3D Rigid Registration

Algorithms survey

Pasquale Antonante

September 14, 2017

Introduction

Problem definition

Surface registration

Surface registration **transforms** multiple sets of 3D data points into the same coordinate system so as to **align overlapping** components of these sets.

Basic notation i

Let's consider two point sets $\boldsymbol{P}$ and $\boldsymbol{Q}$ we want to find a rigid transformation $\boldsymbol{T}$ (rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$) that aligns $\boldsymbol{Q}$ to $\boldsymbol{P}$.

In general we want to minimize the error $E(\boldsymbol{T})$

$$E(\boldsymbol{T}) = E_{\text{align}} + E_{\text{reg}}, \quad \text{where} \boldsymbol{T} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix}$$

E_{align} measure the **alignment error** while E_{reg} is a **regularization** term.

Basic notation ii

There are two main metrics to measure the E_{align}

$$E_{\text{pt-to-pt}} = \sum_i \|\boldsymbol{q}_i - \boldsymbol{T}\boldsymbol{p}_i\|^2 \tag{1}$$

$$E_{\text{pt-to-plane}} = \sum_i \left((\boldsymbol{q}_i - \boldsymbol{T}\boldsymbol{p}_i) \cdot \boldsymbol{n}_{q_i}\right)^2 \tag{2}$$

Basic notation iii

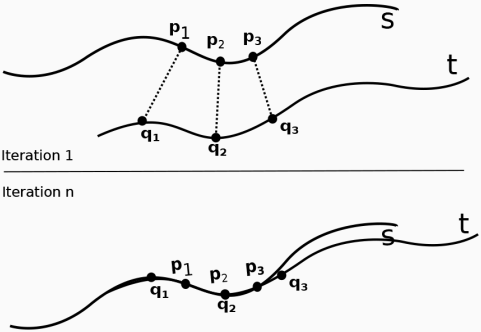


Figure 1: Point-to-Point Distance

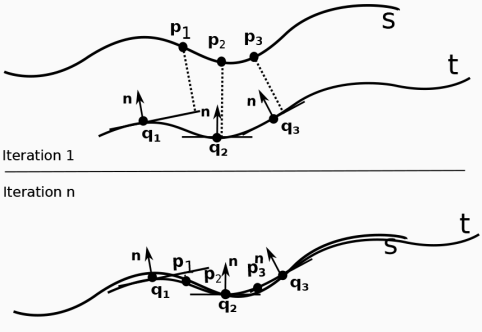


Figure 2: Point-to-Plane Distance

Figure 3: Error metrics (Bellekens et al., 2014)

Basic notation iv

Point-to-plane proved to be more stable and faster to converge but...**Least Sum of Square Errors** has a closed-form solution!

If we know the **perfect correspondence** of a subset of points (at least 3) we can compute the rigid transformation

- Rotation matrix, e.g. (Schönemann, 1966), (Arun et al., 1987), (Horn et al., 1988), (Umeyama, 1991)
- Quaternions, e.g. (Horn, 1987)

But **true correspondences** are difficult do be found.

(Non)-Convexity Analysis i

Let's focus on the **point-to-point** formulation.

Define

- Transformation function as $T_x(\alpha)$ that **affinely transforms** a point x, according to some parameter $\alpha = (\boldsymbol{R}, \boldsymbol{t})$
- The **distance-to-a-set operator** $d(x) = \inf_{y \in \mathcal{Y}} \|x - y\|$

The **residual function** $E(\alpha) = d(T_x(\alpha))$ is *convex* if

- *(Condition 1).* Domain D_α is a convex set
- *(Condition 2).* The set $\mathcal{Y}$ is convex[1]

(Non)-Convexity Analysis ii

In our case

- *Condition 1* cannot be fulfilled for registration with rotation (due to the constraint $\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}$)
- *Condition 2* is rarely fulfilled because set $\mathcal{Y}$ is a scan of complex surfaces

Therefore, $E(\alpha)$ is **non-convex**.

