Multi-instance Point Cloud Registration by Efficient Correspondence Clustering

Weixuan Tang and Danping Zou
Shanghai Jiao Tong University
Shanghai,China
{weixuantang|dpzou}@sjtu.edu.cn

1. About this appendix

We provide extra details about the proposed method, the evaluation metrics, and the experiments because of the limited space in the main draft. We also present additional qualitative results of our experiments in Figure 6-7 (benchmark data) and Figure 8 (Real-world data).

2. Definition of evaluation metrics

Three metrics are used to evaluate the performance of multi-instance point cloud registration. They are Mean Hit Recall, Mean Hit Precision, and Mean Hit F1.

Before we start to compute those metrics, the first step is to establish the ground truth and estimation pair. Assuming there are K ground truth transformations and M estimated transformations, we construct an assignment matrix $\mathcal{F} \in \mathbb{K} \times \mathbb{M}$ and obtain the one-to-one mapping by solving the linear assignment problem [3]. To construct the assignment matrix $\mathcal{F}$, we use F-norm to compute the distance between each pair of ground-truth and estimated transformation, $\mathbf{T}_i^*$ and $\hat{\mathbf{T}}_j$:

$$\mathcal{F}(i,j) = \|\mathbf{T}_i^* - \hat{\mathbf{T}}_j\|_F \tag{1}$$

where the transformation matrix is defined as

$$\mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \in \mathbb{R}^{4\times 4}. \tag{2}$$

After solving the linear assignment problem, we obtain $S = \min(K, M)$ GT-estimation pairs. We then define the Relative Rotation Error (RRE) and the Relative Translation Error (RTE) between a GT-estimation pair $\mathbf{T}_s^*$ and $\hat{\mathbf{T}}_s$ as

$$\begin{aligned} RRE_s &= \arccos\left((tr(\hat{\mathbf{R}}_s^T \mathbf{R}_s^*) - 1)/2\right) \\ RTE_s &= \|\mathbf{t}_s^* - \hat{\mathbf{t}}_s\| \end{aligned} \tag{3}$$

With the two errors, we define the evaluation metrics in the following sections.

2.1. Mean Hit Recall (MHR)

The Mean Hit Recall between two registered point clouds is defined as

$$MHR = \frac{1}{K}\sum_{s=1}^{S} I_s \tag{4}$$

where $I_s = \{0, 1\}$ represents whether a GT-estimation pair being 'hit'. Specifically,

$$I_s = I(RRE_s < \tau_r) \times I(RTE_s < \tau_t) \tag{5}$$

where $I(\cdot) = \{0, 1\}$ denotes an indicating function. RRE_s and RTE_s are the relative rotation error and relative translation error for the sth GT-estimation pair defined in (3). The two thresholds τ_r and τ_t are set to be 20° and $0.5m$ respectively in all our experiments.

The final Mean Hit Recall is obtained by averaging the MHR of all the point cloud pairs.

2.2. Mean Hit Precision (MHP)

The Mean Hit Precision between two registered point clouds is defined as

$$MHP = \frac{1}{M}\sum_{s=1}^{S} I_s. \tag{6}$$

The final Mean Hit Precision is obtained by averaging the MHP of all the point cloud pairs.

2.3. Mean Hit F1 (MHF1)

The Mean Hit F1 between two registered point clouds is defined as

$$MHF1 = \frac{2 * MHP * MHR}{MHP + MHR} \tag{7}$$

The final Mean Hit F1 is obtained by averaging the MHF1 of all the point cloud pairs.

3. Some details about the experiments

3.1. Feature extraction & one-to-many feature matching

We extract features for all the points both in the source point cloud and the target point cloud before we start to match the corresponding points. We use the state-of-art point cloud feature extractors PREDATOR and D3Feat in our synthetic and benchmark experiments. We found the pre-trained models (trained on ModelNet40) of both PREDATOR and D3Feat perform so poorly that we train those models from scratch using the datasets used for evaluation.

To obtain multi-instance correspondences, instead of matching the source point cloud to the target point clouds, we match them in the reversed order. In other words, for each point in the target point cloud, we find the most similar one in the source point cloud by nearest neighborhood feature matching. In this way, we can associate multiple target points with a single source point.

3.2. About estimated outlier ratios

In Table 2 $\sim$ 4 of the main draft, we present the estimated outlier ratio for each result. Here we present how the outlier ratio is estimated. Given the ground truth transformations from the source points to the target points, we may establish the ground truth correspondences between the source points and the target points. The outlier correspondences are those who are different from the ground truth ones. Namely, for an estimated correspondence, we find the ground truth correspondence with the same target point and check if its source point is the same as that of the ground truth.

