Robust Multiview Point Cloud Registration with Reliable Pose Graph Initialization and History Reweighting Supplementary Material

Haiping Wang*
Wuhan University

Yuan Liu*
The University of Hong Kong

Zhen Dong†
Wuhan University

Yulan Guo
Sun Yat-sen University

Yu-Shen Liu
Tsinghua University

Wenping Wang
Texas A&M University

Bisheng Yang†
Wuhan University

Abstract

In this supplementary material, we provide the detailed solution of pose synchronization in Sec. A.1, the implementation details in Sec. A.2, additional analysis in Sec. A.3, the running time analysis in Sec. A.4, and more qualitative results in Sec. A.6. The source code is available at https://github.com/WHU-USI3DV/SGHR.

A.1. Pose synchronization

In this section, we provide the detailed solution of pose synchronization in Sec. 3.3.2 of the main paper. Given the edge weights and input relative poses $\{w_{ij}, T_{ij} | (i,j) \in \mathcal{E}\}$, we solve the transformation synchronization by dividing it into rotation synchronization [2, 9] and translation synchronization [8]. In the following, our pairwise transformation $T_{ij} = (R_{ij}, t_{ij})$ on edge $(i,j) \in \mathcal{E}$ aligns the source scan P_j to the target scan P_i. The scan poses are assumed to be camera-to-world matrices. Thus scans under the correctly recovered poses $\{(R_i, t_i)\}$ should reconstruct the whole scenario.

Rotation synchronization. Following [2, 6, 11], we treat the synchronization of rotations $\{R_i\}$ as an over-constrained optimization problem:

$$\underset{R_1,\dots R_N \in SO(3)}{\arg\min} \sum_{(i,j)\in\mathcal{E}} w_{ij} \| R_{ij} - R_i^T R_j \|_F^2, \tag{A.1}$$

where $\|\cdot\|_F$ means the Frobenius norm of the matrix. Under the spectral relaxation, a closed-from solution of Eq. A.1 can be computed as follows [2, 6]. Consider a symmetric matrix $L \in \mathbb{R}^{3N*3N}$ containing N^2 3×3 blocks:

$$L = \begin{pmatrix} \sum\limits_{(1,j)\in\mathcal{E}} w_{1j}\mathbf{I}_3 & -w_{12}R_{12} & \cdots & -w_{1N}R_{1N} \\ -w_{21}R_{21} & \sum\limits_{(2,j)\in\mathcal{E}} w_{2j}\mathbf{I}_3 & \cdots & -w_{2N}R_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ -w_{N1}R_{N1} & -w_{N2}R_{N2} & \cdots & \sum\limits_{(N,j)\in\mathcal{E}} w_{Nj}\mathbf{I}_3 \end{pmatrix}, \tag{A.2}$$

where $\mathbf{I}_3 \in \mathbb{R}^{3*3}$ denotes the identity matrix. For each edge $(i,j) \in \mathcal{E}$, we fill $-w_{ij}R_{ij}$ and $-w_{ij}R_{ij}^T$ to the (i,j) and (j,i) block. For unconnected edges, we set the corresponding blocks to zeros.

We first calculate three eigenvectors $\tau_1, \tau_2, \tau_3 \in \mathbb{R}^{3N}$ corresponding to the three smallest eigenvalues $\lambda_1 < \lambda_2 < \lambda_3$ of L and stack them to form $\gamma = [\tau_1, \tau_2, \tau_3] \in \mathbb{R}^{3N*3}$. Then, $v_i = \gamma[3i-3 : 3i] \in \mathbb{R}^{3*3}$ is an approximation of the absolute rotation R_i for point cloud P_i but may not satisfy the constraint $v_i v_i^T = \mathbf{I}_3$. Therefore, we rectify this by applying singular value decomposition on v_i by $v_i = U_i \sum_i V_i^T$ and deriving $R_i = V_i U_i^T$ [2]. Then, we further check $det(R_i)$ and exchange the first two rows of R_i if $det(R_i) = -1$.

Translation synchronization. Translation synchronization retrieves the translation vectors $\{t_i\}$ that minimize the problem:

$$\underset{t_1,\dots,t_N \in \mathbb{R}^3}{\arg\min} \sum_{(i,j)\in\mathcal{E}} w_{ij} \| R_i t_{ij} - t_j + t_i \|^2. \tag{A.3}$$

We solve it by the standard least square method [8].