[1] Proof in (Olsson et al., 2009)

Global vs Local

Global vs. Local

3D Rigid Registration can be classified according to the used **underlying optimization method** (Rusu et al., 2009):

- **Global**
- **Local** (e.g. Iterative Closest Point (ICP))

Iterative Closest Points (ICP)

Basic ICP

Algorithm 1 Iterative Closest Points (ICP)

Input: $\boldsymbol{P}$, $\boldsymbol{Q}$ and initial estimation $\boldsymbol{T}_0$

Output: $\boldsymbol{T}$

```
T ← T_0
while not converged do
    for i ← 1 to N do
        m_i ← FindClosestPointInQ(T p_i)
        if ||m_i − T p_i|| ≤ d_max then
            ω_i ← 1
        else
            ω_i ← 0
        end if
    end for
    T ← argmin_T Σ_i ω_i ||m_i − T p_i||
end while
```

Considerations

- 👍 Solves the correspondence problem
- 👎 May get caught in local minima
- 👎 Require (and depends) an initial registration guess
- 👎 Computes closest points set each iteration

Local Optimization Method

There is a lot of work around ICP addressing the local minima issue, some relevant methods are:

- **Robustified Local Methods**
- **Global Methods** (GA, Particle filtering, Simulated Annealing)
- **Globally Optimal Methods** (i.e. BnB)

GO-ICP (2016)

Introduction

Global-ICP (GO-ICP) (Yang et al., 2016) optimally solves 3D registration by mixing ICP and BnB.

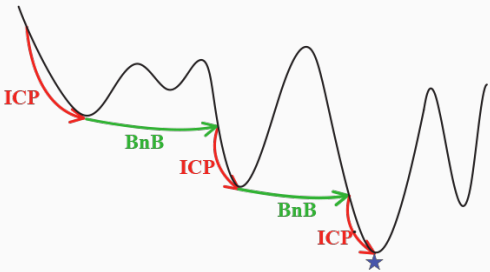
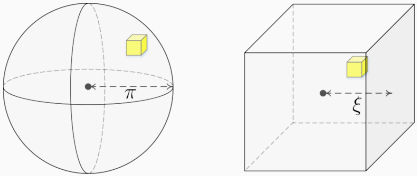


Figure 4: Collaboration of BnB and ICP.

Domain Parametrization i

To parametrize the search space let's consider the **angle-axis** representation of rotations, obtaining

- the rotation space $SO(3)$ is parametrized in a solid radius-π ball
- the translation is assumed to be within the cube $[-\xi, \xi]^3$ (denoted as C_t)



(a) Rotation domain (b) Translation domain

Figure 5: $SE(3)$ space parametrization in GO-ICP

Bounding Functions i

For ease of manipulation, let's use the minimum cube $[\pi, \pi]^3$ that encloses the π-ball centered in $\boldsymbol{r}_0$ as C_r.

Theorem (Uncertainty radius)

Given a 3D point $\boldsymbol{p}$, a rotation cube C_r of half side-length σ_r with r_0 as the center and examining the maximum distance from $\boldsymbol{R_r p}$ to $\boldsymbol{R_{r_0} p}$, we have $\forall \boldsymbol{r} \in C_r$,

$$\|\boldsymbol{R}_r \boldsymbol{p} - \boldsymbol{R}_{r_0} \boldsymbol{p}\| \leq 2 \sin(\min(\sqrt{3}\sigma_r/2, \pi/2)) \, \|\boldsymbol{p}\| = \gamma_r$$

Or similarly, given a translation cube C_t with half side-length σ_t centered at $\boldsymbol{t_0}$, we have $\forall \boldsymbol{t} \in C_t$

$$\|(\boldsymbol{p} + \boldsymbol{t}) - (\boldsymbol{p} - \boldsymbol{t}_0)\| \leq \sqrt{3}\sigma_t = \gamma_t$$

BnB Bounds

For a given rotation cube C_r

$$\begin{aligned} \overline{E}_r &= \min_{\forall \boldsymbol{t} \in \mathcal{C}_t} \sum_i e_i(\boldsymbol{R}_{\boldsymbol{r}_0}, \boldsymbol{t})^2 \\ \underline{E}_r &= \min_{\forall \boldsymbol{t} \in \mathcal{C}_t} \sum_i \max\left(e_i(\boldsymbol{R}_{\boldsymbol{r}_0}, t) - \gamma_{r_i}, 0\right)^2 \end{aligned} \tag{3}$$