4. Ablation Study

Tab1 shows ablation study on the two successive steps of our method - **clustering** and **refinement** - on both synthetic and real-world tests (see Tab 2 and Tab 3 for results of other methods). The clustering step produces noisy clusters leading to high recall but low precision. The refinement step can remove outliers and merge duplicated clusters, achieving both high recall and high precision.

Fig1 shows performance change with five parameters, the top line are results tested on the synthetic dataset and the bottom line are those on the Scan2CAD dataset. The first three columns show the ablation study on three key parameters. The performance curves are smooth (except γ_thresh, which is used to trade-off recall and precision) and have similar shapes across different datasets, indicating those parameters are easy to be tuned and generalize well for different datasets. The last two columns imply that the parameters in Eq (6) and (7) have little impact on the performance except for a high IOU threshold (> 0.9). The values used in all our experiments are indicated by vertical dot lines.

Step	MHR(%) ↑	MHP(%) ↑	MHF1(%) ↑	Time	#Clusters
Synthetic Dataset with PREDATOR					
clustering	**61.56**	6.09	10.70	0.23s	76.63
clustering+refinement	53.39	**61.44**	**51.80**	0.48s	8.10
Scan2CAD Dataset with PREDATOR					
clustering	**41.66**	4.19	7.19	0.36s	51.38
clustering+refinement	31.63	**29.23**	**27.04**	0.51s	5.86

Table 1. Ablation study on **clustering** and **refinement**.

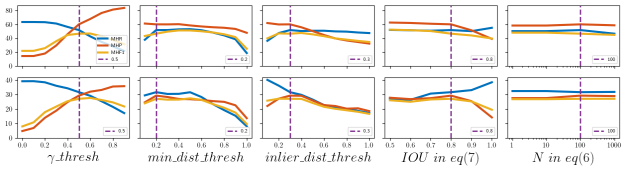
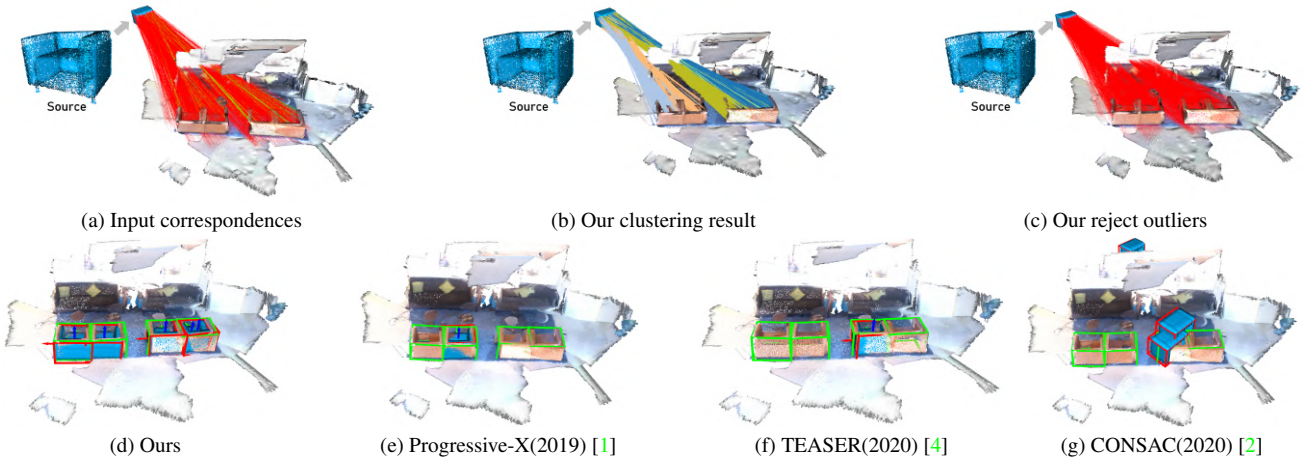


Figure 1. **Top**: Results on the synthetic dataset. **Bottom:** Results on the Scan2CAD dataset.

