7. Details of Method

7.1. Transformation Clustering

For each hypothesis $\mathbf{h}_i = (\mathbf{R}_i, \mathbf{t}_i)$, where rotation matrix $\mathbf{R}_i \in SO(3)$ and translation vector $\mathbf{t}_i \in \mathbb{R}^3$, we convert \mathbf{R}_i to an Euler angle vector $\mathbf{r}_i = (\alpha, \beta, \gamma)^T$. Given a rotation matrix

$$\mathbf{R}_{i} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix},$$
(5)

$$\mathbf{r}_{i} = (\alpha, \beta, \gamma)^{\mathrm{T}} = (\tan^{-1}(\frac{r_{32}}{r_{33}}), -\sin^{-1}(r_{31}), \tan^{-1}(\frac{r_{21}}{r_{11}}))^{\mathrm{T}}.$$
 (6)

Since the converted angles fall within the range of $(-\pi, \pi)$, we add 2π to negative angles to ensure that all angles are within $(0, 2\pi)$. Each angle vector can be treated as a 3D point, and all the angular vectors of the transformations form a point cloud representing the resolution space. For two points \mathbf{r}_i and \mathbf{r}_j , the segmentation algorithm first checks if the spatial distance is less than the threshold t_1 (*e.g.*, 5 degrees in radians). If this condition is met, the points are further evaluated by a condition function f_c , which is formulated as:

$$f_c(\mathbf{r}_i, \mathbf{r}_j) = \|\mathbf{t}_i - \mathbf{t}_j\|.$$
(7)

If the result of f_c is less than the threshold t_2 (e.g., 10cm), the points are considered to belong to the same cluster. It should be noted that clustering primarily aims to organize discrete rotations and translations in the solution space, while the subsequent evaluation of transformations still employs the original matrix form, not Euler angles. The clustering method is replaceable.

7.2. Post Verification

After the point patches $\mathbf{Q}_1, \mathbf{Q}_2$ are obtained, we verify each candidate transformation $\mathbf{h}_k = (\mathbf{R}_k, \mathbf{t}_k)$ based on the modified Chamfer distance, which is calculated as follows:

$$Score_{CD}(\mathbf{h}_{k}) = \frac{1}{M} \sum_{\mathbf{x} \in \mathbf{Q}_{1}} \phi_{mae}(\min_{\mathbf{y} \in \mathbf{Q}_{2}} \|\mathbf{R}_{k}\mathbf{x} + \mathbf{t}_{k} - \mathbf{y}\|) + \frac{1}{N} \sum_{\mathbf{y} \in \mathbf{Q}_{2}} \phi_{mae}(\min_{\mathbf{x} \in \mathbf{Q}_{1}} \|\mathbf{y} - \mathbf{R}_{k}\mathbf{x} - \mathbf{t}_{k}\|).$$
(8)

Each x in patch \mathbf{Q}_1 will find its nearest neighbor y in patch \mathbf{Q}_2 after being transformed by \mathbf{R}_k and \mathbf{t}_k , and the average MAE score is calculated. Similarly, for each point y in \mathbf{Q}_2 , the nearest point x in \mathbf{Q}_1 is determined, and the average MAE score is computed. The verification score is finally calculated by averaging the two scores. The candidate with the greater score is more likely to correctly align the point patches and will be selected.