Similarly, for a given translation cube C_t

$$\begin{aligned} \overline{E}_t &= \sum_i \max\left(e_i(\boldsymbol{R}_{\boldsymbol{r}_0}, \boldsymbol{t_0}) - \gamma_{r_i}, 0\right)^2 \\ \underline{E}_t &= \sum_i \max\left(e_i(\boldsymbol{R}_{\boldsymbol{r}_0}, \boldsymbol{t_0}) - (\gamma_{r_i} + \gamma_t), 0\right)^2 \end{aligned} \tag{4}$$

where $e_i(\boldsymbol{R}, \boldsymbol{t}) = \|\boldsymbol{q}_{j^*} - \boldsymbol{R}\boldsymbol{p}_i\|$, with $\boldsymbol{q}_{j^*}$ denoted as optimal correspondence of $\boldsymbol{p}_i$; $\mathcal{C}_r$ and $\mathcal{C}_t$ are the initial cubes.

Go-ICP algorithm

Algorithm 2 Go-ICP – Outer BnB

Input: $\boldsymbol{P}$, $\boldsymbol{Q}$, threshold ϵ, initial cubes $\mathcal{C}_r$, $\mathcal{C}_t$

Output: Globally minimal error E^* and corresponding $\boldsymbol{r}^*$, $\boldsymbol{t}^*$

```
Put C_r into priority queue Q_r
Set E* = +∞
loop
    Read out a cube with lowest E_r from Q_r
    Quit if E* − E_r < ε
    Divide the cube into 8 sub-cubes
    for all sub-cube C_r do
        Compute Ē_r for C_r and corresponding optimal t by calling alg.3 with r_0, γ_r = 0 and E*
        if Ē_r < E* then
            Run ICP with initialization (r_0, t)
            Update E*, r* and t* with ICP results
        end if
        Compute E_r for C_r by calling alg.3 with r_0, γ_r and E*
        if E_r ≥ E* then
            Discard C_r and continue the loop
        end if
        Put C_r into Q_r
    end for
end loop
```

Algorithm 3 Go-ICP – Inner BnB

Input: $\boldsymbol{P}$, $\boldsymbol{Q}$, threshold ϵ, initial cube $\mathcal{C}_t$, rotation r_0, rotation uncertainty radii γ_r, so-far-best-error E^*

Output: Minimal error E_t^* and corresponding $\boldsymbol{t}^*$

```
Put C_t into priority queue Q_t
Set E_t* = E*
loop
    Read out a cube with lowest E_t from Q_t
    Quit the loop if E_t* − E_t < ε
    Divide the the cube into 8 sub-cubes
    for all sub-cube C_T do
        Compute Ē_t for C_t by 4 with r_0, t_0 and γ_r
        if Ē_t < E_t* then
            Update E_t* = Ē_t, t* = t_0
        end if
        Compute E_t for C_t by 4 with r_0, T_0 and γ_t
        if E_t ≥ E_t* then
            Discard C_t and continue the loop
        end if
        Put C_t into Q_r
    end for
end loop
```

Experiment i

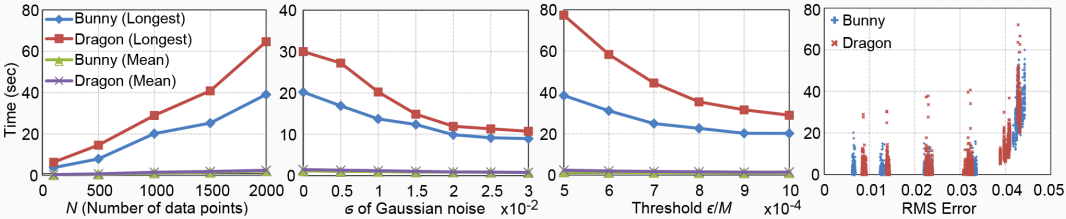


Figure 6: Running time of the Go-ICP method on the bunny and dragon point-sets with respect to different factors.

