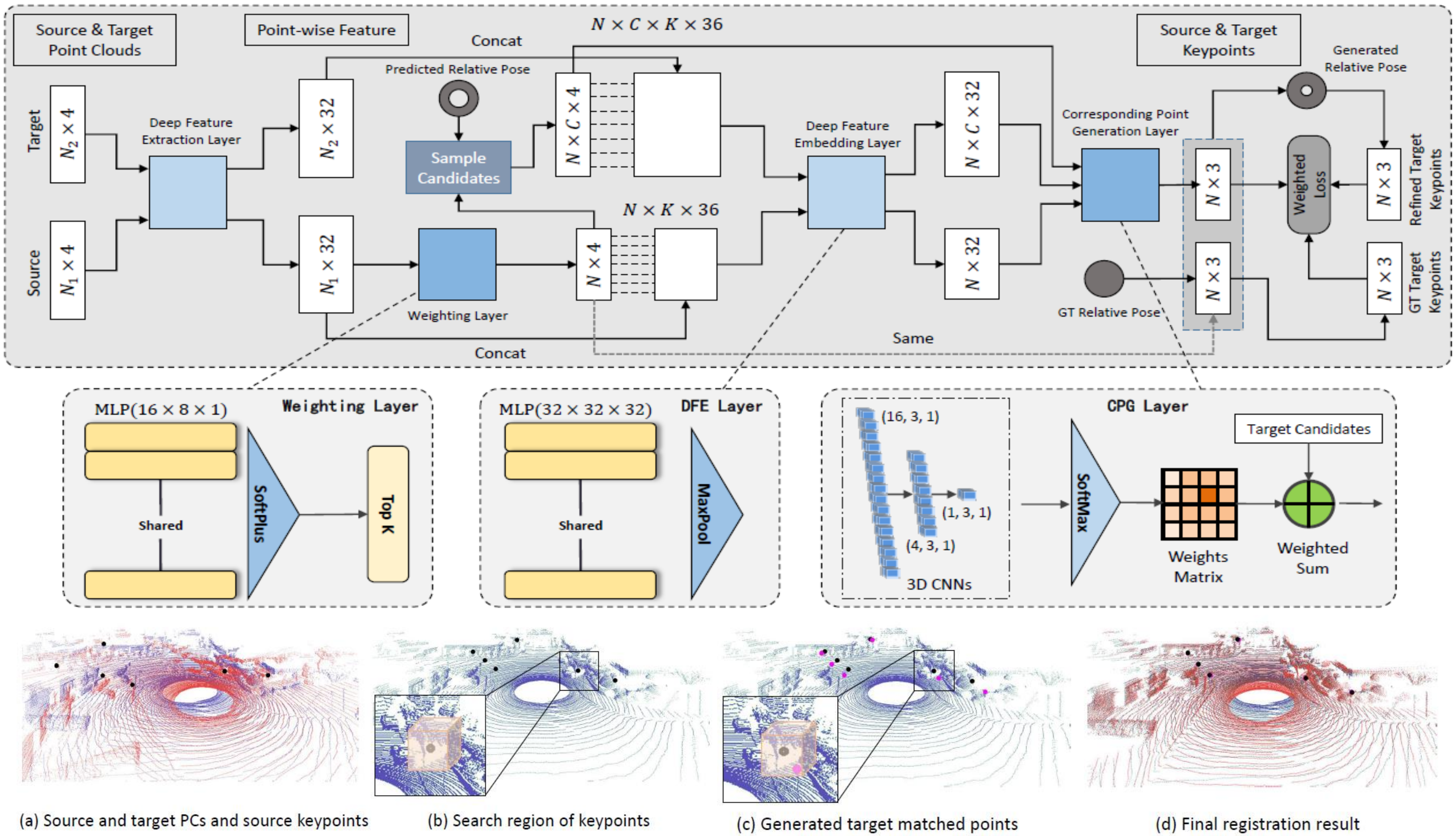
- End-to-end: features and correspondences
- Example: 3D Smooth Net



Gojcic et.al. (2019 CVPR)