5. Qualitative result

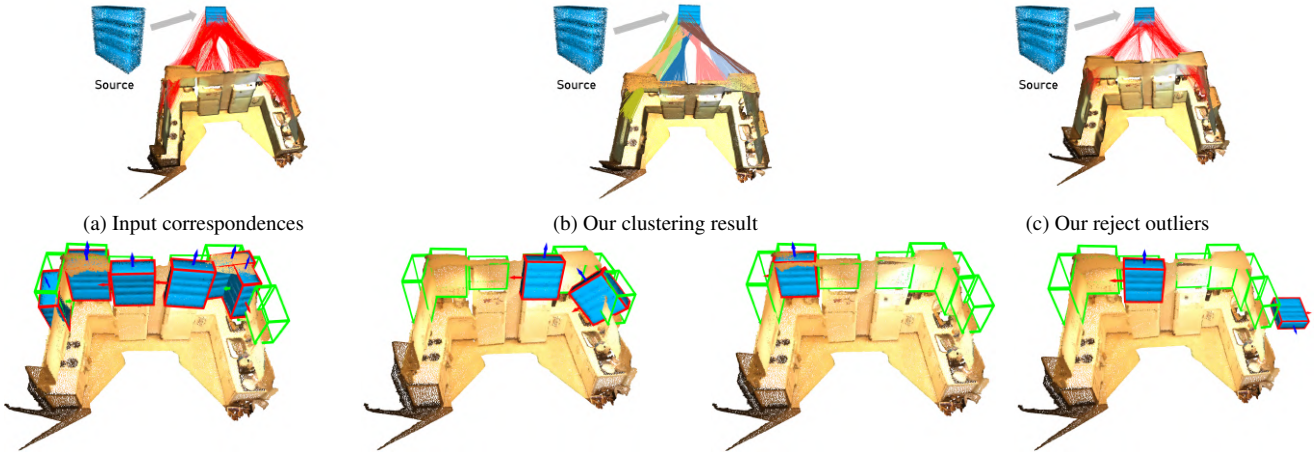
We show extra qualitative results of our experiments. The results of Scan2CAD are shown in Figure 2-13, while the results of real-world tests are shown in Figure 14-31. Video results are also available for real-world results.



(a) Input correspondences (b) Our clustering result (c) Our reject outliers

(d) Ours (e) Progressive-X(2019) [1] (f) TEASER(2020) [4] (g) CONSAC(2020) [2]

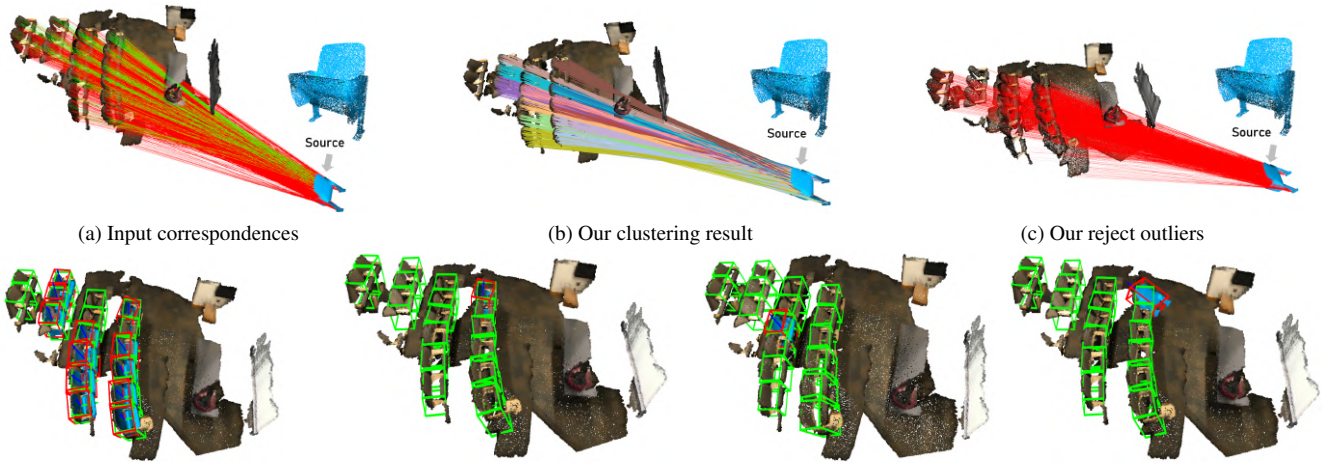
Figure 2. **Scan2CAD results.**



(a) Input correspondences (b) Our clustering result (c) Our reject outliers

(d) Ours (e) Progressive-X(2019) [1] (f) TEASER(2020) [4] (g) CONSAC(2020) [2]