Assuming E edges are connected in $\mathcal{G}$, we thus construct three matrices A, B, and H as follows. $A \in \mathbb{R}^{3E*3E}$ is initialized as an identity matrix. $B \in \mathbb{R}^{3E*3N}$ contains $E*N$ 3×3 blocks and is initialized as a zero matrix. $H \in$

*Both authors contribute equally to this research.
†Corresponding authors: [dongzhenwhu, bshyang]@whu.edu.cn

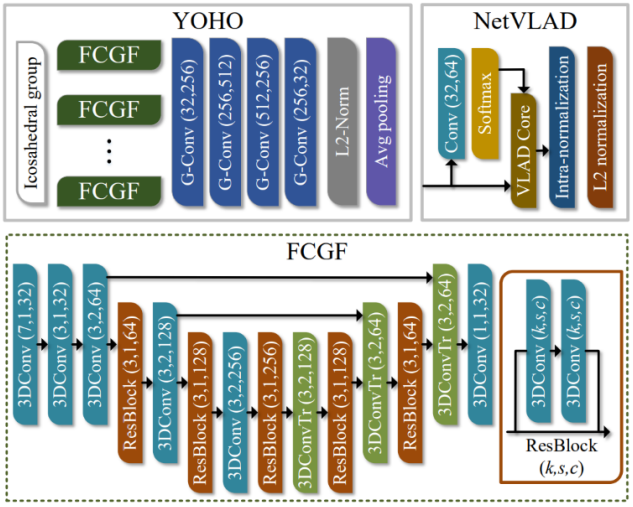

Figure A.1. Network architecture for global feature extraction. "G-Conv" means group convolution defined on Icosahedral group same as [12]. "VLAD core" is the same as [1]. For FCGF [4], "3DConv" and "3DConvTr" denotes a sparse convolution layer and the transpose convolution layer for upsampling, respectively.

$\mathbb{R}^{3E*1}$ is a vector containing E 3×1 blocks. For the *e-th* edge $(i, j) \in \mathcal{E}$, we multiply $A[3e-3:3e]$ with w_{ij}, fill $\mathbf{I}_3$ and -$\mathbf{I}_3$ to the (e, j) and (e, i) block of B respectively, and fill $R_i t_{ij}$ to the *e-th* block of H. We thus solve $t = (B^T AB)^{-1} B^T AH$ and obtain the translation vector t_i of each scan P_i as $t[3i-3:3i]$.

A.2. Implementation details

A.2.1. Architecture

The architecture of our global feature extraction network is shown in Fig. A.1. We adopt YOHO with the same architecture as [12] for 32-dim local feature extraction. More local feature extraction details can be found in [12]. The extracted local features are aggregated to a global feature by a *NetVLAD* layer [1]. We set the number of clusters in *NetVLAD* to 64 and the dimension of the global feature is thus 2048. Please refer to [1] for more global feature aggregation details.

A.2.2. Training details

We use the pretrained YOHO [12] for local feature extraction and train the $NetVLAD$ layer using the 46 scenes in the training split of 3DMatch [15]. We adopt the following data augmentations. For each scene in the train set of 3DMatch, we first randomly sample $\alpha \in [8, 60]$ scans as the graph node. Then, on each scan, we randomly sample $\beta \in [1024, 5000]$ keypoints to extract YOHO features. The local features of α scans are fed to *NetVLAD* to extract α scan global features. Then, we compute the $\binom{\alpha}{2}$ overlap scores by exhaustively correlating every two global features and compute the L1 distance between the ground-truth overlap ratios and the predicted overlap scores as the loss for training. We set the batch size to 1 and use the Adam optimizer with a learning rate of 1e-3. The learning rate is exponentially decayed by a factor of 0.7 every 50 epoch. In total, we train the $NetVLAD$ for 300 epochs.

Overlap Estimation	*3D-RR(%)*	*3DLo-RR(%)*
Predator [7]	95.2	78.4
Ours	96.2	81.6

Table A.1. Registration recall on 3D(Lo)Match using estimated overlap scores from Predator [7] and ours.