Experiment ii

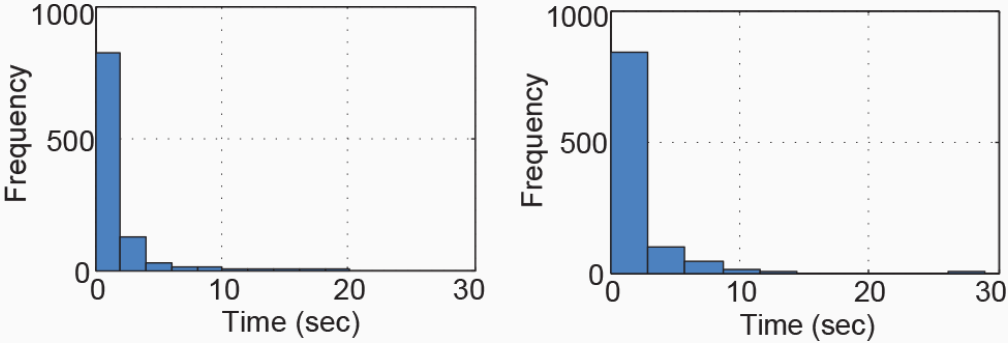


Figure 7: Running time histograms of Go-ICP for the bunny (left) and dragon (right) point-sets.

Go-ICP Trimming

In (Yang et al., 2016) a **trimmed** version of the algorithm is also proposed. Specifically, in each iteration, only a subset $\boldsymbol{S}$ of the data points are used for motion computation.

This is a strategy to obtain a more robust statistic by excluding some of the extreme values.

Considerations

- 👍 Finds the global optimal solution
- 👍 Defines upper and lower bound in domain regions
- 👍 Not constrained to ICP variants
- 👍 Parallelism can speed up computation a lot
- 👎 Limited to pt-to-pt distance
- 👎 Computes closest points set each iteration
- 👎 Computationally demanding

Generalized ICP (2009)

Derivation i

The Generalized-ICP (Segal et al., 2009) uses **probabilistic approach** to increase robustness by changing

$$\boldsymbol{T} \leftarrow \underset{T}{\operatorname{argmin}} \sum_i \omega_i \left\| \boldsymbol{m}_i - \boldsymbol{T}\boldsymbol{p}_i \right\|$$

in the basic ICP algorithm (alg. 1).

Derivation ii

For the purpose of this section let's assume that:

- $\boldsymbol{p}_i$ is a correspondence for $\boldsymbol{q}_i$ (and vice versa) - W.L.O.G.
- $\hat{\boldsymbol{P}} = \{\hat{\boldsymbol{p}}_i\}, \quad \boldsymbol{p}_i \sim \mathcal{N}(\hat{\boldsymbol{p}}_i, \boldsymbol{C}_i^P)$
- $\hat{\boldsymbol{Q}} = \{\hat{\boldsymbol{q}}_i\}, \quad \boldsymbol{q}_i \sim \mathcal{N}(\hat{\boldsymbol{q}}_i, \boldsymbol{C}_i^Q)$

where $\boldsymbol{C}_i^P$ and $\boldsymbol{C}_i^Q$ are covariance matrices associated with the measured points.

Derivation iii

We can use **Maximum likelihood estimation** to iteratively compute $\boldsymbol{T}$

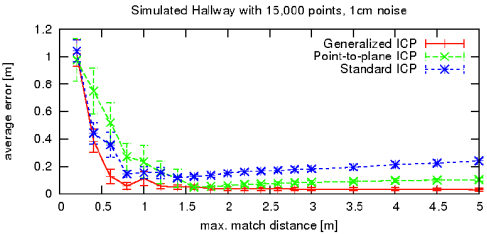
$$\boldsymbol{T} = \underset{\boldsymbol{T}}{\operatorname{argmax}} \prod_i p\big(\boldsymbol{q}_i - \boldsymbol{T}\boldsymbol{p}_i\big) = \underset{\boldsymbol{T}}{\operatorname{argmax}} \sum_i \log\big(p\big(\boldsymbol{q}_i - \boldsymbol{T}\boldsymbol{p}_i\big)\big)$$