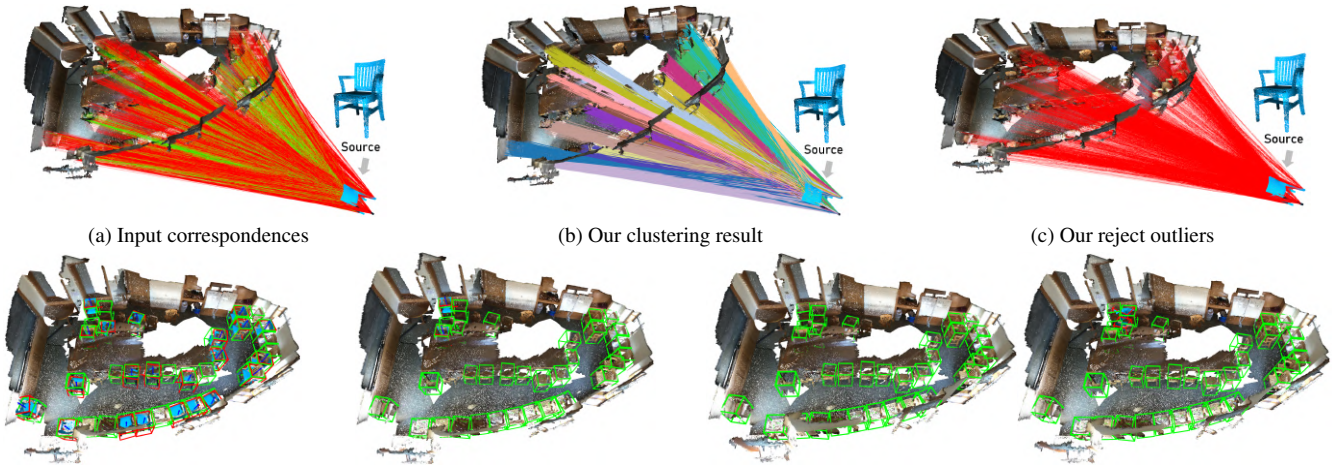
Figure 3. **Scan2CAD results.**



(a) Input correspondences (b) Our clustering result (c) Our reject outliers

(d) Ours (e) Progressive-X(2019) [1] (f) TEASER(2020) [4] (g) CONSAC(2020) [2]

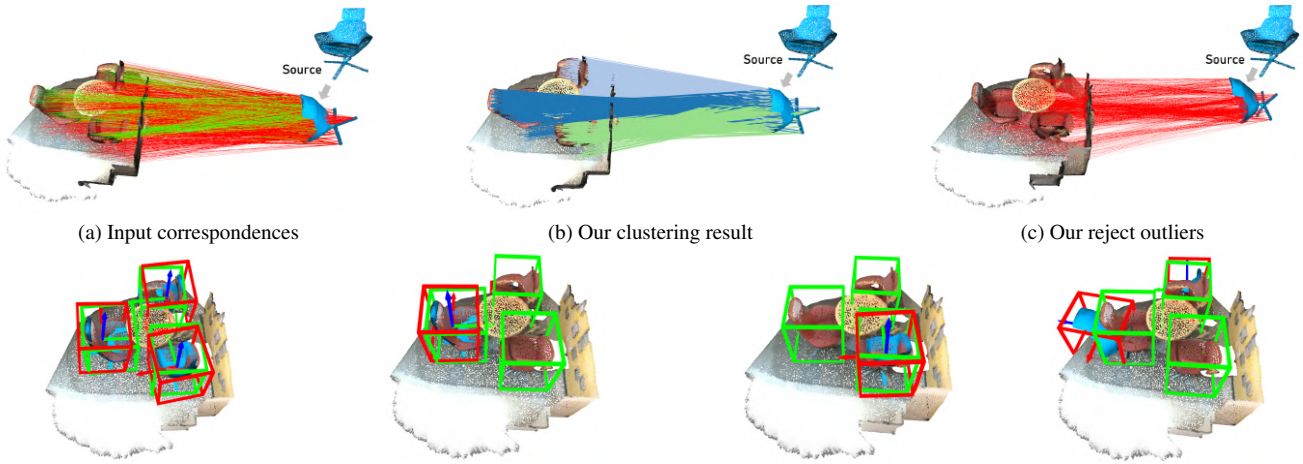
Figure 4. **Scan2CAD results.**



(a) Input correspondences (b) Our clustering result (c) Our reject outliers

(d) Ours (e) Progressive-X(2019) [1] (f) TEASER(2020) [4] (g) CONSAC(2020) [2]

