Extend Your Own Correspondences: Unsupervised Distant Point Cloud Registration by Progressive Distance Extension

Supplementary Material

7. Detailed Experiment Setup

7.1. Comparison Methods

Considering the lack of genuine unsupervised distant point cloud registration methods at present, we compare EYOC against supervised methods instead. The most compared baselines are the two fully-convolutional methods, FCGF [9] and Predator [20]. The former utilizes MinkowskiNet for sparse voxel convolution, while the latter builds upon KPConv which classifies as a point convolution method. On the other hand, performances of SpinNet [2], D3Feat [4], CoFiNet [54], and GeoTransformer [37] are quoted verbatim from GCL [29].

7.2. Formal Metric Definition

Given a test set with labels $X_{[d_1,d_2]} = \{(S^i, T^i, R^i, t^i) \big| \, ||t^i||_2 \in [d_1, d_2]\}$ where S^i, T^i are point clouds and $R^i \in SO(3), t^i \in \mathbb{R}^3$ are the ground truth transformation, along with the estimated transformation $\hat{R}^i, \hat{t}^i$, the absolute rotational error and absolute translational error are defined as Eqs. (5) and (6). Please note that we abbreviate X for $X_{[d_1,d_2]}$ hereafter to save space where the subscript does not matter.

$$\mathbf{RE}^i_X = \arccos\left(\frac{trace({R^i}^T\hat{R}^i) - 1}{2}\right) \tag{5}$$

$$\mathbf{TE}^i_X = ||t^i - \hat{t}^i|| \tag{6}$$

It is generally observed that, when registration performs well, these errors are usually limited and predictable; However, they could drift randomly during failures, often leading to more than 90° or $50m$ of error. It is neither interpretable nor repeatable to average the error over all the pairs containing occasional arbitrarily large errors; On the contrary, we often choose to average only those errors of the successful pairs. The registration success is assessed based on the criterion of $S(X, i) = \mathbb{1}(RE^i_X < T_{rot}) \times \mathbb{1}(TE^i_X < T_{trans})$, where $\mathbb{1}(\cdot)$ is the Iverson Bracket, and $T_{rot} = 5°, T_{trans} = 2m$ are two generally accepted thresholds. After that, we could calculate the RRE, RTE as the average of RE and TE of succeeded pairs, and RR as the portion of successful pairs over all pairs, as formulated in Eqs. (7) to (9):

$$\mathbf{RRE}_X = \frac{1}{\sum\limits_{i=1}^{|X|} S(X,i)} \sum_{i=1}^{|X|} \left(S(X,i) \times \mathbf{RE}^i_X\right) \tag{7}$$

$$\mathbf{RTE}_X = \frac{1}{\sum\limits_{i=1}^{|X|} S(X,i)} \sum_{i=1}^{|X|} \left(S(X,i) \times \mathbf{TE}^i_X\right) \tag{8}$$

$$\mathbf{RR}_X = \frac{1}{|X|} \sum_{i=1}^{|X|} S(X,i) \tag{9}$$

Next, mRR is defined as the average of RR over five registration subsets with $||t|| \in [d_1, d_2]$ meters, and the tuple (d_1, d_2) is parameterized according to our specification, *i.e.*, $D_{V2V} = \{(5, 10), (10, 20), (20, 30), (30, 40), (40, 50)\}$, respectively according to Eq. (10):

$$\mathbf{mRR} = \frac{1}{|D_{V2V}|} \sum_{(d_1,d_2)\in D_{V2V}} \mathbf{RR}_{X_{[d_1,d_2]}} \tag{10}$$

Lastly, given a dataset X and the estimated correspondences $(j, k) \in C^i$ denoting that $p^j \in S^i, q^k \in T^i$ are a pair of correspondence, the inlier ratio is defined as Eq. (11):

$$\mathbf{IR}_X = \sum_{i=1}^{|X|} \sum_{(j,k)\in C^i} \frac{\mathbb{1}(||R^i p^j + t^i - q^k|| \leq T_{inlier})}{|X| \times |C^i|} \tag{11}$$

Where $T_{inlier} = 0.3m$ is the inlier distance threshold.

8. Method Details

8.1. Description of SC^2-PCR

We describe the design philosophy and algorithm of SC^2-PCR [7] for better stand-alone completeness. SC^2-PCR consists of two cascading contributions: a spatial compatibility measure, SC^2, and a complete registration pipeline built upon fascinating properties of the SC^2 measure.

