**Supplemental Material for**

# Accurate and Robust Registration of Low Overlapping Point Clouds

Jieyin Yang[a,b], Mingyang Zhao[c,d], Xiaohong Jia[a,b], and Dong-Ming Yan[b,d]

[a]KLMM, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, China
[b]University of Chinese Academy of Sciences, Beijing, China
[c]Beijing Academy of Artificial Intelligence
[d]State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS) & NLPR, Institute of Automation, Chinese Academy of Sciences

In this document, we provide additional details, explanations, and experiments to support the original paper. Below is a summary of the contents:

- Approximation of the posterior distribution and final likelihood of the joint distribution of observed variable $\mathbf{y}$ and hidden state $\mathbf{z}$.
- Theoretical deduction of concrete steps of EM algorithm and the specific expression of involved parameters.
- Theoretical proof of the proposed proposition that two points lie in the same cylinder if and only if their symmetric function Eq. 15 equals to zero.
- Detailed explanations of our proposed point-pair sampling method and the corresponding pseudo-code of the algorithm.
- Experimental details of competitors along with their parameter settings in our experiments and more qualitative results of different overlapping rates on TUM RGB-D and KITTI datasets.

## 1. Distribution Approximation and the Final Likelihood Function

To simplify the computation, we use the mean field to approximate the posterior distribution $P_G(\mathbf{z}|\beta)$:

$$P_G(\mathbf{z}|\beta) \approx P^{mf}(\mathbf{z}|\beta) = \prod_i P_i^{mf}(z_i|\beta, \tilde{\mathbf{z}}), \tag{1}$$

where $\tilde{\mathbf{z}} = \mathbb{E}[P_G(\mathbf{z}|\beta)]$ is the expectation of $\mathbf{z}$ under the parameter $\beta$ and each component of the above equation only relies on the local information:

$$P_i^{mf}(z_i|\beta, \tilde{\mathbf{z}}) = \frac{\exp(\beta \sum_{i' \sim i} w_{i,i'} z_i \tilde{z}_{i'})}{\exp(\beta \sum_{i' \sim i} w_{i,i'}(+1)\tilde{z}_{i'}) + \exp(\beta \sum_{i' \sim i} w_{i,i'}(-1)\tilde{z}_{i'})}. \tag{2}$$

As the observation $\mathbf{y}$ is conditionally independent given $\mathbf{z}$ in the hidden Markov model, then the conditional density function $f$ has the following form:

$$f(\mathbf{y}|\mathbf{z}, \theta) = \prod_i (N(\mathbf{y}_i|\mu_{+1}, \Sigma_{+1}))^{(1+z_i)/2} (N(\mathbf{y}_i|\mu_{-1}, \Sigma_{-1}))^{(1-z_i)/2}, \tag{3}$$

where $N$ represents the density function of the k-dimensional normal distribution with parameters $\mu_{+1}, \Sigma_{+1}$ (inliers) or $\mu_{-1}, \Sigma_{-1}$ (outliers):

$$N(\mathbf{y}_i|\mu_{+1}, \Sigma_{+1}) = \frac{1}{(2\pi)^{k/2}\sqrt{\det \Sigma_{+1}}} \exp^{\{\frac{1}{2}(\mathbf{y}_i - \mu_{+1})^T \Sigma_{+1}^{-1} (\mathbf{y}_i - \mu_{+1})\}}. \tag{4}$$

We adopt $k = 4$ in our case. We unify these parameters by $\theta$. Hence the likelihood of the joint distribution becomes

$$\begin{aligned} P^{mf}(\mathbf{y}, \mathbf{z}|\beta, \theta, \tilde{\mathbf{z}}) &= f(\mathbf{y}|\mathbf{z}, \theta) P^{mf}(\mathbf{z}|\beta) = \\ \prod_i P_i^{mf}(z_i|\beta, \tilde{z}) &(N(\mathbf{y}_i|\mu_{+1}, \Sigma_{+1}))^{(1+z_i)/2} \\ &\cdot (N(\mathbf{y}_i|\mu_{-1}, \Sigma_{-1}))^{(1-z_i)/2}. \end{aligned} \tag{5}$$

## 2. EM Algorithm

We use EM algorithm to calculate the estimated hidden state iteratively. The specific iterative steps are listed as follows.