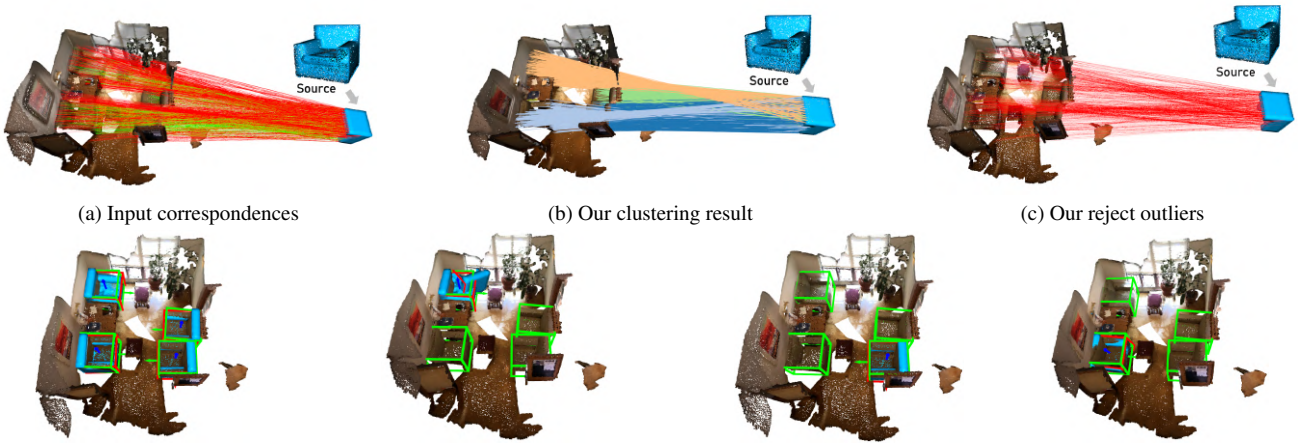
Figure 5. **Scan2CAD results.**



(a) Input correspondences (b) Our clustering result (c) Our reject outliers

(d) Ours (e) Progressive-X(2019) [1] (f) TEASER(2020) [4] (g) CONSAC(2020) [2]

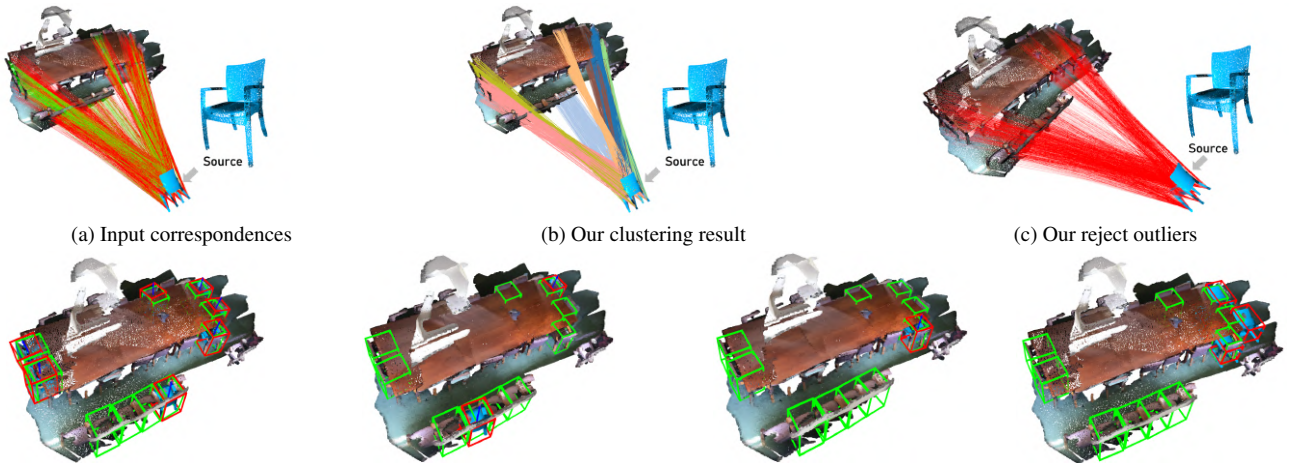
Figure 6. **Scan2CAD results.**



(a) Input correspondences (b) Our clustering result (c) Our reject outliers

(d) Ours (e) Progressive-X(2019) [1] (f) TEASER(2020) [4] (g) CONSAC(2020) [2]