Registration based on learned feature

- End-to-end: features, correspondence and transformation
- Example: DeepVCP



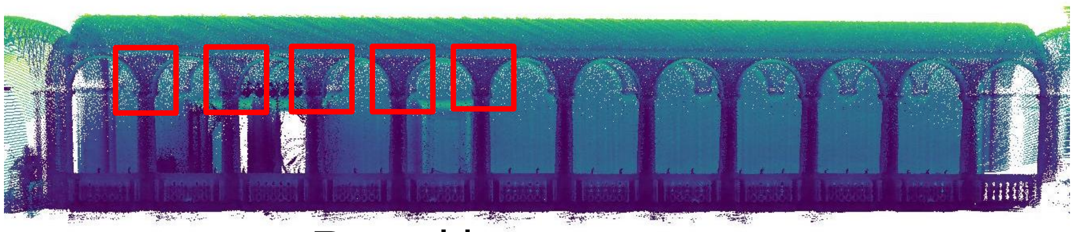
Lu et.al. (2019 ICCV)

Feature based methods: Pros and Cons

- Pros:
  - No requirement of transformation initial guess **(Global registration)**
  - Robust (with proper matching strategy and outlier filter)

- Cons:
  - Not accurate enough **(Coarse registration)**
  - Often based on keypoints (the raw point cloud should be dense)
  - May be time-consuming

- Outlook:
  - Deep learning based



Repetitive structures

Randomized hypothesize-and-verify

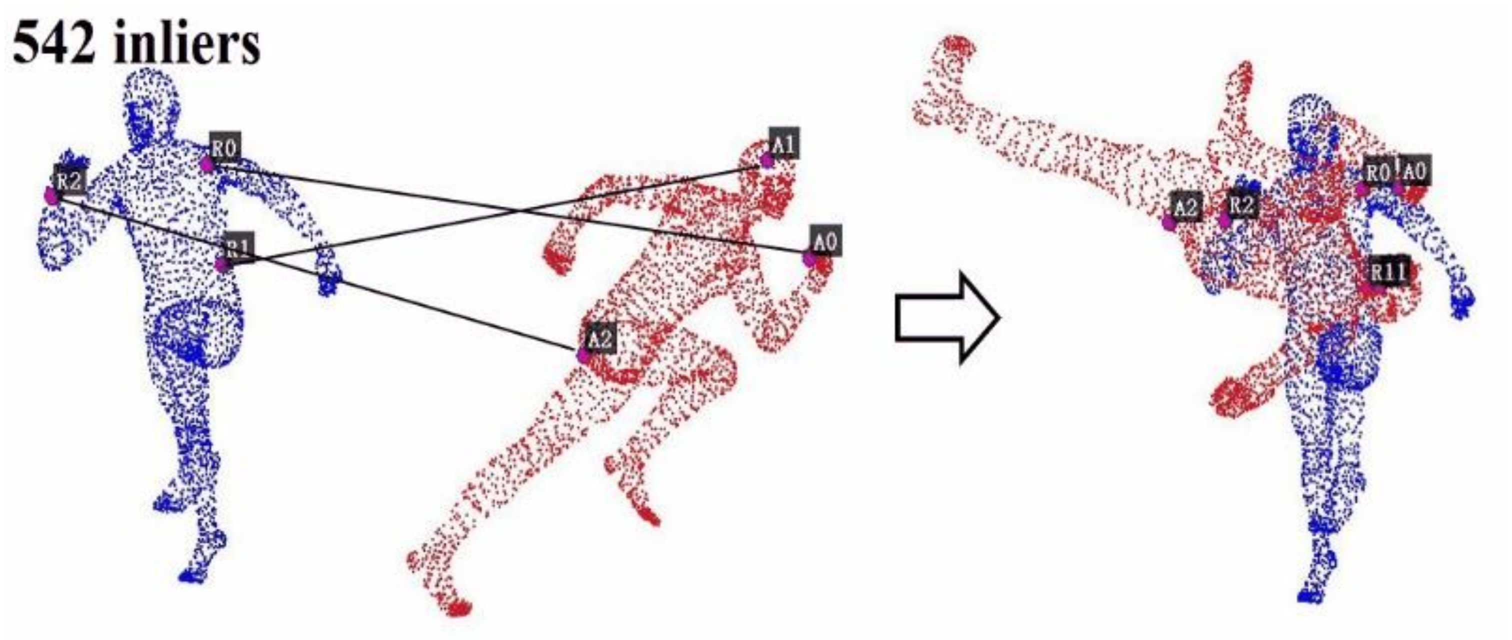
- RANSAC without correspondences

Chen et.al. (1999 PAMI)