A.3. More analysis

A.3.1. Use Predator [7] for overlap estimation

In Table. A.1, we use the overlap scores predicted by Predator [7] in the sparse graph construction, which yields slightly worse results. Moreover, Predator [7] applies cross-attention layers between local features of a scan pair to estimate overlap while we only need to compute a global feature for every scan and efficiently correlate the global features to estimate overlap. In our test, the proposed method is 10× more efficient than Predator.

A.3.2. Concurrent multiview registration works

After our submission to CVPR 2023, two concurrent mulitview registration works are available online, namely, SynMatch [5] and HL-MRF [13]. SyncMatch and HL-MRF are specifically designed for registering raw RGB-D sequences and TLS point clouds, respectively, while the proposed method offering a more general approach. In our test, the proposed method notably outperforms SynMatch by 27% on the 3DMatch dataset. HL-MRF indeed performs well on the TLS-based ETH dataset but fails on the indoor datasets.

A.3.3. Estimated overlap vs. ground truth overlap

In Fig. A.2, each point (o_{gt}, o_{est}) represents a scan pair with the ground truth overlap ratio o_{gt} and estimated overlap ratio o_{est}. The plot reveals several observations: (1) scan pairs with larger ground truth overlaps indeed have larger overlap scores; (2) the constructed sparse graph mainly contains scan pairs with higher overlap ratios, as evidenced by the green and red points; (3) the proposed transformation synchronization algorithm further eliminates unreliable scan pairs effectively to achieve accurate scan poses, as shown by the red points.

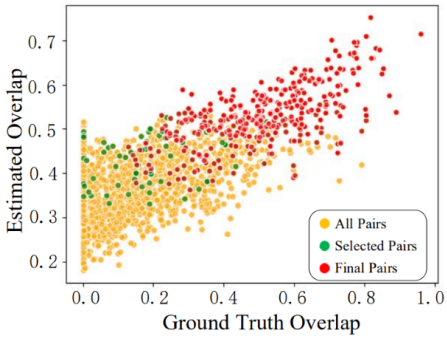

Figure A.2. Estimated overlap ratio versus the ground truth overlap ratio on scan pairs of the *Kitchen* scene of 3DMatch. "All pairs" means all $\binom{N}{2}$ scan pairs. "Selected pairs" means the scan pairs selected to construct the sparse pose graph. "Final pairs" means the scan pairs with an edge weight greater than 10^{-2} after transformation synchronization.

Top-	4	6	8	10	12	15	Full
# Pair	**1167**	1707	2250	2798	3349	4129	11905
Sync-time (s)	**20.2**	30.4	37.4	54.8	66.6	90.3	405.4
3D-RR (%)	91.3	91.6	95.5	96.2	**96.6**	96.0	93.2
3DL-RR (%)	71.0	74.7	80.9	**81.6**	81.2	80.3	76.8

Table A.2. Ablation study on k in sparse graph construction. "Full" means using fully-connected graphs. "Sync-time" means the time for transformation synchronization.

A.3.4. Construct sparse graph with different *top-k*

In Table. A.2, we show the results with different k in the sparse graph construction. Retaining too many scan pairs with larger k may include more outliers while using too small k could split the whole graph into several disconnected subgraphs. Results show that using $k = 10$ or 12 brings the best results.

A.3.5. Performances using different IRLS iterations

In Fig. A.3, we show the registration performance on 3D(Lo)Match with different iteration numbers. It can be seen that the results will be better with more iterations. However, using more iterations also costs more time. We thus select 50 iterations for its stable performance and efficiency by default.

A.4. Runtime analysis

In Table. A.3, we provide the runtime for the graph construction and the IRLS-based transformation synchronization averaged on the 8 scenes of the 3DMatch dataset. We evaluate the runtimes on a computer with Intel(R) Core(TM) i7-10700 CPU@ 2.90GHz with GeForce GTX 2080Ti and 64 GB RAM. Our sparse pose graph construction is nearly $67s$ faster than baselines for conducting much fewer pairwise registrations. In total, our method is $61s \sim 160s$ faster than baselines for registering a scene in 3DMatch.