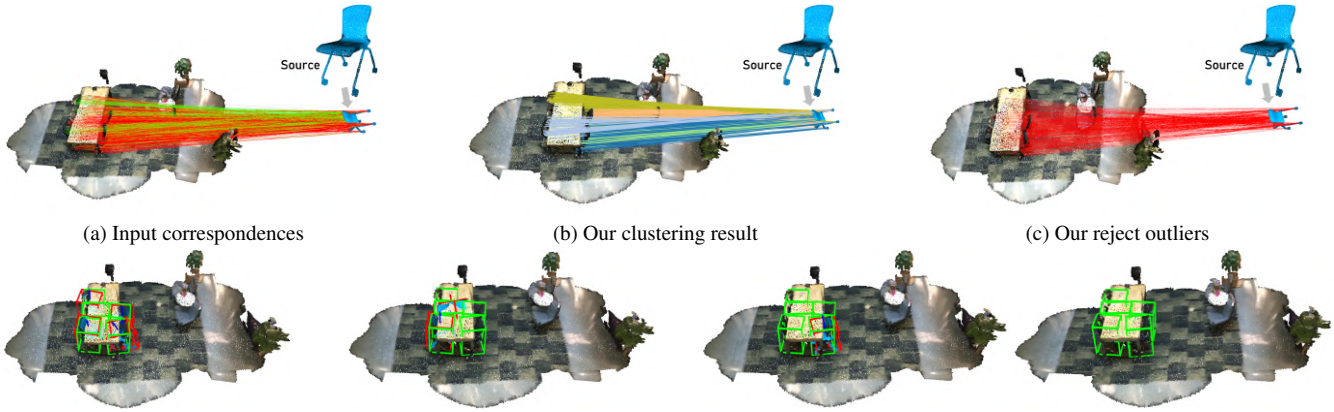
Figure 7. **Scan2CAD results.**



(a) Input correspondences (b) Our clustering result (c) Our reject outliers

(d) Ours (e) Progressive-X(2019) [1] (f) TEASER(2020) [4] (g) CONSAC(2020) [2]

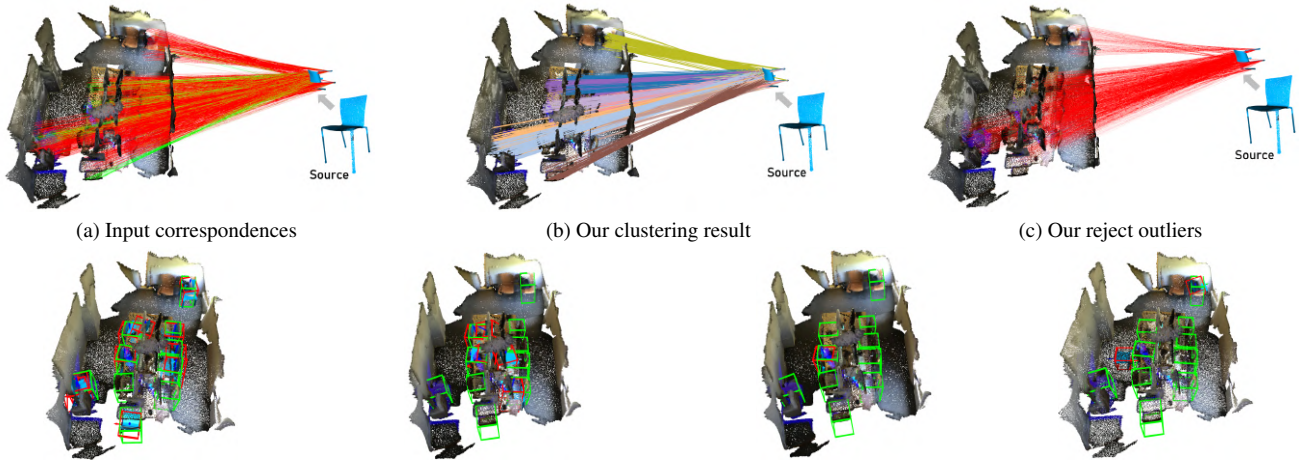
Figure 8. **Scan2CAD results.**



(a) Input correspondences (b) Our clustering result (c) Our reject outliers

(d) Ours (e) Progressive-X(2019) [1] (f) TEASER(2020) [4] (g) CONSAC(2020) [2]