  - Find two outlier-free 3-point subsets with correct correspondences
- Problem: efficiency (Total sampling number M)

$p = 1 - (1 - r^N)^M > 0.99$ $\quad r$: *point inlier ratio*, $N = 3$

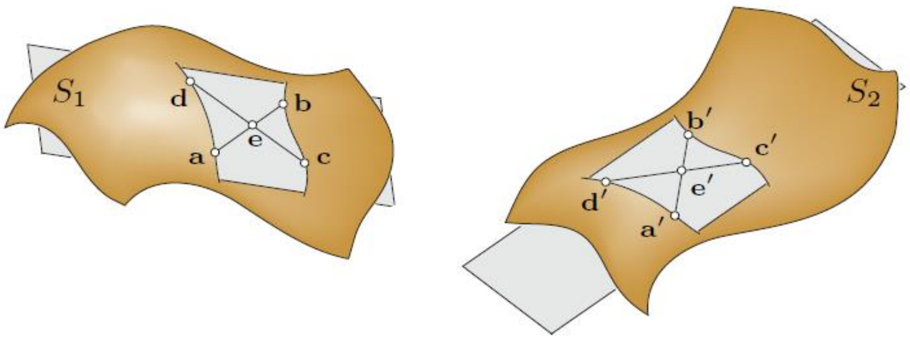
For each hypothesize to find the geometric consistent 3-point subset in the other point cloud $O(n^3)$

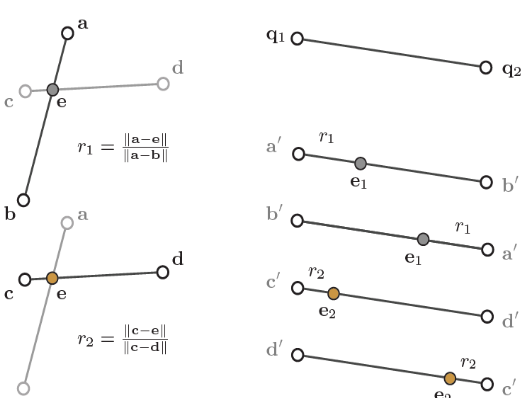


Efficiency improvement: 4PCS

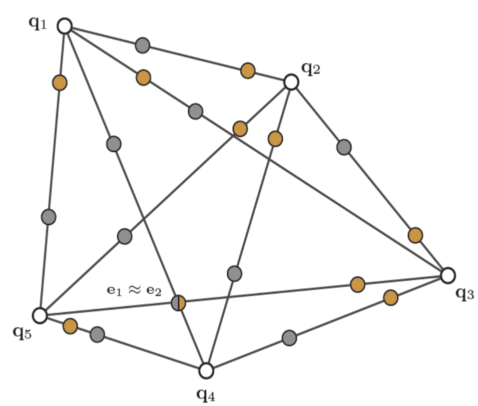
4 Points Congruent Set

- $O(n^2)$ searching of affine-invariant 4 coplanar point sets
- Invariant of intersection ratio





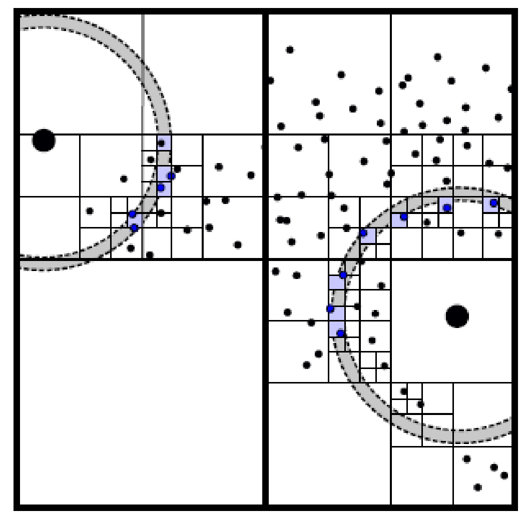
Aiger et.al. (2008 SIGGRAPH)