(1) E-step: We replace the hidden state $z_i$ with $\mathbb{E}[z_i|\mathbf{y}_i, \tilde{z}, \beta, \theta)]$, where

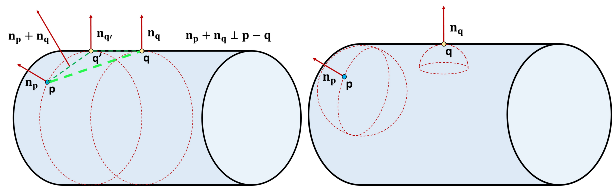



Figure 1. Left: Points $\mathbf{p}$ and $\mathbf{q}$ together with their normals lie in the same cylinder, hence satisfying $(\mathbf{p}-\mathbf{q}) \perp (\mathbf{n}_p + \mathbf{n}_q)$ Right: The situation that $\mathbf{p}$ and $\mathbf{q}$ have different curvatures. In this situation $\mathbf{p}$ and $\mathbf{q}$ are not the right corresponding points but they are without punishment in Eq. 15.

$$\begin{aligned}&\mathbb{E}[z_i|\mathbf{y}_i, \tilde{z}, \beta, \theta)] = \\ &P(z_i = 1|\mathbf{y}_i, \tilde{z}, \beta, \theta) - P(z_i = -1|\mathbf{y}_i, \tilde{z}, \beta, \theta),\end{aligned} \tag{6}$$

$$\begin{aligned}&P(z_i = 1|\mathbf{y}_i, \tilde{z}, \beta, \theta) \propto \\ &\exp(\beta \sum_{i' \sim i} w_{i,i'} \tilde{z}_{i'}) \frac{1}{(2\pi)^{k/2}\sqrt{\det \Sigma_{+1}}} \\ &\cdot \exp^{\{\frac{1}{2}(\mathbf{y}_i - \mu_{+1})^T \Sigma_{+1}^{-1}(\mathbf{y}_i - \mu_{+1})\}},\end{aligned} \tag{7}$$

$$\begin{aligned}&P(z_i = -1|\mathbf{y}_i, \tilde{z}, \beta, \theta) \propto \\ &\exp(-\beta \sum_{i' \sim i} w_{i,i'} \tilde{z}_{i'}) \frac{1}{(2\pi)^{k/2}\sqrt{\det \Sigma_{-1}}} \\ &\cdot \exp^{\{\frac{1}{2}(\mathbf{y}_i - \mu_{-1})^T \Sigma_{-1}^{-1}(\mathbf{y}_i - \mu_{-1})\}}.\end{aligned} \tag{8}$$

(2) M-step: We use the estimated hidden state to update the parameters. The results are as follows.
For inliers,

$$n_{+1} = \sum_i \frac{1+z_i}{2}, \tag{9}$$

$$\mu_{+1} = \frac{1}{n_{+1}} \sum_i \frac{1+z_i}{2} \mathbf{y}_i, \tag{10}$$

$$\Sigma_{+1} = \frac{1}{n_{+1}} \sum_i \frac{1+z_i}{2} (\mathbf{y}_i - \mu_{+1})(\mathbf{y}_i - \mu_{+1})^T. \tag{11}$$

For outliers,

$$n_{-1} = \sum_i \frac{1-z_i}{2}, \tag{12}$$

$$\mu_{-1} = \frac{1}{n_{-1}} \sum_i \frac{1-z_i}{2} \mathbf{y}_i, \tag{13}$$

$$\Sigma_{-1} = \frac{1}{n_{-1}} \sum_i \frac{1-z_i}{2} (\mathbf{y}_i - \mu_{-1})(\mathbf{y}_i - \mu_{-1})^T. \tag{14}$$

## 3. Theoretical Proof of Proposed Proposition

The symmetric function of point cloud registration is

$$(\mathbf{p}-\mathbf{q}) \cdot (\mathbf{n}_p + \mathbf{n}_q), \tag{15}$$

**Proposition 1.** *Points $\mathbf{p}$ and $\mathbf{q}$ together with their normals $\mathbf{n}_p$ and $\mathbf{n}_q$ lie in the same cylinder, if and only if Eq. 15 equals zero.*

*Proof.* " $\Leftarrow$ ": (1) If $\mathbf{n}_p = \mathbf{n}_q$, we get $\mathbf{n}_p \perp (\mathbf{p}-\mathbf{q})$ and $n_q \perp (\mathbf{p}-\mathbf{q})$. In this situation, two points lie on the cylinder that takes $(\mathbf{p}-\mathbf{q})$ as one of the generatrix.

(2) If $\mathbf{n}_p \neq \mathbf{n}_q$, then $\mathbf{n}_p$ and $\mathbf{n}_q$ can generate a plane $\mathcal{S}$ with $\mathbf{p}$ on it. Then we have $(\mathbf{n}_p \times \mathbf{n}_q) \perp \mathcal{S}$. Translate $\mathbf{q}$ along the direction of $\mathbf{n}_p \times \mathbf{n}_q$ (or its negative direction) until it is on plane $\mathcal{S}$. We call the intersection point as $\mathbf{q}'$ and its normal as $\mathbf{n}_{q'}$, where $\mathbf{n}_{q'}$=$\mathbf{n}_q$ . Since $\mathbf{p}$ and $\mathbf{q}'$ lie on the same plane $\mathcal{S}$ with their normals satisfying

$$(\mathbf{p}-\mathbf{q}') \cdot (\mathbf{n}_p + \mathbf{n}_{q'}) = 0, \tag{16}$$

we can conclude that $\mathbf{p}$ and $\mathbf{q}'$ together with their normals $\mathbf{n}_p$ and $\mathbf{n}_{q'}$ lay in the same circle. Then we generate the cylinder based on this circle and the axial direction $\mathbf{n}_p \times \mathbf{n}_q$. According to our construction of cylinder, $\mathbf{q}$ and $\mathbf{n}_q$ also lies on it.