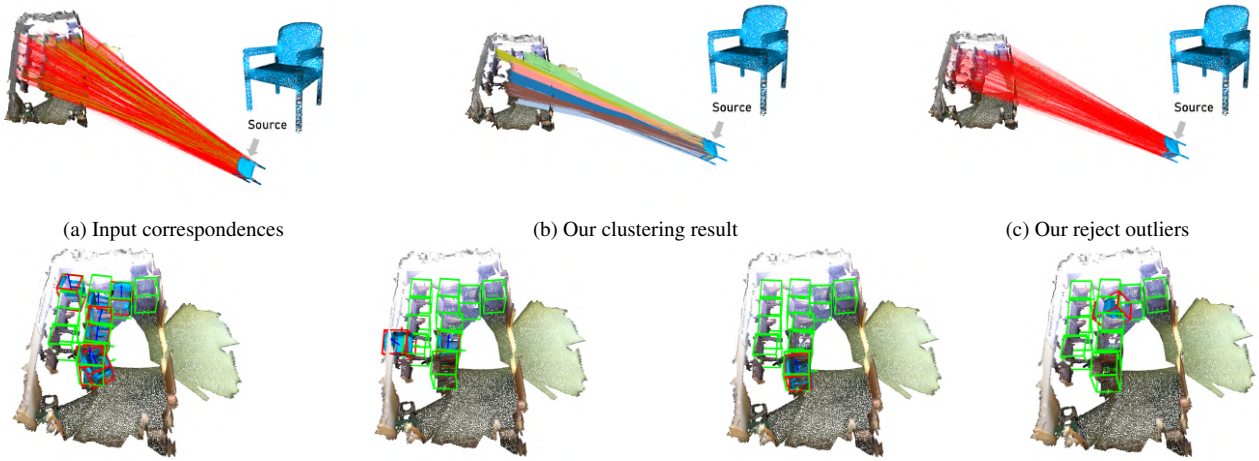
Figure 9. **Scan2CAD results.**



(a) Input correspondences (b) Our clustering result (c) Our reject outliers

(d) Ours (e) Progressive-X(2019) [1] (f) TEASER(2020) [4] (g) CONSAC(2020) [2]

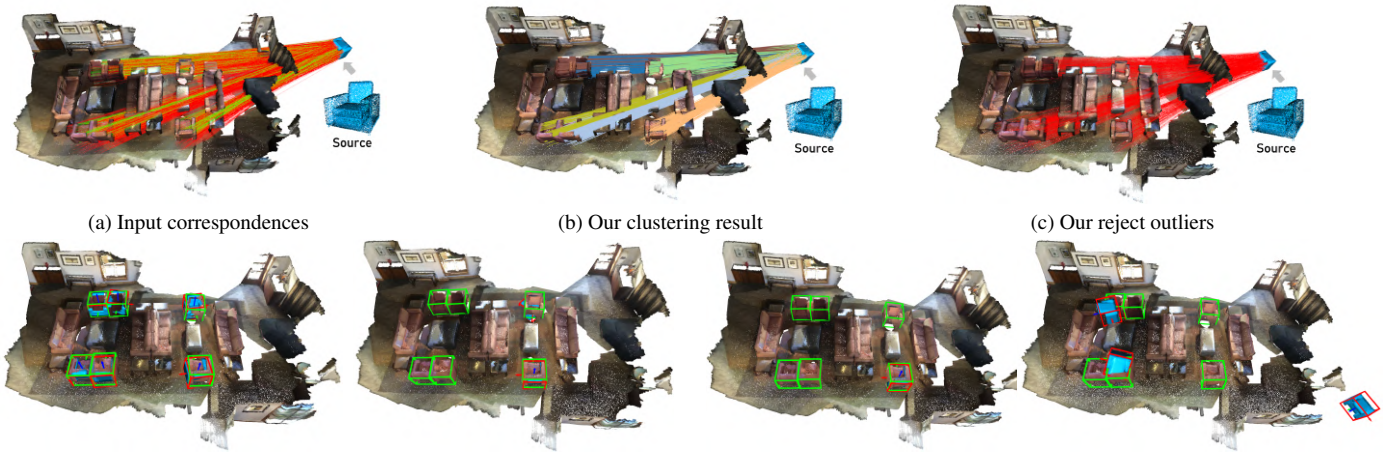
Figure 10. **Scan2CAD results.**



(a) Input correspondences (b) Our clustering result (c) Our reject outliers

(d) Ours (e) Progressive-X(2019) [1] (f) TEASER(2020) [4] (g) CONSAC(2020) [2]