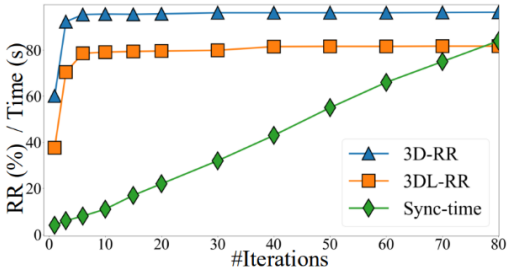

Figure A.3. Results of the proposed history reweighting IRLS with different iterations.

Method	*Graph Cons* (s)	*Trans Sync* (s)	*Total* (s)
RotAvg [3] + Full	86.3	49.4	135.7
LITS [14] + Full	86.3	0.7	87.0
HARA [10] + Full	87.3	8.5	95.8
RotAvg [3] + Pruned [6]	164.1	22.6	186.7
LITS [14] + Pruned [6]	164.1	**0.7**	164.8
HARA [10] + Pruned [6]	164.8	7.5	172.4
Ours	**20.0**	6.9	**26.8**

Table A.3. Detailed time consumption for registering a scene on 3DMatch. "Graph Cons" means the time for constructing the input pose graph. "Trans Sync" means the time for IRLS-based transformation synchronization.

Figure A.4. A failure case in ScanNet. (a) The ground truth multiview registration (30 scans). (b) The multiview registration from the proposed method.

A.5. Limitations

When the overlap ratios of two scans are too small and there are no other scans which forms a cycle with these two scans, our method may fail in this case. A typical example is shown in Fig. A.4, where overlap region in the red rectangle is very small and mainly consists of feature-less planar points. In this case, our method fails to register the whole scene but separately recover poses on two subgraphs. This also shows that our method may have the potential to automatically separate scans from two different scenes, which is

beyond the discussion of this paper.

A.6. More qualitative results

We provide additional qualitative results including success cases (Fig. A.5 and Fig. A.6) and failure cases (Fig. A.7). We also compare our results with the registration results of RotAvg [3], HARA [10], and LITS [14]. The failure of our method occurs when some overlap regions mainly contain the repetitive structures (top of Fig. A.7) or feature-less regions (bottom of Fig. A.7).

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. *IEEE TPAMI*, 40(06):1437–1451, 2018. 2

[2] Mica Arie-Nachimson, Shahar Z Kovalsky, Ira Kemelmacher-Shlizerman, Amit Singer, and Ronen Basri. Global motion estimation from point matches. In *3DIMPVT*, 2012. 1

[3] Avishek Chatterjee and Venu Madhav Govindu. Robust relative rotation averaging. *IEEE TPAMI*, 40(4):958–972, 2017. 3, 4, 5, 6, 7

[4] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *ICCV*, 2019. 2

[5] Mohamed El Banani, Ignacio Rocco, David Novotny, Andrea Vedaldi, Natalia Neverova, Justin Johnson, and Ben Graham. Self-supervised correspondence estimation via multiview registration. In *WACV*, 2023. 2

[6] Zan Gojcic, Caifa Zhou, Jan D Wegner, Leonidas J Guibas, and Tolga Birdal. Learning multiview 3d point cloud registration. In *CVPR*, 2020. 1, 3

[7] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *CVPR*, 2021. 2

[8] Xiangru Huang, Zhenxiao Liang, Chandrajit Bajaj, and Qixing Huang. Translation synchronization via truncated least squares. *NeurIPS*, 2017. 1

[9] Xiangru Huang, Zhenxiao Liang, Xiaowei Zhou, Yao Xie, Leonidas J Guibas, and Qixing Huang. Learning transformation synchronization. In *CVPR*, 2019. 1

[10] Seong Hun Lee and Javier Civera. Hara: A hierarchical approach for robust rotation averaging. In *CVPR*, 2022. 3, 4, 5, 6, 7

[11] Daniel Martinec and Tomas Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *CVPR*, 2007. 1

[12] Haiping Wang, Yuan Liu, Zhen Dong, and Wenping Wang. You only hypothesize once: Point cloud registration with rotation-equivariant descriptors. In *ACM Multimedia*, 2022. 2

[13] Hao Wu, Li Yan, Hong Xie, Pengcheng Wei, and Jicheng Dai. A hierarchical multiview registration framework of tls point clouds based on loop constraint. *ISPRS J*, 2023. 2