" $\Rightarrow$ ": If $\mathbf{p}$, $\mathbf{q}$ and their normals are on the same cylinder, this situation is illustrated in Fig. 1. Let $\mathbf{q}'$ be a point on the same circle as $\mathbf{p}$ and its normal direction $\mathbf{n}_q'$ is equal to $\mathbf{n}_q$. Since $(\mathbf{n}_p + \mathbf{n}_q) \perp (\mathbf{p}-\mathbf{q}')$ and $(\mathbf{n}_p + \mathbf{n}_q) \perp (\mathbf{q}-\mathbf{q}')$, we can get $(\mathbf{n}_p + \mathbf{n}_q) \perp (\mathbf{p}-\mathbf{q})$, which means Eq. 15 equals to zero. $\square$

## 4. Point-Pair Sampling Method

According to our sampling method, we aim to sample the point pairs for transformation stability. For sake of that, the unconstrained direction should be avoided. So we sample the point pairs to let the condition number $c = \frac{\lambda_1}{\lambda_6}$ of $\mathbf{C}$ be as close to one as possible.

Specifically, we define $\mathbf{v}_i = [(\tilde{\mathbf{p}}_i + \tilde{\mathbf{q}}_i) \times \mathbf{n}_i, \mathbf{n}_i]$, $i = 1, 2, \cdots, n$. Then we sort $\mathbf{v}_i$ by its projection value on the eigenvalue $\mathbf{x}_k$ which determines the constraint of $\mathbf{v}_i$ in the direction $\mathbf{x}_k$, and record it in the list $\mathbf{L}_k$, $k = 1, 2, \cdots, 6$. After modeling the sorted lists $\mathbf{L}_1, \cdots, \mathbf{L}_6$, we randomly choose a point pair $(\hat{\mathbf{p}}_1, \hat{\mathbf{q}}_1)$, and let $\hat{\mathbf{v}}_1 = [(\hat{\mathbf{p}}_i + \hat{\mathbf{q}}_i) \times \mathbf{n}_i, \mathbf{n}_i]$. We then initialize $t_1, t_2, \cdots, t_6$ as $t_1 = (\hat{\mathbf{v}}_1 \cdot \mathbf{x}_1)^2, t_2 = (\hat{\mathbf{v}}_1 \cdot \mathbf{x}_2)^2, \cdots, t_6 = (\hat{\mathbf{v}}_1 \cdot \mathbf{x}_6)^2$, and delete $\hat{\mathbf{v}}_1$ from the sorted lists $\mathbf{L}_1, \cdots, \mathbf{L}_6$. After initialization, each time we select the point pair from the top of a certain list based on the minimal value of $t_1, \cdots, t_6$, which can be viewed as the current estimation of the eigenvalues. We then update the value of $t_1, \cdots, t_6$ as $t_1 = t_1 + (\hat{\mathbf{v}}_j \cdot \mathbf{x}_1)^2, t_2 = t_2 + (\hat{\mathbf{v}}_j \cdot \mathbf{x}_2)^2, \cdots, t_6 = t_6 + (\hat{\mathbf{v}}_j \cdot \mathbf{x}_6)^2$. The whole procedure is summarized in Algorithm 1.

## 5. Experimental Settings, Details, and More Qualitative Registration Results

The competitors along with their parameters used in our experiments are listed in Table 1.

Table 1. The compared methods and their parameter settings in experiments.