that can be simplified to

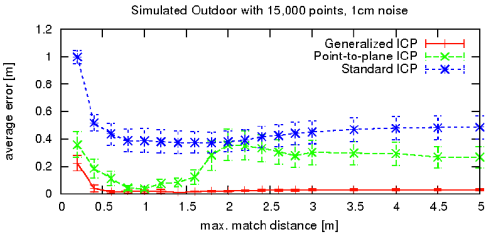
$$\boldsymbol{T} = \underset{\boldsymbol{T}}{\operatorname{argmin}} \sum_i (\boldsymbol{q}_i - \boldsymbol{T}\boldsymbol{p}_i)^T (\boldsymbol{C}_i^Q + \boldsymbol{T}\boldsymbol{C}_i^P\boldsymbol{T}^T)^{-1} (\boldsymbol{q}_i - \boldsymbol{T}\boldsymbol{p}_i) \tag{5}$$

This defines the **key step** of the Generalized-ICP algorithm (Mahalanobis distance).

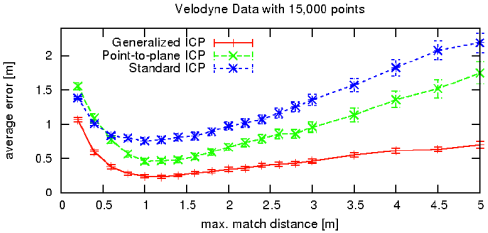
Results



(a) simulated indoor



(b) simulated outdoor



(c) Velodyne data

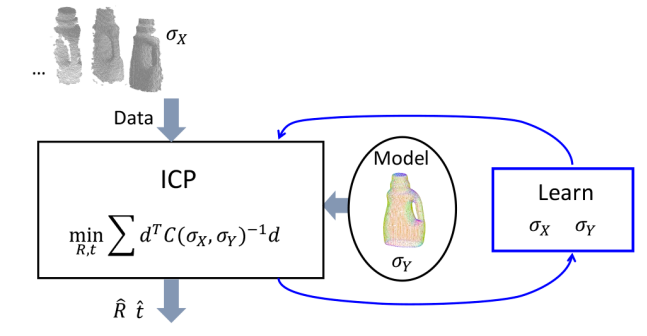
Figure 8: Average error

Considerations

- 👍 Probabilistic approach
- 👍 Covariance matrices give good flexibility
- 👎 Covariance matrices can be complex to compute
- 👎 Might get caught in local minima
- 👎 Still dependent on initial guess
- 👎 No closed form solution for $\boldsymbol{T}$ is available

Learning Anisotropic ICP (2016)

Principle



Idea: Learning Anisotropic ICP (Lee and Lee, 2016), assumes anisotropic Gaussian, and estimates the covariance. The learning scheme does not require manual tuning and the covariance is continually updated from observed data.

Pros and Cons

- 👍 Assumes anisotropic noise
- 👍 Reduces computational overhead due to covariance matrices
- 👎 Still local optimization method

Fast Global Registration (2016)

Let $\mathcal{K} = \{(p, q)\}$ be a set of **correspondences** collected by matching points from $\boldsymbol{P}$ and $\boldsymbol{Q}$.

We want to minimize:

$$E(\boldsymbol{T}) = \sum_{(p,q)\in\mathcal{K}} \rho(\|q_i - \boldsymbol{T}p_i\|) \tag{6}$$

where $\rho(\cdot)$ is a **robust penalty function**, i.e. scaled Geman-McClure estimator.