[14] Zi Jian Yew and Gim Hee Lee. Learning iterative robust transformation synchronization. In *3DV*, 2021. 3, 4, 5, 6, 7

[15] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, 2017. 2

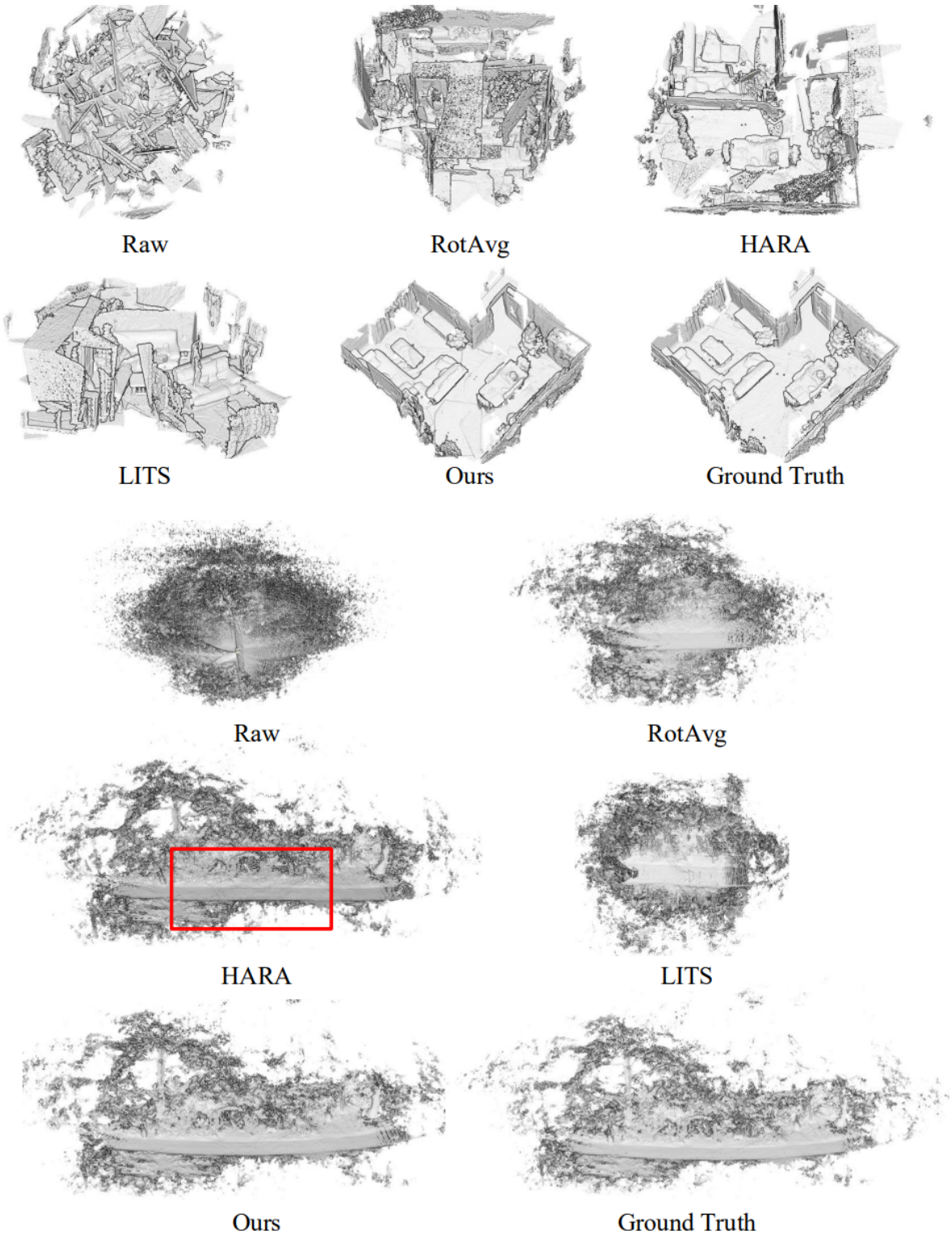

Figure A.5. Registration results of our method, RotAvg [3], HARA [10], and LITS [14] on the 3DMatch dataset and the ETH dataset. Top: the *Home1* scene of 3DMatch. Bottom: the *Wood_Summer* scene of ETH.