| Methods | Parameters | Implementations |
|---|---|---|
| P2P-ICP [1] | Metric: pointToPoint; MaxIterations: 200<br>motion: rigid3d | MATLAB code<br>https://ww2.mathworks.cn/products/matlab.html |
| P2N-ICP [3] | Metric: pointToPlane; MaxIterations: 200;<br>motion: rigid3d | MATLAB code<br>https://ww2.mathworks.cn/products/matlab.html |
| GMM [4] | Downsample: 0.2; motion: rigid3d | MATLAB code<br>https://github.com/bing-jian/gmmreg |
| CPD [5] | Downsample: 0.2; Noise Weight: 0.5;<br>motion: rigid3d | MATLAB code<br>www.bme.ogi.edu/~myron/matlab/cpd |
| G-ICP [6] | Maximum Correspondence Distance: 0.07;<br>MaxIterations: 100 | C++ code<br>https://github.com/isl-org/Open3D |
| FGR [9] | Maximum Correspondence Distance: 0.025;<br>Annealing Rate: 1.4 | C++ code<br>https://github.com/isl-org/Open3D |
| S-ICP [2] | Norm: 0.4; MaxIterations: 100; Optimization: ADMM;<br>Penalty Increase Factor: 1.2 | C++ code<br>https://github.com/OpenGP/sparseicp |
| FR-ICP [8] | Robust Function: WELSCH; para: 0.1; use AA?: true;<br>MaxIterations: 100 | C++ code<br>https://github.com/yaoyx689/Fast-Robust-ICP |
| HMRF-ICP [7] | EM inlier: Gaussian; EM outlier: logistic;<br>MaxIterations: 200; EM-MaxIterations: 150 | MATLAB code<br>https://github.com/JStech/ICP |
| Ours | EM inlier: Gaussian; EM outlier: Gaussian;<br>MaxIterations: 200; EM-MaxIterations: 150 | MATLAB code<br>Our code will be publicly available online |

In our low overlapping tests, four representative methods, namely Sparse ICP, Fast and Robust ICP, HMRF-ICP and Ours, are chosen to analyze the influence of different overlapping rates on registration results. Fig. 2 shows more qualitative registration results with different overlapping rates.

In our sequence point cloud test, we choose KITTI sequence dataset and align the point clouds in the sequence whose index intervals are 1, 4 and 8. More experimental results when intervals equal to 1 and 4 are presented in Fig. 3.

## References


[1] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992. 3

[2] S. Bouaziz, A. Tagliasacchi, and M. Pauly. Sparse iterative closest point. In *Computer graphics forum*, volume 32, pages 113–123. Wiley Online Library, 2013. 3

[3] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992. 3

[4] B. Jian and B. C. Vemuri. Robust point set registration using gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1633–1645, 2010. 3

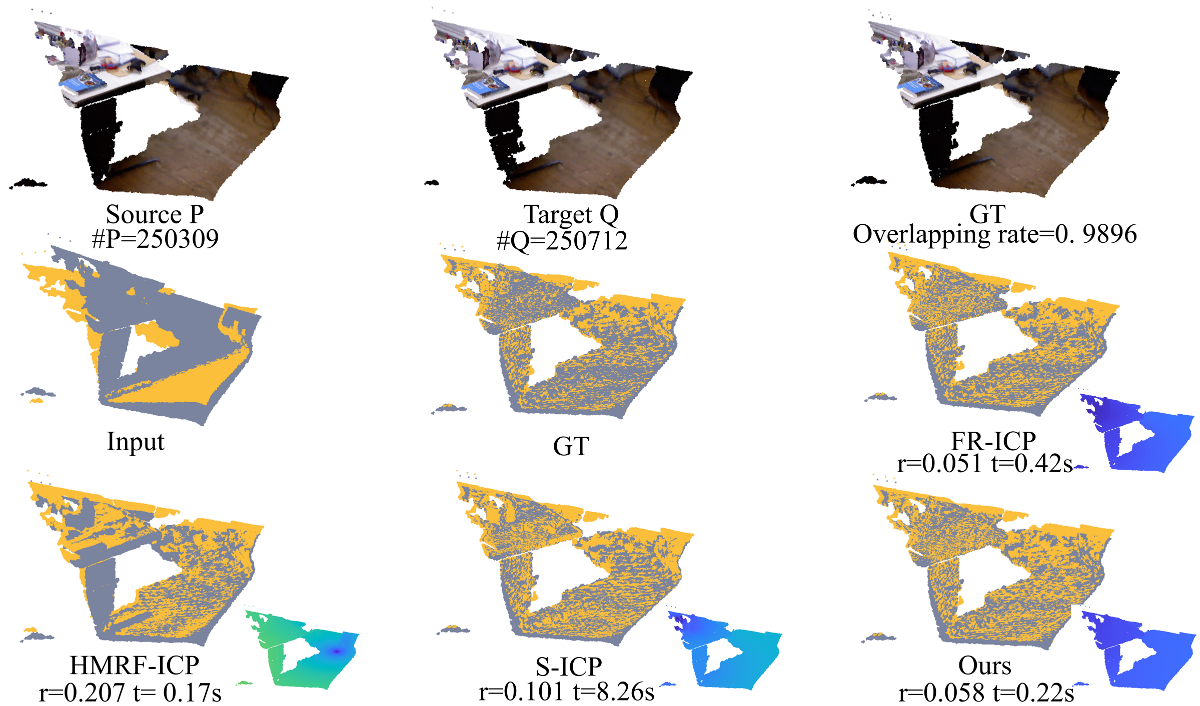
[5] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010. 3