- Variants:
  - Super4PCS: Mellado et.al. (2014 SGP)

    Smart-indexing

    (efficiency) $O(n^2) \Rightarrow O(n)$

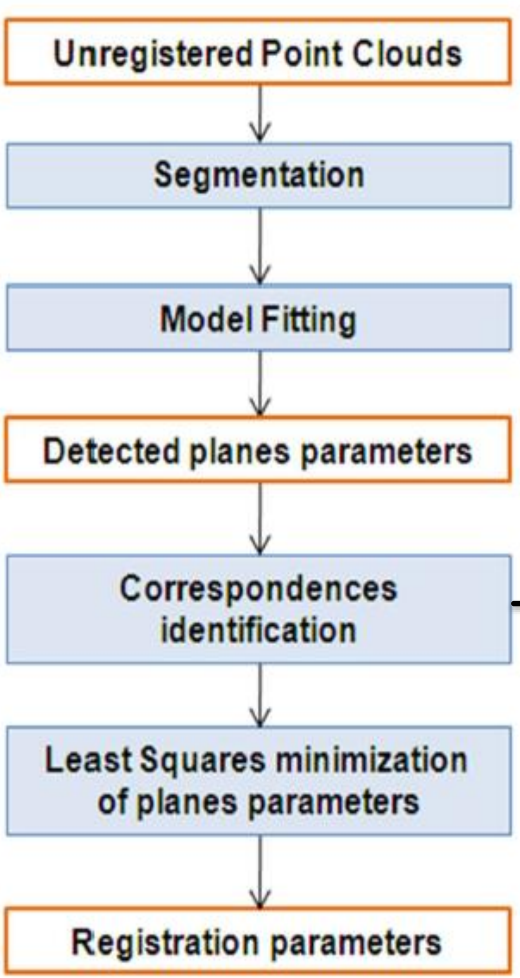


  - K4PCS: Theiler et.al. (2017 ISPRSJ)

    Base on keypoints

    (efficiency)