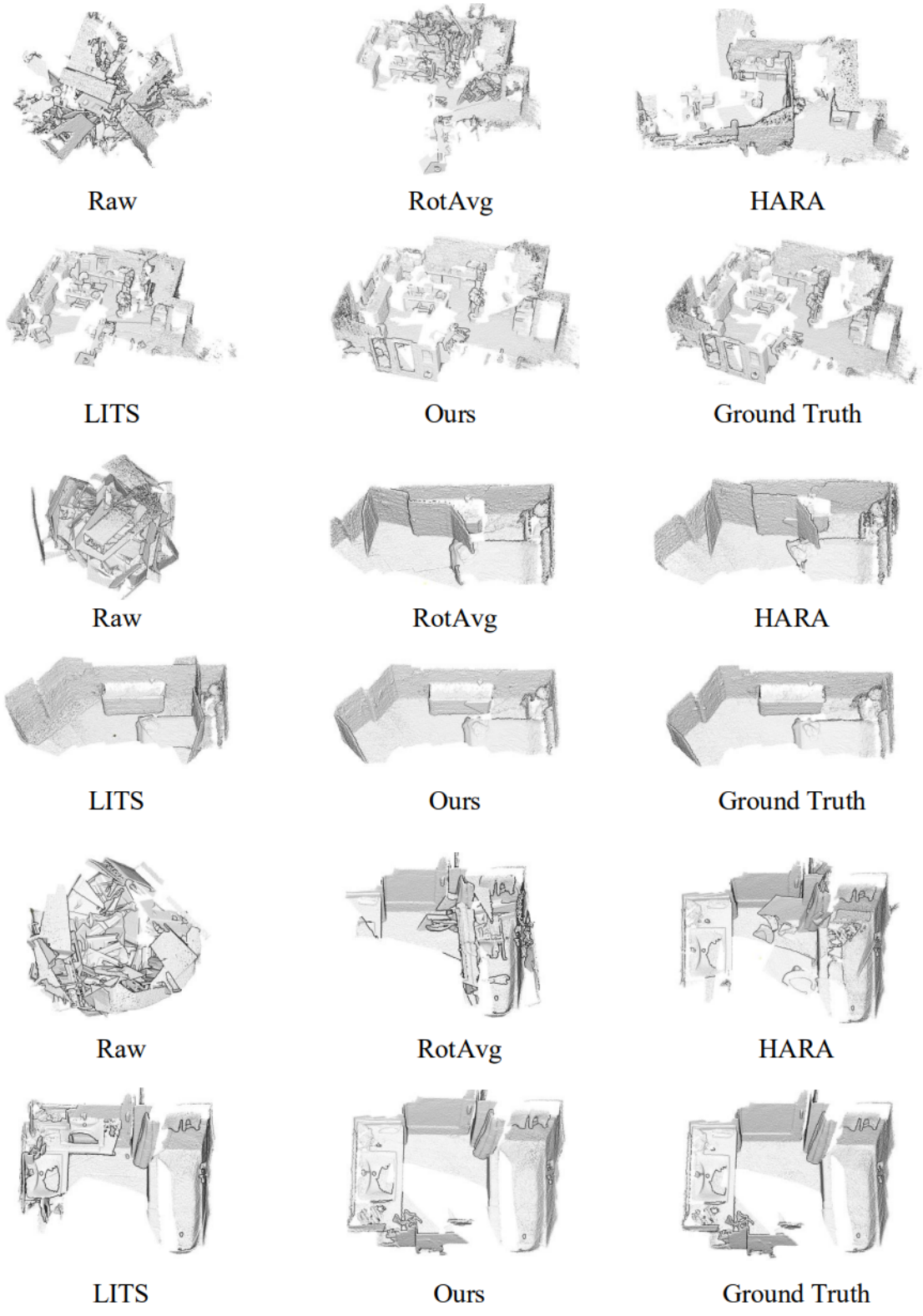

Figure A.6. Registration results of our method, RotAvg [3], HARA [10], and LITS [14] on scenes of ScanNet dataset including *Scene0309_00* (top), *Scene0286_02* (middle), and *Scene0265_02* (bottom).