[6] A. Segal, D. Haehnel, and S. Thrun. Generalized-icp. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009. 3

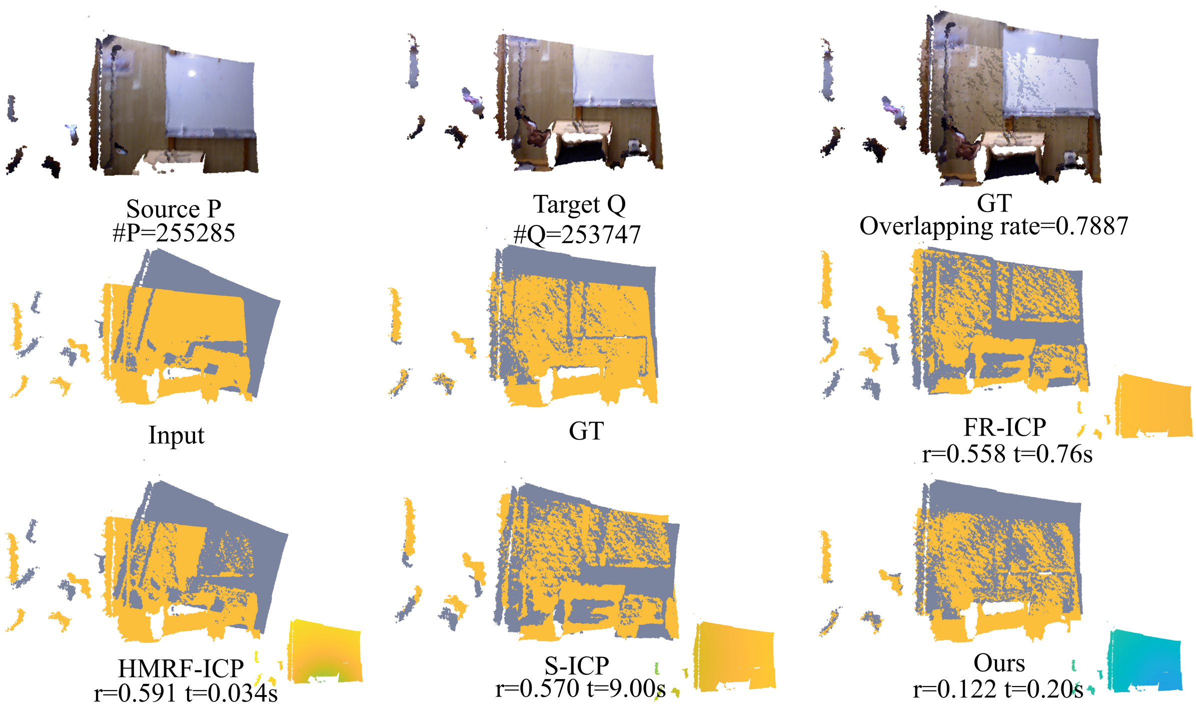
[7] J. Stechschulte, N. Ahmed, and C. Heckman. Robust low-overlap 3-d point cloud registration for outlier rejection. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7143–7149, 2019. 3

[8] J. Zhang, Y. Yao, and B. Deng. Fast and robust iterative closest point. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 3

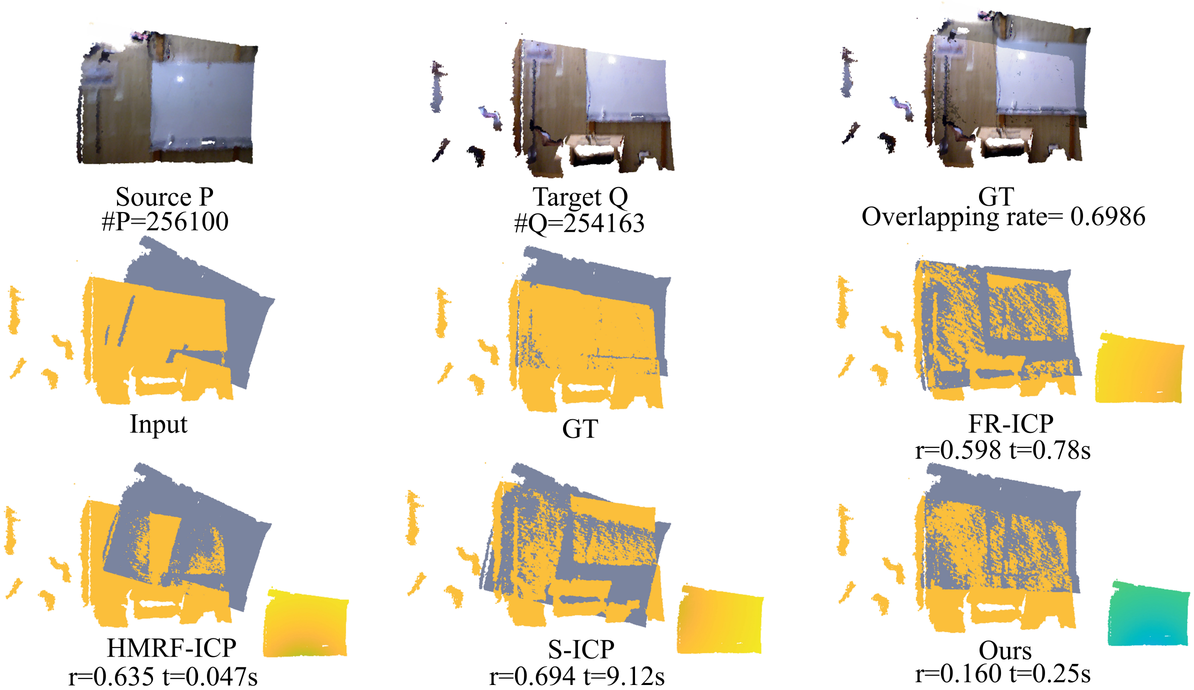
[9] Q.-Y. Zhou, J. Park, and V. Koltun. Fast global registration. In *European conference on computer vision*, pages 766–782, 2016. 3