Scaled Geman-McClure estimator

Using this estimator small residuals are penalized in the LS sense:

$$\rho(x) = \frac{\mu x^2}{\mu + x^2} \tag{7}$$

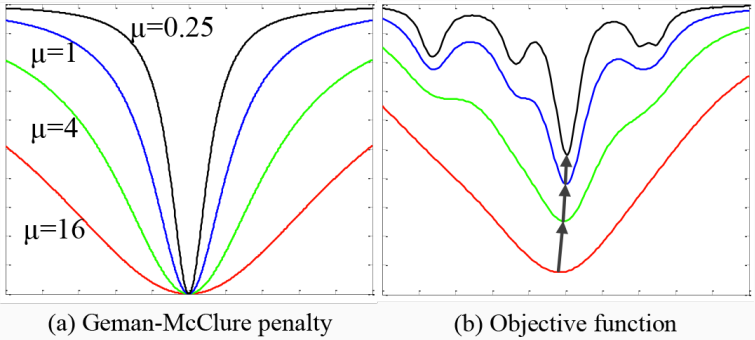


Figure 9: Geman-McClure estimator vs. Objective function (6)

Black-Ragarajan duality i

Objective function (6) is difficult to optimize directly, so the authors used **Black-Ragarajan duality** between line process and robust estimation (Black and Rangarajan, 1996).

Let's $\mathbb{L} = \{l_{(p,q)}\}$ a line process over the correspondences, we can write a new objective function optimized over $\boldsymbol{T}$ and $\mathbb{L}$.

$$E(\boldsymbol{T}, \mathbb{L}) = \sum_{(p,q)\in\mathcal{K}} \left(l_{(p,q)} \left\| p - \boldsymbol{T} q \right\|^2 + \Psi(l_{(p,q)}) \right) \tag{8}$$

where

$$\Psi(l_{(p,q)}) = \mu \left(\sqrt{l_{(p,q)}} - 1 \right)^2$$

Black-Ragarajan duality ii

Minimizing (8) over $\mathbb{L}$ we have:

$$\frac{\partial E}{\partial l_{(p,q)}} = \|p - \boldsymbol{T}q\| + \mu \frac{\sqrt{l_{(p,q)}} - 1}{\sqrt{l_{(p,q)}}} = 0$$

Solving for $l_{(p,q)}$ yields

$$l_{(p,q)} = \left(\frac{\mu}{\mu + \|p - \boldsymbol{T}q\|^2} \right)^2 \tag{9}$$

Substituting $l_{(p,q)}$ in (8) we obtain (6). Thus **optimizing objective (8) yields a solution $\boldsymbol{T}$ that is also optimal for the original objective (6)**.

What about the correspondences?

Fast Point Feature Histogram (FPFH) i

- The goal of FPFH is to encode a point's **k-neighborhood geometrical properties**.
- The complexity is $O(kN)$, where k is the number of neighbors for each point (Rusu et al., 2009)
- It is based on the **Simple Point Feature Histogram (SPFH)**

Fast Point Feature Histogram (FPFH) ii

Simple Point Feature Histogram (SPFH)

For each query point $\boldsymbol{p}_q$,

- Compute the k-neighborhood set P_k^q
- Compute the tuple (α, ϕ, θ) between $\boldsymbol{p}_q$ and each $\boldsymbol{p}_k \in P_k^q$

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{n}_q \\ \boldsymbol{v} &= \boldsymbol{u} \times \frac{(\boldsymbol{p}_k - \boldsymbol{p}_q)}{\|\boldsymbol{p}_k - \boldsymbol{p}_q\|} \\ \boldsymbol{w} &= \boldsymbol{u} \times \boldsymbol{v} \end{aligned} \quad \Rightarrow \quad \begin{aligned} \alpha &= \boldsymbol{v} \cdot \boldsymbol{n}_k \\ \phi &= \boldsymbol{u} \cdot \frac{(\boldsymbol{p}_k - \boldsymbol{p}_q)}{\|\boldsymbol{p}_k - \boldsymbol{p}_q\|_2} \\ \theta &= \arctan(\boldsymbol{w} \cdot \boldsymbol{n}_q, \boldsymbol{u} \cdot \boldsymbol{n}_k) \end{aligned} \tag{10}$$

- Bin all (α, ϕ, θ) into a histogram

Fast Point Feature Histogram (FPFH) iii

To compute the FPFH histogram feature:

- for each query point $\boldsymbol{p}_q$ compute the SPFH
- for each point its k neighbors are re-determined, and the neighboring SPFH values are used to weight the final histogram of $\boldsymbol{p}_q$ (called FPFH) as in eq.11

$$\text{FPFH}(\boldsymbol{p}_q) = \text{SPFH}(\boldsymbol{p}_q) + \frac{1}{k}\sum_{i=1}^{k}\frac{1}{\omega_k}\cdot\text{SPFH}(\boldsymbol{p}_k) \tag{11}$$

where the weight ω_k represents a distance between the query point $\boldsymbol{p}_q$ and a neighbor point $\boldsymbol{p}_k$ in some given metric space.