Past literature have extensively used first order spatial compatibility to measure correspondence quality, which is defined as $M_{x,y} = \Big| ||p^i_S - p^j_S||_2 - ||p^k_T - p^l_T||_2 \Big|$ for two correspondences $c_x = (p^i_S, p^k_T)$ and $c_y = (p^j_S, p^l_T)$, where

```
# labeler, student  - MinkowskiNet backbones
# lambda            - EMA decay factor
# B, b              - frame interval bound, and batch size
# update_distance() - recalculates all frame pairs with increased B
# spatial_filter()  - match and adaptively filter features
# SC2_PCR()         - original SC2-PCR implementation
# NN_search()       - KD-Tree nearest-neighbor search

for epoch in range(num_epochs):
    # EMA update
    labeler.state_dict = labeler.state_dict * lambda + \
                         student.state_dict * (1-lambda)
    # increase the frame interval B and update dataset
    dataset, B = update_distance(dataset, epoch, num_epochs, B)
    for iter in range(len(loader)):
        inputs = dataset.__getitem__(iter)
        feat0_s, feat1_s = student(inputs) # [b,N1,C], [b,N2,C]
        if B != 1:
            # generate matches with EYOC
            with torch.no_grad():
                feat0_l, feat1_l = labeler(inputs) # [b,N1,C], [b,N2,C]
                initial_match = spatial_filter(feat0_l, feat1_l) # [b,Nx,2]
                trans = SC2_PCR(initial_match) # [b,4,4]
                match = NN_search(inputs[“points”], trans) # [b,5000,2]
        else:
            # use pseudo matches generated using identity pose
            match = NN_search(inputs[“points”], [np.eye(4) for _ in range(b)])
        contrastive_loss(feat0_s, feat1_s, match).backward()
```

Figure 7. Python style pseudo code of the core implementation of EYOC.

$c_x, c_y \in C$ and M is a matrix of size $|C| \times |C|$. The higher the metric is, the more likely both correspondences c_x, c_y are correct. However, there is still a chance that outliers can be compatible with inliers, making them hard to distinguish. In contrast, the SC^2 measure uses $M \cdot M^2$ to measure the number of correspondences in the universe that are simultaneously compatible with two compatible correspondences. As all inliers are compatible with each other, the inliers receive skyrocketing compatibility scores ($\geq \#inliers - 2$) and hence are easily identified from outliers.

Built upon the SC^2 measure, SC^2-PCR takes a two-stage filtering pipeline using the spectral technique to select the most promising seed correspondences and to determine the optimal transformation. The algorithm is both GPU-compatible and non-parametric, resulting in outstanding registration recall, FPS, and generalization capability. All these features entitle SC^2-PCR as an ideal labelling algorithm on unlabelled point cloud data.

8.2. Pseudo Code

We provide a skeletal structure of EYOC in Fig. 7. All components of EYOC are displayed in the figure. EMA update and distance extension of B precede every epoch, effectively preparing proper weights and data for the next epoch. Inside every training step, if the current frame interval is one, then identity pose will be used for supervised training. Otherwise, the labeler, SR and CR will work together to produce fake correspondence labels. Finally, such labels can be used to calculate a contrastive loss.

9. Additional Results

We place the comparison between EYOC and other distant point cloud registration methods, APR and GCL, in Tab. 6. While EYOC lags a little bit from the SOTA work GCL with oracle labels on new data (K→K, N→N), scoring -10.2% and 23.8% less mRR on KITTi and nuScenes respectively, EYOC scores consistently better than APR. Moreover, existing supervised methods deteriorate greatly when placed out-of-distribution (K→W, N→W), where EYOC gets a lot closer to GCL with -9.6% and -2.2% (ϕ →W). When finetuned from the pretrained GCL weights, EYOC achieves even better results with $X\%$ and $Y\%$ gap from GCL instead. We conclude that EYOC, although suffering a

Method	Labelled→ Unlabelled	mRR	[5,10]	[10,20]	[20,30]	[30,40]	[40,50]
APR	K→ K	77.9	99.2	96.8	88.3	67.6	37.8
	K→ W	69.1	97.1	87.4	68.2	53.2	39.8
	N→ N	58.8	99.5	85.6	43.8	45.7	19.2
	N→ W	68.4	95.2	84.5	60.0	56.1	46.3
GCL	K→ K	93.5	99.0	98.8	96.1	91.7	82.0
	K→ W	88.0	100.0	99.0	91.8	79.9	69.1
	N→ N	85.5	99.3	97.7	91.8	77.8	60.7
	N→ W	80.6	99.0	95.2	81.2	67.6	60.2
EYOC	$\phi \to$ K	83.2	99.5	96.6	89.1	78.6	52.3
	$\phi \to$ W	78.4	97.6	91.3	78.2	65.5	59.3
	$\phi \to$ N	61.7	96.7	85.6	61.8	37.5	26.9

Table 6. **Comparison of EYOC, FCGF+APR(a) and GCL+Conv**, where *K, W, N, ϕ* represent KITTI, WOD, nuScenes and scratch. While we observe GCL > EYOC > APR in supervised settings, EYOC excels on new unlabelled data by unsupervised finetuning. This will be included in the revision.

performance gap with the SOTA distant PCR method GCL, boasts top-tier performance on unlabelled new data distributions. Furthermore, the defeat can be potentially negated or even overturned should EYOC uses the same group-wise training scheme as GCL, which counts as our future work.