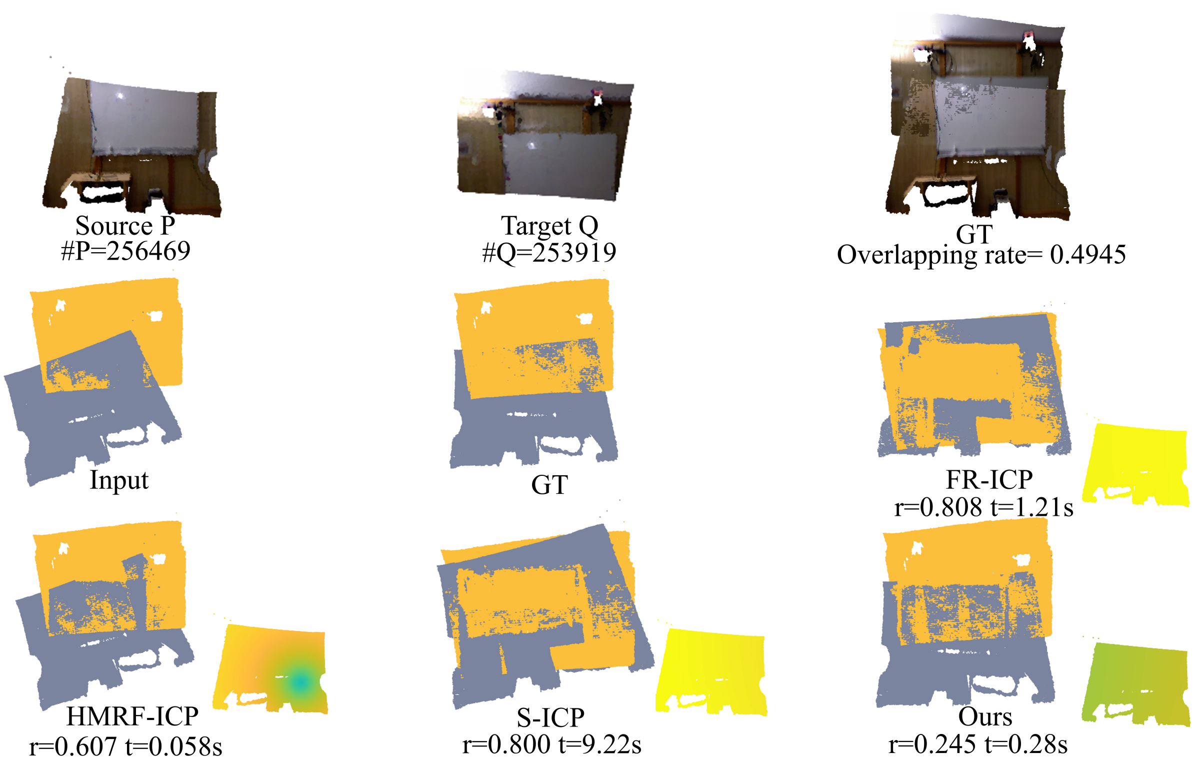

(a) Overlapping rate=0.9896




(b) Overlapping rate=0.7887

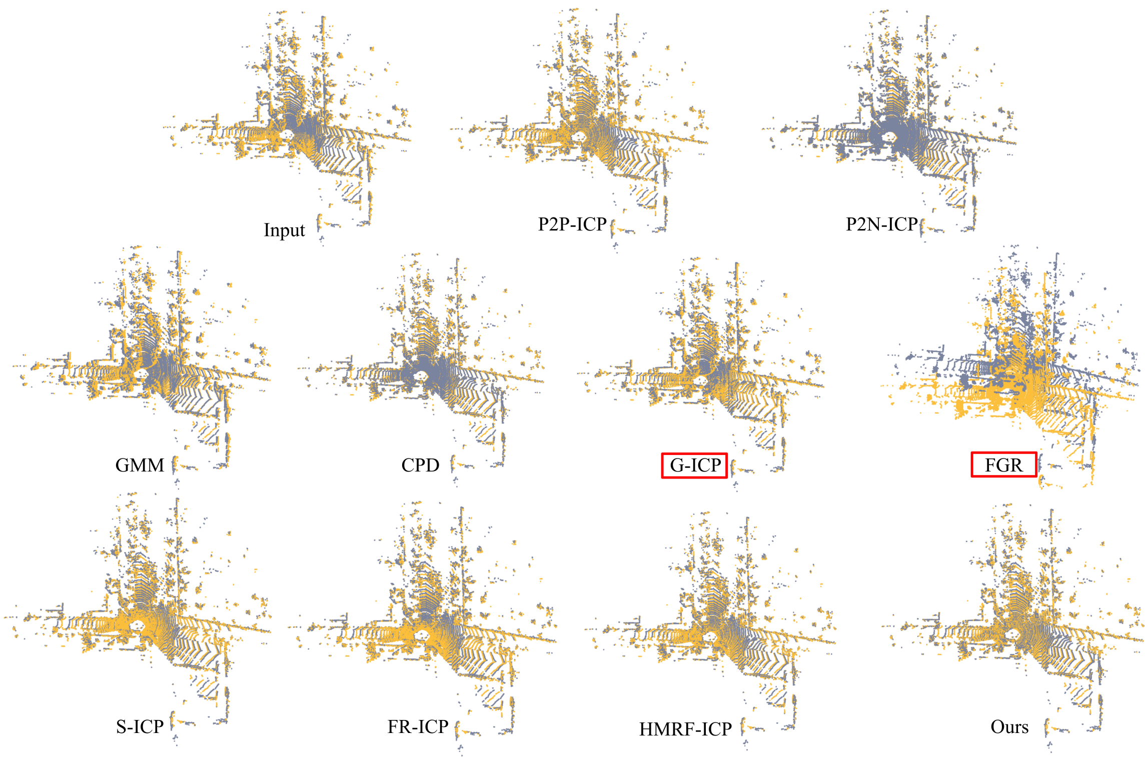



(c) Overlapping rate=0.6986

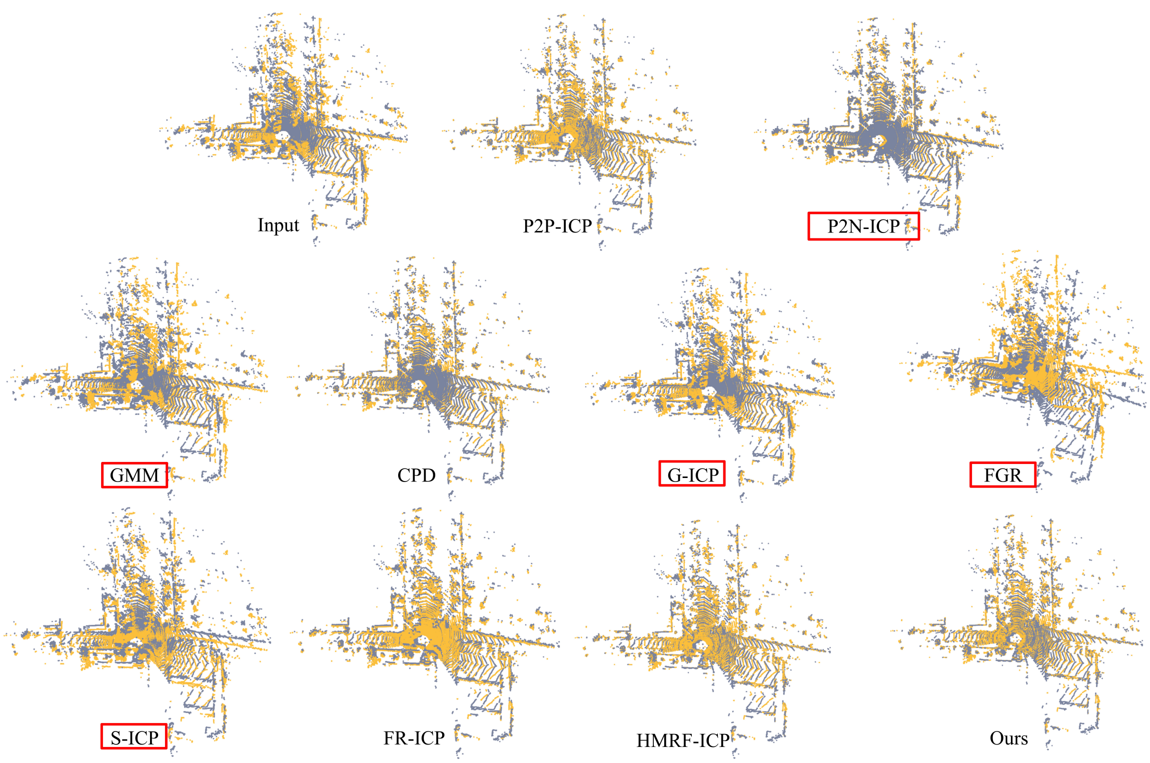



(d) Overlapping rate=0.4945

Figure 2. More qualitative registration results under different overlapping rates. The first row illustrates the original point clouds attained by RGB image and depth map. The second row and the third row show the registration results of different methods and their log-scale color coding. Our method has the most reliable results under a set of low overlapping rates.




(a) Index interval = 1




(b) Index interval = 4

Figure 3. Registration results of the KITTI dataset with the index interval equal to 1, 4. Red boxes indicate methods that have relatively large deviations.

**Algorithm 1: The Proposed Geometrically Stable Point-Pair Sampling**

**Input:** Corresponding point pairs $(\mathbf{p}_i, \mathbf{q}_i)$ together with the normal pairs $(\mathbf{n}_{p,i}, \mathbf{n}_{q,i})$, $i = 1, 2, \cdots, n$; Desired number of point pairs $m$;

**Output:** Chosen point pairs with their normal $(\hat{\mathbf{p}}_i, \hat{\mathbf{q}}_i, \mathbf{n}_{\hat{\mathbf{p}}_i}, \mathbf{n}_{\hat{\mathbf{q}}_i})$, $i = 1, 2, \cdots, m$.

Form the covariance matrix $\mathbf{C}$ and then perform eigenvalue decomposition $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_6$; The corresponding eigenvectors are denoted as $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_6$; Define $\mathbf{v}_i = [(\tilde{\mathbf{p}}_i + \tilde{\mathbf{q}}_i) \times \mathbf{n}_i, \mathbf{n}_i]$, $i = 1, 2, \cdots, n$.

**for** $k = 1$ *to* $6$ **do**

  $\mathbf{L}_k = [\mathbf{v}_{\sigma_k(1)}, \cdots, \mathbf{v}_{\sigma_k(n)}]$ where $(\mathbf{v}_{\sigma_k(1)} \cdot \mathbf{x}_k)^2 \geq \cdots \geq (\mathbf{v}_{\sigma_k(n)} \cdot \mathbf{x}_k)^2$