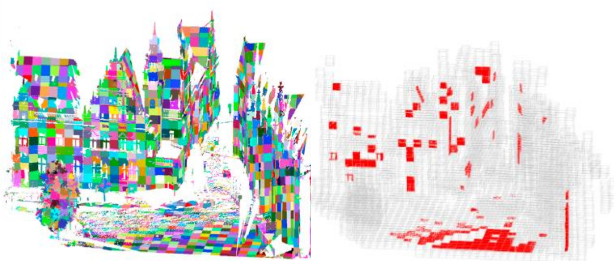
Example: Plane Based RANSAC

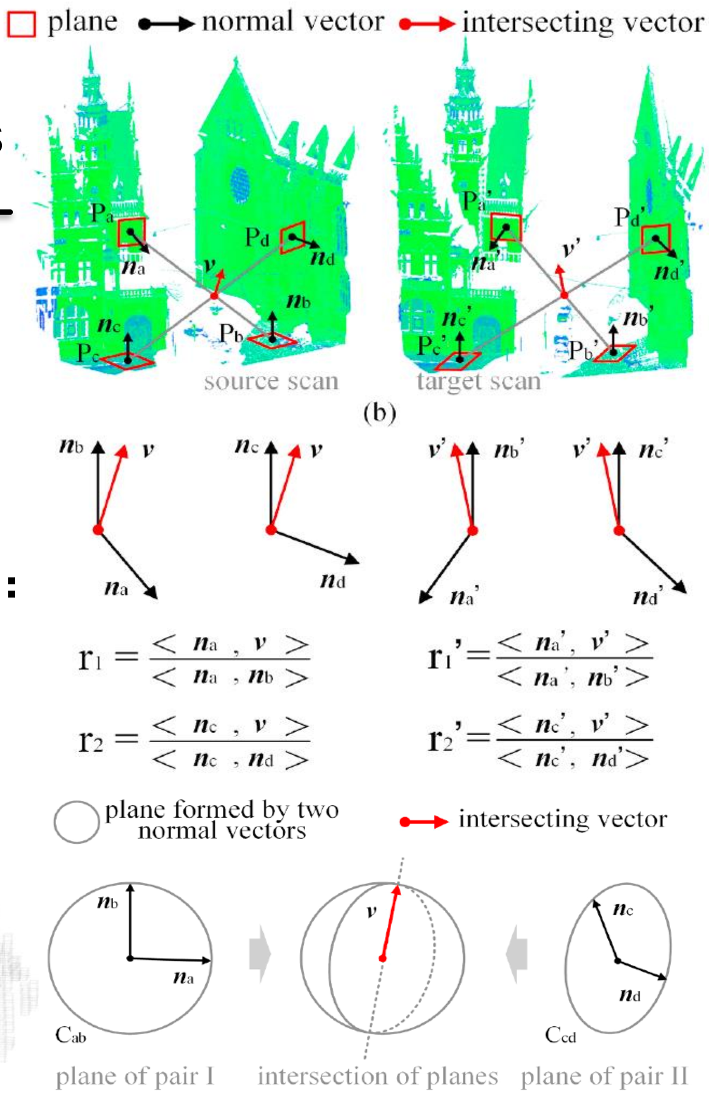


Previtali et.al. (2014 ISPRSA)

Voxel-based
4-plane congruent sets
(**V4PCS**):
Improvement of sampling strategies, plane version of 4PCS

Plane based RANSAC：
Minimum 3-planes correspondence





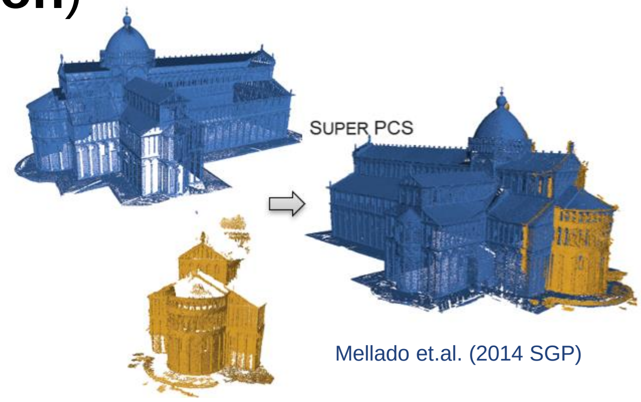
Xu et.al. (2019 ISPRSJ)

RANSAC based methods: Pros and Cons

- Pros:
  - No requirement of transformation initial guess **(Global registration)**
  - Robust to noise, outliers and similar / repetitive structures

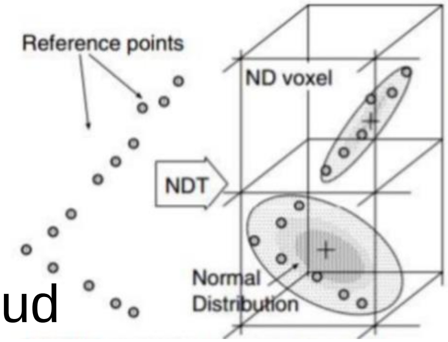
- Cons:
  - With noise and outliers, the likelihood of picking outlier-free subsets degrades rapidly
  - Could be very time-consuming or you need to do downsampling
  - Not accurate enough (**Coarse registration**)

- Outlooks:
  - Better sampling strategies



Mellado et.al. (2014 SGP)

Another way: Probability
a brief intro.