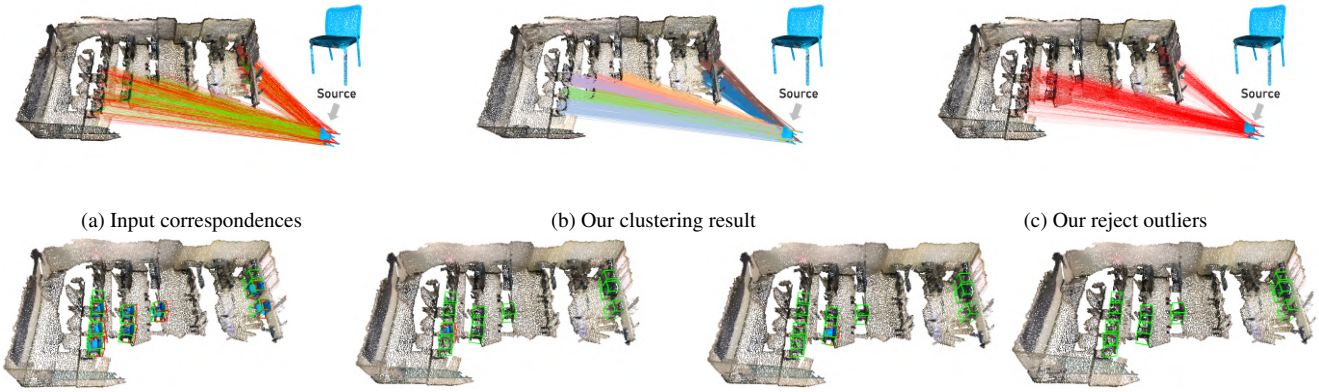
Figure 11. **Scan2CAD results.**



(a) Input correspondences (b) Our clustering result (c) Our reject outliers

(d) Ours (e) Progressive-X(2019) [1] (f) TEASER(2020) [4] (g) CONSAC(2020) [2]

Figure 12. **Scan2CAD results.**



(a) Input correspondences (b) Our clustering result (c) Our reject outliers

(d) Ours (e) Progressive-X(2019) [1] (f) TEASER(2020) [4] (g) CONSAC(2020) [2]

Figure 13. **Scan2CAD results.**

(a) Ours (b) Progressive-X(2019) [1] (c) CONSAC(2020) [2]

Figure 14. **Real-world tests on RGB-D scans.**

(a) Ours (b) Progressive-X(2019) [1] (c) CONSAC(2020) [2]

Figure 15. **Real-world tests on RGB-D scans.**

(a) Ours (b) Progressive-X(2019) [1] (c) CONSAC(2020) [2]

Figure 16. **Real-world tests on RGB-D scans.**

(a) Ours (b) Progressive-X(2019) [1] (c) CONSAC(2020) [2]

Figure 17. **Real-world tests on RGB-D scans.**

(a) Ours (b) Progressive-X(2019) [1] (c) CONSAC(2020) [2]

Figure 18. **Real-world tests on RGB-D scans.**

(a) Ours (b) Progressive-X(2019) [1] (c) CONSAC(2020) [2]

Figure 19. **Real-world tests on RGB-D scans.**

(a) Ours (b) Progressive-X(2019) [1] (c) CONSAC(2020) [2]

Figure 20. **Real-world tests on RGB-D scans.**

(a) Ours (b) Progressive-X(2019) [1] (c) CONSAC(2020) [2]

Figure 21. **Real-world tests on RGB-D scans.**

(a) Ours (b) Progressive-X(2019) [1] (c) CONSAC(2020) [2]

Figure 22. **Real-world tests on RGB-D scans.**

(a) Ours (b) Progressive-X(2019) [1] (c) CONSAC(2020) [2]

Figure 23. **Real-world tests on RGB-D scans.**

(a) Ours (b) Progressive-X(2019) [1] (c) CONSAC(2020) [2]

Figure 24. **Real-world tests on RGB-D scans.**

(a) Ours (b) Progressive-X(2019) [1] (c) CONSAC(2020) [2]

Figure 25. **Real-world tests on RGB-D scans.**

(a) Ours (b) Progressive-X(2019) [1] (c) CONSAC(2020) [2]

Figure 26. **Real-world tests on RGB-D scans.**

(a) Ours (b) Progressive-X(2019) [1] (c) CONSAC(2020) [2]

Figure 27. **Real-world tests on RGB-D scans.**

(a) Ours (b) Progressive-X(2019) [1] (c) CONSAC(2020) [2]

Figure 28. **Real-world tests on RGB-D scans.**

(a) Ours (b) Progressive-X(2019) [1] (c) CONSAC(2020) [2]

Figure 29. **Real-world tests on RGB-D scans.**

(a) Ours (b) Progressive-X(2019) [1] (c) CONSAC(2020) [2]

Figure 30. **Real-world tests on RGB-D scans.**

(a) Ours (b) Progressive-X(2019) [1] (c) CONSAC(2020) [2]

Figure 31. **Real-world tests on RGB-D scans.**

References

[1] Daniel Barath and Jiri Matas. Progressive-x: Efficient, anytime, multi-model fitting algorithm. In *ICCV*, pages 3780–3788, 2019. 3, 4, 5, 6, 7, 8, 9, 10

[2] Florian Kluger, Eric Brachmann, Hanno Ackermann, Carsten Rother, Michael Ying Yang, and Bodo Rosenhahn. Consac: Robust multi-model fitting by conditional sample consensus. In *CVPR*, 2020. 3, 4, 5, 6, 7, 8, 9, 10

[3] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 1

[4] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *TRO*, 37(2):314–333, 2020. 3, 4, 5, 6