Mathad	Deferment		3DMatch	ı	31	3DLoMatch		
Method	Reference	RR(%)	RE(°)	TE(m)	RR(%)	RE(°)	TE(m)	
3DSN [21]	CVPR 2019	78.4	2.19	0.071	33.0	3.52	0.103	
FCGF [14]	CVPR 2019	85.1	2.14	0.070	40.1	3.74	0.100	
D3Feat [2]	CVPR 2020	81.6	2.16	0.067	37.2	3.36	0.103	
DGR [15]	CVPR 2020	85.3	2.10	0.067	48.7	3.95	0.113	
PCAM [7]	ICCV 2021	85.5	1.80	0.059	54.9	3.52	0.099	
DHVR [26]	ICCV 2021	91.9	2.25	0.078	65.4	4.97	0.123	
PREDATOR [23]	CVPR 2021	89.0	2.02	0.064	62.5	3.04	0.093	
CoFiNet [53]	NeurIPS 2021	89.3	2.44	0.067	67.5	5.44	0.155	
RegTR [51]	CVPR 2022	92.0	1.57	0.049	64.8	2.83	<u>0.077</u>	
Lepard [29]	CVPR 2022	93.5	2.48	0.072	69.0	4.10	0.108	
SC ² -PCR [11]	CVPR 2022	93.3	2.08	0.065	69.5	3.46	0.096	
GeoTrans [35]	CVPR 2022	92.0	1.72	0.062	75.0	2.93	0.089	
OIF-PCR [42]	NeurIPS 2022	92.4	-	-	76.1	-	-	
UDPReg [31]	CVPR 2023	91.4	1.64	0.064	64.3	2.95	0.086	
RoITr [54]	CVPR 2023	91.9	-	-	74.7	-	-	
VBReg [25]	CVPR 2023	93.5	2.04	0.095	69.9	-	-	
MRA [8]	ICCV 2023	<u>95.1</u>	1.32	0.043	75.4	2.49	0.072	
SIRA-PCR [9]	ICCV 2023	94.1	<u>1.54</u>	0.051	76.6	2.39	0.072	
RolTr w.		04.9	2.06	0.070	70 /	2.60	0.117	
MAC++	-	74.0	2.00	0.070	/0.4	5.09	0.117	
GeoTrans w.		05.7	2.06	0.064	70.1	2.00	0.117	
MAC++	-	33.1	2.00	0.004	13.1	5.99	0.117	

Table 13. Performance on the 3DMatch/3DLoMatch benchmarks.

Dataset	Data type	Application scenario	# Matching pairs
U3M [32]	Object	Registration	496
3DMatch [55]	Indoor scene	Registration	1623
3DLoMatch [23]	Indoor scene	Registration	1781
KITTI-LC [34]	Outdoor scene	Loop closure, Registration	3325

Table 14. Information of all tested datasets.

8. Data generation in controlled experiments

First, we randomly select a specified number of inliers from ground-truth matching and introduce small-scale random noise to the coordinates of these inlier points. Additionally, we introduce outliers by randomly generating some and using those produced by the FPFH descriptor. We assess the robustness by varying the number of inliers from 50 to 250. Each pair generates a total of 5000 matches.

9. Additional Results

# Corr.	Graph construction	Search MACs	VMP	GCCI	Total
250	0.72 (0.89%)	0.76 (0.94%)	0.44 (0.54%)	78.98 (97.63%)	80.90
500	2.41 (2.06%)	7.46 (6.38%)	1.59 (1.36%)	105.39 (90.20%)	116.85
1000	12.04 (8.68%)	18.48 (13.33%)	3.87 (2.79%)	104.29 (75.20%)	138.68
2500	57.82 (24.56%)	41.07 (17.45%)	15.90 (6.75%)	120.62 (51.24%)	235.41
5000	589.50(12.75%)	715.03(15.47%)	126.49(2.74%)	3191.38(69.04%)	4622.40

Table 15. Average consumed time (ms) of each part.

Time consumption of each part. The qualitative comparisons are summarized in Table 13. The details of all tested datasets are presented in Table 14. We analyzed the average time consumption of each part, as shown in Table 15. The increased runtime of MAC++ is mainly reflected in the hypothesis clustering part, which involves the use of point cloud segmentation functions provided by PCL. In the future, we hope to design a more suitable algorithm for MAC++ to improve runtime. However, the current method's runtime remains at a relatively low level.

More results varying K_1 . We have evaluated the quality of the generated hypotheses to demonstrate the rationale behind VMP (Please refer to Table 10). We supplement the following experiments on the datasets used in the controlled experiments. 1) We test the theoretical performance of MAC++ following "*Performance Upper Bound*" section. '*n*' refers to the judging threshold for the number of correct hypotheses. As shown in Table 16, on data with an inlier rate $\leq 1\%$, we observed a positive correlation between the performance and different size thresholds, indicating that increasing the size of the MAC pool is beneficial for improving RR.