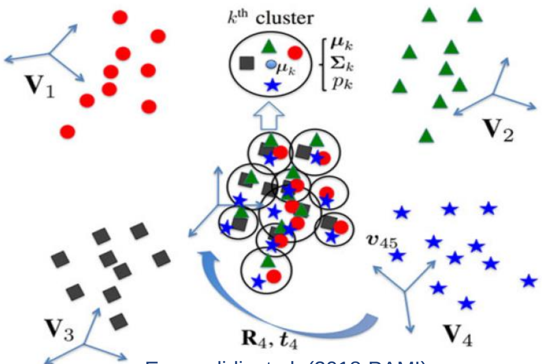
Magnusson et.al. (2009 ICRA)

Magnusson (2013 thesis)

- Normal Distribution Transformation (NDT)
  - Fit Gaussian distribution $p_N^{(t)}$ in each voxel of target cloud

$$R^*, t^* = \underset{R,t}{arg\max} \left(\prod_i p_N^{(t)} (Rx_i^{(s)} + t) \right)$$

  - Faster and has larger convergence domain than ICP, but not stable
  - Fine registration (Applications like SLAM, Localization)
- Coherent Point Drift (CPD) Myronenko et.al. (2010 PAMI)
  - Each point in Target cloud: one Gaussian kernel centroid
  - Points in Source cloud: Generate by the GMM of Target cloud
  - Solved by EM
- GMM-Registration Jian et.al. (2011 PAMI)
- GUGMA : GMMReg.+BnB Campbell et.al. (2016 CVPR)

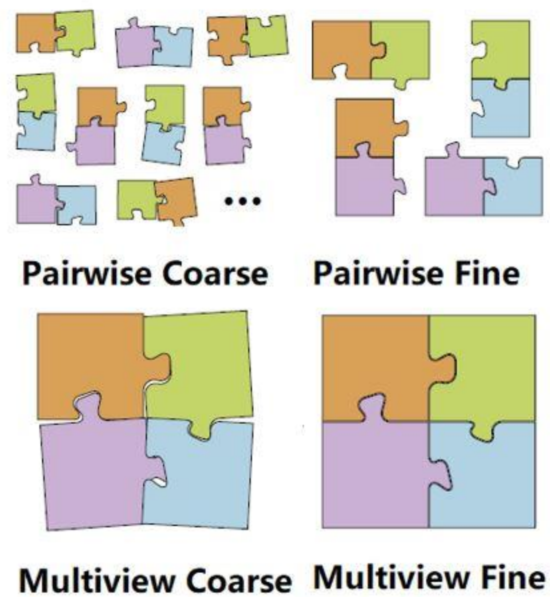


Evangelidiset.al. (2018 PAMI)

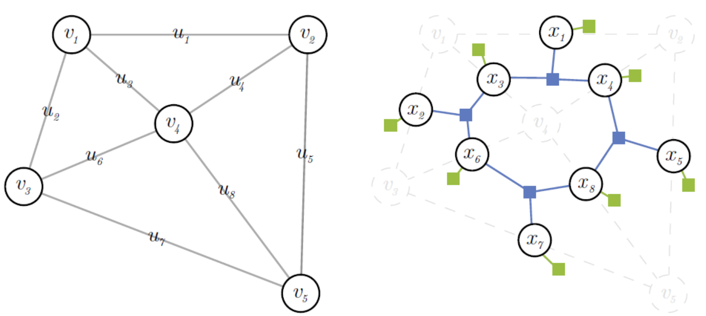
Combination

- Coarse to Fine

- Pairwise to Multiview
  - Joint optimization (adjustment)
  - Minimum Spanning Tree
  - Pose Graph Optimization