Correspondences set i

The correspondence set is build as following, let

- $F(\boldsymbol{P}) = \{F(\boldsymbol{p}) : \boldsymbol{p} \in \boldsymbol{P}\}$ the FPFH of points in $\boldsymbol{P}$
- $F(\boldsymbol{Q}) = \{F(\boldsymbol{q}) : \boldsymbol{q} \in \boldsymbol{Q}\}$ the FPFH of points in $\boldsymbol{Q}$

$\mathcal{K}_1$ is the set containing the nearest neighbor of $F(\boldsymbol{p})$ among $F(\boldsymbol{Q})$, and vice versa.

Correspondences set ii

We can improve the correspondences set by applying:

Reciprocity test A correspondence pair $(\boldsymbol{p}, \boldsymbol{q})$ is selected from $\mathcal{K}_1$ if and only if $F(\boldsymbol{p})$ is the nearest neighbor of $F(\boldsymbol{q})$ among $F(\boldsymbol{P})$ and $F(\boldsymbol{q})$ is the nearest neighbor of $F(\boldsymbol{p})$ among $F(\boldsymbol{Q})$. The resulting correspondence set is denoted by $\mathcal{K}_2$

Tuple test Randomly pick 3 correspondence pairs $(\boldsymbol{p_1}, \boldsymbol{q_1})$, $(\boldsymbol{p_2}, \boldsymbol{q_2})$, $(\boldsymbol{p_3}, \boldsymbol{q_3})$ from $\mathcal{K}_2$ and check if the tuples $(\boldsymbol{p_1}, \boldsymbol{p_2}, \boldsymbol{p_3})$ and $(\boldsymbol{q_1}, \boldsymbol{q_2}, \boldsymbol{q_3})$ are compatible

$$\forall i \neq j, \quad \tau \leq \frac{\|p_i - p_j\|}{\|q_i - q_j\|} \leq \frac{1}{\tau}, \quad \tau = 0.9$$

This is the set used by the algorithm $\mathcal{K} = \mathcal{K}_3$

Fast Global Registration Algorithm

Let D be the diameter of the largest surface and δ the distance threshold for genuine correspondence

Algorithm 4 Fast Global Registration

```
Compute normals {n_p} and {n_q}
Compute F(P) and F(Q)
Compute K = K_3
T ← I,  μ ← D^2
while not converged or μ > δ^2 do
    for (p, q) ∈ K do
        Compute l_(p,q) using equation 9
    end for
    Compute T (closed form (Horn, 1987))
    Every four iterations μ ← μ/2
end while
```

Performance i

	$\sigma = 0$		$\sigma = 0.0025$		$\sigma = 0.005$	
	Average RMSE	Maximal RMSE	Average RMSE	Maximal RMSE	Average RMSE	Maximal RMSE
GoICP [42]	0.029	0.130	0.032	0.133	0.037	0.127
GoICP-Trimming [42]	0.035	0.473	0.039	0.475	0.044	0.478
Super 4PCS [26]	0.012	0.019	0.014	0.029	0.017	0.095
OpenCV [8]	0.009	0.013	0.018	0.212	0.032	0.242
PCL [34, 19]	**0.003**	**0.005**	0.009	0.061	0.111	0.414
CZK [7]	**0.003**	**0.005**	0.008	0.022	0.035	0.274
Our approach	**0.003**	**0.005**	**0.006**	**0.011**	**0.008**	**0.017**

Figure 10: Average and maximal RMSE achieved by global registration algorithms on synthetic range images with noise level σ