**end**

Randomly choose a point pair $(\hat{\mathbf{p}}_1, \hat{\mathbf{q}}_1)$, $\hat{\mathbf{v}}_1 = [(\hat{\mathbf{p}}_i + \hat{\mathbf{q}}_i) \times \mathbf{n}_i, \mathbf{n}_i]$ let $t_1 = (\hat{\mathbf{v}}_1 \cdot \mathbf{x}_1)^2, t_2 = (\hat{\mathbf{v}}_1 \cdot \mathbf{x}_2)^2, \cdots, t_6 = (\hat{\mathbf{v}}_1 \cdot \mathbf{x}_6)^2$. Delete $\hat{\mathbf{v}}_1$ from the sorted lists $\mathbf{L}_1, \cdots, \mathbf{L}_6$.

**for** $j = 2$ *to* $m$ **do**

  $t_s = \min_{i=1,\cdots,6} t_i$.

  Find $\hat{\mathbf{v}}_j$ from the top of the sorted list $\mathbf{L}_s$ and choose the corresponding point pair $(\hat{\mathbf{p}}_j, \hat{\mathbf{q}}_j)$.

  $t_1 = t_1 + (\hat{\mathbf{v}}_j \cdot \mathbf{x}_1)^2, t_2 = t_2 + (\hat{\mathbf{v}}_j \cdot \mathbf{x}_2)^2, \cdots, t_6 = t_6 + (\hat{\mathbf{v}}_j \cdot \mathbf{x}_6)^2$.

  Delete $\hat{\mathbf{v}}_j$ from $\mathbf{L}_1, \cdots, \mathbf{L}_6$.

**end**

**return** the chosen point pairs with their normal $(\hat{\mathbf{p}}_i, \hat{\mathbf{q}}_i, \mathbf{n}_{\hat{\mathbf{p}}_i}, \mathbf{n}_{\hat{\mathbf{q}}_i})$, $i = 1, 2, \cdots, m$.