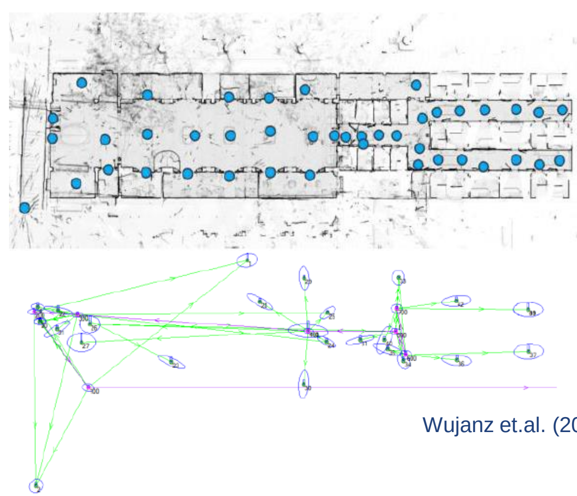


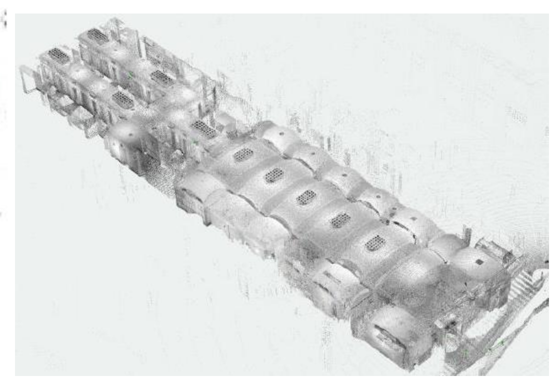
Theiler et.al. (2015 ISPRSJ)



(a) Basic scan graph $\mathcal{G}_S$ (b) Corresponding graphical model $\mathcal{G}_R$

Theiler et.al. (2015 ISPRSJ)





Wujanz et.al. (2019 ISPRSA)

Summary

- Two steps
  - Determine correspondences (points, planes, etc.)
  - Estimate transformation according to the target function

- Methods
  - Feature based (Coarse)
  - RANSAC based (Coarse)
  - ICP (Fine)
  - Probability based

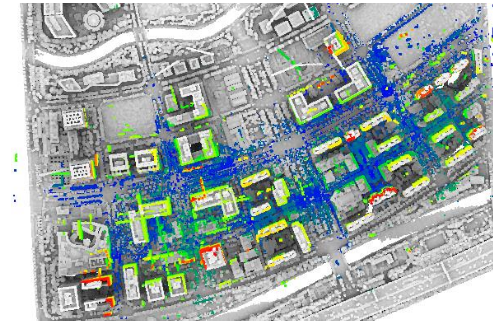
- Challenges
  - small overlap, occlusion, noise…
  - large variance of point density
  - consuming time (for real-time applications)

Open Questions

- Cross-platform Registration
  - ALS and TLS

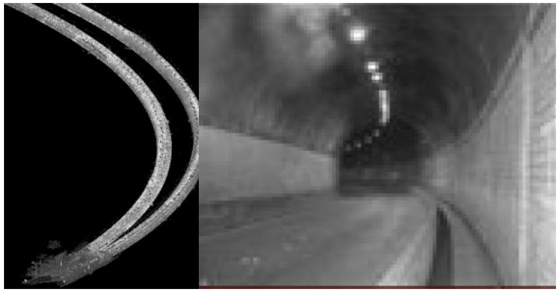
- Real-time Lidar Odometry
  - KITTI Odometry top 1: V-LOAM
  - (ATE 0.55, ARE 0.13) still not accurate enough
  - Challenging case: Tunnel mapping

- Little-overlapping registration
  - Digital cultural relic restoration

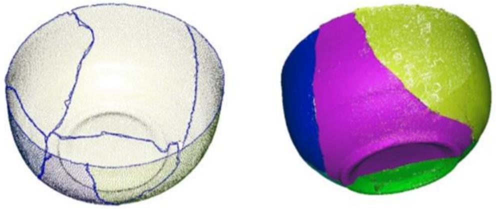


Liang et.al. (2019 CLC)

KITTI Dataset: Odometry



Zhang et al. (2018 JFR)



Wang et.al. (2019 CLC)

Q & A:

- Questions for audiences:

What's your idea about the listed open questions?

What's the future of a complete end-to-end point cloud registration network?

What's the open question on the topic of point cloud registration in your opinion?

Thanks

- Thanks.

Transformation Estimation
SVD in detail

$$\overline{p} = \frac{1}{N}\sum_{i=1}^{N} p_i ,\ \overline{q} = \frac{1}{N}\sum_{i=1}^{N} q_i$$