Performance ii

	Average # of points	GoICP [42]	GoICP-Trimming [42]	OpenCV [8]	Super 4PCS [26]	PCL [34, 19]	CZK [7]	Our approach
Bimba	9,416	19.3	19.4	41.0	311.4	18.2	12.8	**0.13**
Children	11,148	21.0	19.2	136.3	238.2	4.8	6.6	**0.20**
Dragon	11,232	94.1	38.4	57.7	483.7	8.6	11.9	**0.23**
Angel	12,072	21.0	20.4	80.9	171.5	8.7	11.3	**0.26**
Bunny	13,357	74.7	72.4	12.3	283.8	55.6	12.7	**0.28**
Average	11,445	46.0	34.0	65.6	297.7	19.2	11.1	**0.22**

Figure 11: Running times of global registration methods, measured in seconds

Performance iii

	Average # of points	PCL ICP point-to-point	PCL ICP point-to-plane	Sparse ICP point-to-point [5]	Sparse ICP point-to-plane [5]	Our approach
Bimba	9,416	0.73	0.31	3.1	11.8	**0.13**
Children	11,148	0.75	0.46	3.9	15.0	**0.20**
Dragon	11,232	0.99	0.47	3.6	13.8	**0.23**
Angel	12,072	0.81	1.01	4.9	18.5	**0.26**
Bunny	13,357	2.10	1.70	9.2	10.3	**0.28**
Average	11,445	1.08	0.79	4.9	13.9	**0.22**

Figure 12: Timing comparison with local algorithms, measured in seconds

Pros and Cons

- 👍 Global optimization
- 👍 One order of magnitude faster
- 👍 Does not require initialization
- 👍 Correspondences computed only once but their weight change
- 👎 No proof of global convergence in the paper

References

- Arun, K Somani, Thomas S Huang, and Steven D Blostein (1987). “Least-squares fitting of two 3-D point sets”. In: *IEEE Transactions on pattern analysis and machine intelligence* 5, pp. 698–700.
- Bellekens, Ben, Vincent Spruyt, Rafael Berkvens, and Maarten Weyn (2014). “A survey of rigid 3D pointcloud registration algorithms”. In: *AMBIENT 2014: the Fourth International Conference on Ambient Computing, Applications, Services and Technologies, August 24-28, 2014, Rome, Italy*, pp. 8–13.
- Black, Michael J and Anand Rangarajan (1996). “On the unification of line processes, outlier rejection, and robust statistics with applications in early vision”. In: *International Journal of Computer Vision* 19.1, pp. 57–91.

References ii

- Horn, Berthold KP (1987). “Closed-form solution of absolute orientation using unit quaternions”. In: *JOSA A* 4.4, pp. 629–642.
- Horn, Berthold KP, Hugh M Hilden, and Shahriar Negahdaripour (1988). “Closed-form solution of absolute orientation using orthonormal matrices”. In: *JOSA A* 5.7, pp. 1127–1135.
- Lee, Bhoram and Daniel D Lee (2016). “Learning anisotropic ICP (LA-ICP) for robust and efficient 3D registration”. In: *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, pp. 5040–5045.
- Olsson, Carl, Fredrik Kahl, and Magnus Oskarsson (2009). “Branch-and-bound methods for euclidean registration problems”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.5, pp. 783–794.

References iii

- Rusu, Radu Bogdan, Nico Blodow, and Michael Beetz (2009). “Fast point feature histograms (FPFH) for 3D registration”. In: *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*. IEEE, pp. 3212–3217.
- Schönemann, Peter H (1966). “A generalized solution of the orthogonal Procrustes problem”. In: *Psychometrika* 31.1, pp. 1–10.
- Segal, Aleksandr, Dirk Haehnel, and Sebastian Thrun (2009). “Generalized-ICP.”. In: *Robotics: science and systems*. Vol. 2. 4, p. 435.
- Umeyama, Shinji (1991). “Least-squares estimation of transformation parameters between two point patterns”. In: *IEEE Transactions on pattern analysis and machine intelligence* 13.4, pp. 376–380.

References iv

Yang, J., H. Li, D. Campbell, and Y. Jia (2016). “Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.11, pp. 2241–2254. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2015.2513405.