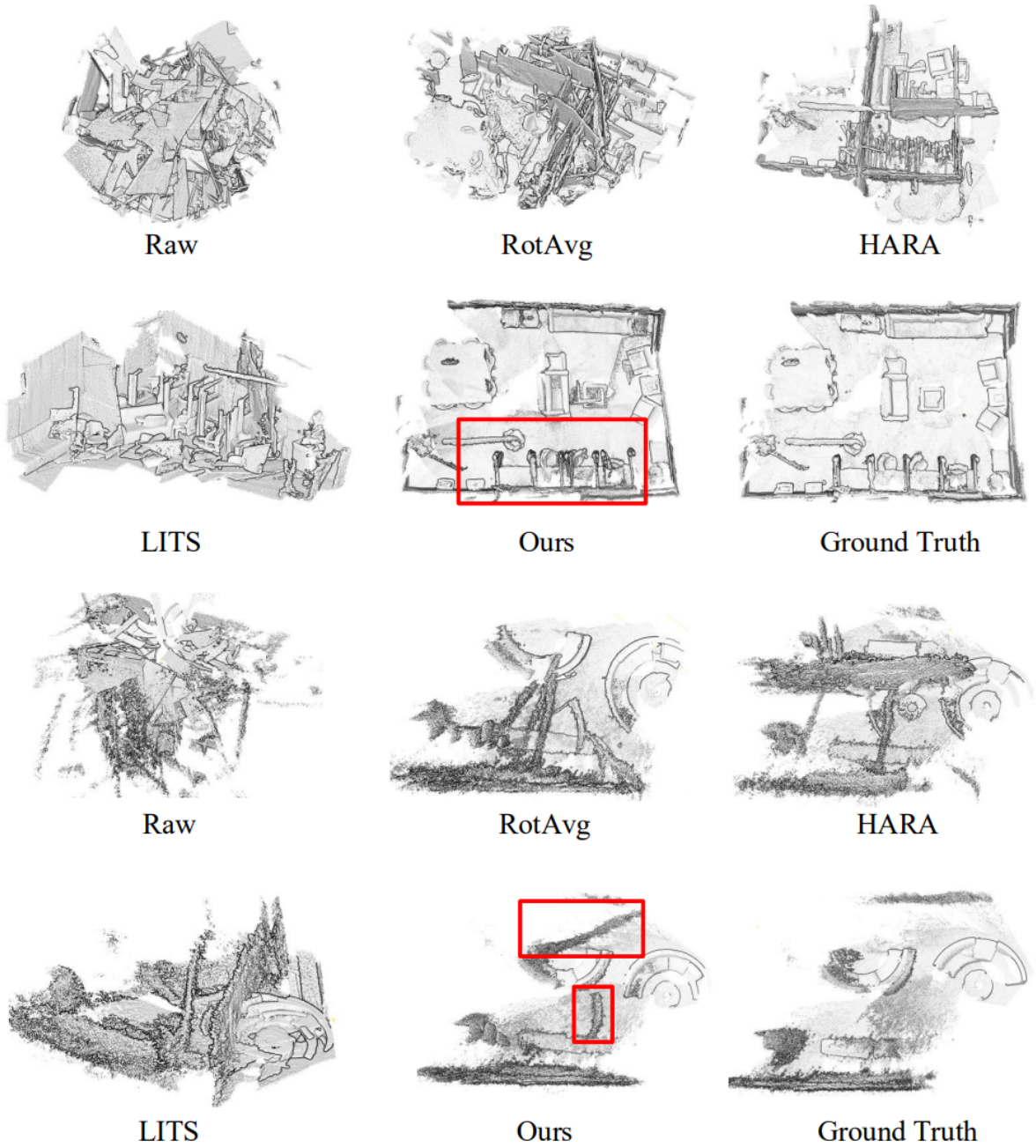

Figure A.7. Registration results of our method, RotAvg [3], HARA [10], and LITS [14] on 3DMatch (top: *Studyroom*) and ScanNet (bottom: *Scene0334_02*). Our method fails to register the scans in the red boxes due to repetitive structures and feature-less regions.