Calculate centroids

$$E(R,t) = \sum_i \| Rp_i + t - q_i \|^2$$

$$= \sum_i \| R(p_i - \overline{p}) - (q_i - \overline{q}) + (R\overline{p} + t - \overline{q}) \|^2$$

$$= \sum_i \| R(p_i - \overline{p}) - (q_i - \overline{q}) \|^2 \qquad R\overline{p} + t - \overline{q} = 0$$

$$R^* = \arg\min_R \sum_i \| R(p_i - \overline{p}) - (q_i - \overline{q}) \|^2$$

$$= \arg\min_R \sum_i \| Rp_i' - q_i' \|^2 \quad \textit{decentralized coordinates}$$

$$= \arg\min_R \sum_i (p_i'^T R^T R p_i' + q_i'^T q_i' - 2q_i'^T R p_i')$$

$$= \arg\max_R \sum_i (q_i'^T R p_i')$$

$$R^* = \arg\max_R \sum_i (trace(Rp_i' q_i'^T))$$

$$= \arg\max_R \left(trace\left(R \sum_i p_i' q_i'^T \right)\right) \quad \textit{the properties of trace}$$

$$= \arg\max_R (trace(RH)) \quad \textit{denote } \sum_i p_i' q_i'^T \textit{ as } H$$

$$= \arg\max_R (trace(RU\Sigma V^T)) \quad SVD:\ H = U\Sigma V^T$$

$$= \arg\max_R (trace(V^T R U \Sigma))$$

$$= \arg\max_R (trace(R'\Sigma)) \quad \textit{demote } V^T RU \textit{ as } R'$$

$$= V\left(\arg\max_{R'} (trace(R'\Sigma)) \right) U^T \quad \textit{so } R = VR'U^T$$

$$= V\left(\arg\max_{R'} (r_{11}'\sigma_1 + r_{22}'\sigma_2 + r_{33}'\sigma_3) \right) U^T$$

$$= VU^T \quad \textit{when } r_{11}' = r_{22}' = r_{33}' = 1 \Rightarrow R' = I$$

$$t^* = \overline{q} - R^* \overline{p}$$

Back

Transformation Estimation
Linear least square

Correspondences known

$$\{p_i, q_i\}$$

$$\{R^*, t^*\} = \underset{\{R,t\}}{\operatorname{argmin}} \left(\sum_i \| Rp_i + t - q_i \|^2 \right)$$

1.Linear Approximation

$$\sin(\alpha) \approx \alpha \; when \; \alpha \to 0$$

$$R \approx \begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}_\times + I$$

2.Observation function

$$v_i = \begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix} \begin{bmatrix} x_i^{(p)} \\ y_i^{(p)} \\ z_i^{(p)} \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} - \begin{bmatrix} x_i^{(q)} \\ y_i^{(q)} \\ z_i^{(q)} \end{bmatrix}$$

3.Parameter estimation by LS

$$\underbrace{\begin{bmatrix} v_i^{(x)} \\ v_i^{(y)} \\ v_i^{(z)} \end{bmatrix}}_{v_i} = \underbrace{\begin{bmatrix} 0 & z_i^{(p)} & -y_i^{(p)} & 1 & 0 & 0 \\ -z_i^{(p)} & 0 & x_i^{(p)} & 0 & 1 & 0 \\ y_i^{(p)} & -x_i^{(p)} & 0 & 0 & 0 & 1 \end{bmatrix}}_{A_i} \underbrace{\begin{bmatrix} \alpha \\ \beta \\ \gamma \\ t_x \\ t_y \\ t_z \end{bmatrix}}_{x} - \underbrace{\begin{bmatrix} x_i^{(q)} - x_i^{(p)} \\ y_i^{(q)} - y_i^{(p)} \\ z_i^{(q)} - z_i^{(p)} \end{bmatrix}}_{l}$$

$$\underbrace{\begin{bmatrix} v_1 \\ \vdots \\ \vdots \\ v_n \end{bmatrix}}_{V} = \underbrace{\begin{bmatrix} A_1 \\ \vdots \\ \vdots \\ A_n \end{bmatrix}}_{A} x - \underbrace{\begin{bmatrix} l_1 \\ \vdots \\ \vdots \\ l_n \end{bmatrix}}_{l}$$

$$\hat{x} = [\alpha \;\; \beta \;\; \gamma \;\; t_x \;\; t_y \;\; t_z]^T = (A^T A)^{-1} A^T l$$

$$get \; R^*, t^*$$

Back

Contact information and credits

ETH Zurich

D-BAUG

Yue Pan

yuepan@student.ethz.ch