K_1	n = 1	n = 5	n = 10	n = 20	n = 50
1	94.95 / 70.63	80.16 / 29.03	66.24 / 12.75	47.63 / 3.97	22.74 / 0.11
5	95.38 / 71.65	81.58 / 29.93	68.52 / 13.42	50.71/4.32	26.86 / 0.22
10	95.38 / 71.76	81.52 / 30.15	68.64 / 13.48	51.08 / 4.44	27.05 / 0.34
20	95.38 / 71.76	81.58 / 30.26	68.76 / 13.48	51.08 / 4.49	27.05 / 0.34

Table 16. Performance upper bound varying K_1 and 'n'.

2) We test the real performance varying inlier rate and pool size. As shown in Table 17, MAC++ reaches the ideal performance when K_1 is 10, consistent with the conclusion in the "*Parameter Analysis*".

K_1	1%	2%	3%	4%	5%
1	67.90 / 37.68	79.30/69.29	90.26 / 83.04	94.58 / 90.29	97.17/93.43
5	67.96 / 38.07	79.67 / 70.19	90.88 / 84.00	94.95 / 90.96	97.41/94.22
10	68.45 / 38.52	80.16 / 70.52	90.70 / 84.11	95.26/91.35	97.97 / 94.72
20	67.65 / 38.01	79.67 / 70.30	90.63 / 84.05	95.19 / 90.85	97.35/94.22

Table 17. Performance upper bound varying K_1 and inlier rate.

Claims about MAC++. We provide outlier rejection results on 3DLoMatch. Table 18 shows that MAC++ achieves the highest inlier precision (IP) and F1 measure. Meanwhile, MAC++ is better than MAC.

FPFH	RR	RE	TE	IP	IR	F1
PointDSC	20.38	4.04	10.25	25.47	27.61	26.12
SC^2 -PCR	38.57	4.03	10.31	32.59	38.37	34.86
MAC	40.88	3.66	9.45	34.05	39.59	36.24
MAC++	44.30	4.12	11.02	36.48	41.78	38.50
FCGF	RR	RE	TE	IP	IR	F1
PointDSC	56.20	3.87	10.48	44.01	52.29	47.26
SC^2 -PCR	58.73	3.80	10.44	46.47	56.02	50.27
MAC	59.85	3.50	9.75	46.97	56.49	50.76
MAC++	61.03	3.93	11.17	47.38	56.45	59.74

Table 18. Outlier rejection results on 3DLoMatch.

Comparison with SC²-PCR++ 1) SC²-PCR++ applied additional 1-to-K matches to achieve more robust registration. However, MAC++ is not designed for such a setting: it achieves robust evaluation by progressively pruning false correspondences and hypotheses simultaneously with no need for additional matching. 2) As shown in Table 19, under the 1-to-1 matching setting, we test SC²-PCR++ and find that MAC++ performs better with PREDATOR and

GeoTrans. 3) As shown in Tabe 20, even under 1-to-K

3DLoMatch #node=5000	FCGF	PREDATOR	GeoTrans
SC ² -PCR++	61.15	71.59	78.72
MAC++	59.50	73.70	79.10

Table 19. Registration results under 1-to-1 matching setting.

matching setting, MAC++ is still 4.33%/2.24% higher than SC²-PCR++ with FPFH/FCGF.

3DLoMatch #node=all	SC ² -PCR++	MAC++	MAC++ w. 1-to-K
FPFH	41.49	44.30	45.82
FCGF	61.15	61.03	63.39

Table 20. Registration results under 1-to-k matching setting.

10. Visualizations

We show more registration results in Fig. 5 and Fig. 6. Fig. 5 visualizes the registration process of MAC++ on the 3DLoMatch dataset, indicating its capability to handle lowoverlap data. Fig. 6 demonstrates that MAC++ can handle long-range point cloud pairs that other methods fail to register.



Figure 5. Visualization of Registration process on 3DLoMatch.



Figure 6. Qualitative comparison on KITTI-LC.