10. Discussions

Compatibility with previous literature. Moreover, we notice that Hypothesis 4.1 would hint that point cloud features would deteriorate (*i.e.*, move) on the feature space slower than linear functions relative to the distance-to-LiDAR (*e.g.*, radical functions). We argue that this does not contradict previous literature [29] which found the relation to be linear; While previous literature looked into the in-domain performance of converged models, we are looking into the out-of-domain performance of models during training. It is natural for networks to behave differently on seen and unseen data.

Performance Upper Bound. We note that better network weight boosts SC^2-PCR's label quality and better labels promote network performance. Consequently, EYOC's upper bound should be the combination of *(i) bound of SC^2-PCR labels given a hypothetical oracle feature extractor*, and *(ii) bound of a feature extraction network given an oracle labeler algorithm, i.e., supervised training*. Our inclination is that bound *(ii)* is tighter and contributes a major decrease in the upper bound while SC^2-PCR, *i.e.*, bound *(i)*, plays a minor part, as evidenced by the RR@[40m,50m] values consistently remaining below 65%, far from the 90+ RR reported in SC^2-PCR.

Error Accumulation. We believe EYOC is capable of avoiding error accumulation thanks to the induction bias present in the filtering pipeline. Pose estimators such as SC^2-PCR tend to output poses that are either close to perfect (Fig. 6) or randomly distributed in the SO(3) space. While the presence of suboptimal features may decrease the percentage of perfect poses, they do not incur significant errors on all output poses, and the precise poses stay correct. In return, during instances of failure, the random erroneous positives and negatives are scattered in feature space (as anything could be matched with anything else), effectively canceling each other out, yielding limited impact compared to the correct labels.

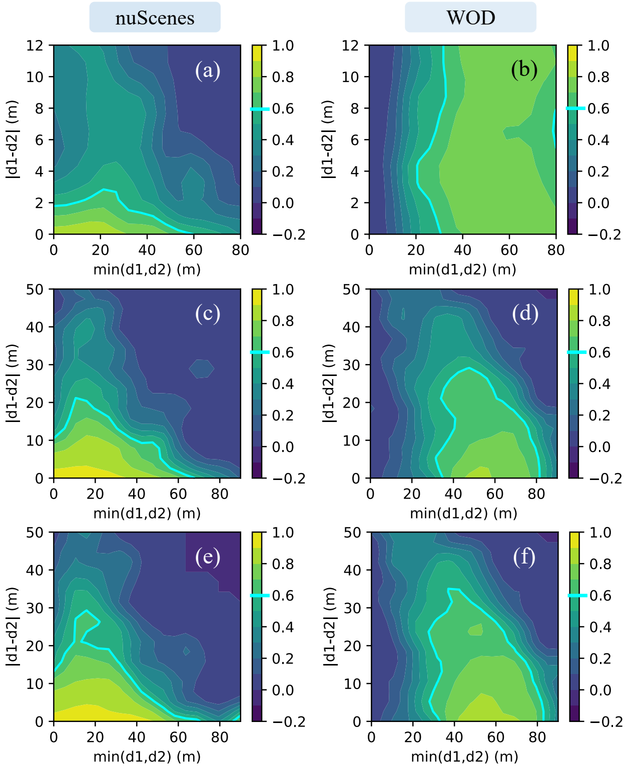

Figure 8. **Visual groundings** for our hypothesis on (a,c,e) nuScenes [6] and (b,d,f) WOD [43]. Cosine similarity of correspondences with its distance to two LiDARs, d_1, d_2, is displayed for $I \in [1, 1]$ (top), $I \in [1, 15]$ (middle), and $I \in [1, 30]$ (bottom). Decision boundaries at $s_{thresh} = 0.6$ are highlighted in cyan.

11. Visualization

11.1. Spatial Filtering on other Datasets

We display the spatial feature similarity results on WOD and nuScenes in Fig. 8, where d_1, d_2 denotes the distance from a correspondence to the two LiDAR centers, and the similarity is indicated by brightness. The decision boundary of $s_thresh = 0.6$ is highlighted in cyan, similar to Fig. 4. WOD exhibits almost identical traits to those on KITTI, showing a drastic feature deterioration in the close-to-LiDAR regions as well as the extremely far regions, and cutting off at 40m would almost always cut the closer half below 0.6 similarity, indicating the similarity between the two filtering strategies. On the other hand, nuScenes displays a similar pattern where high-similarity regions are clustered 20 meters away from the LiDAR. Compared to those on KITTI or WOD, the pinnacle region in nuScenes is slightly shifted towards the LiDAR compared with the other two datasets, due to the lower LiDAR resolution and consequently lower density. In nuScenes, it would be improper to cut off at 40m, although the training does converge and has decent performance as reported in Tab. 1. While this phenomenon is attributed to the discrepancy between KITTI-style datasets and nuScenes-style datasets, we also highlight that EYOC is robust under such discrepancies even when the patterns for the pretraining dataset (WOD) largely differ from the actual one on the finetuning dataset (nuScenes).

11.2. Registration Results

We display the registration results of EYOC on KITTI, nuScenes and WOD in Figs. 9 to 11.

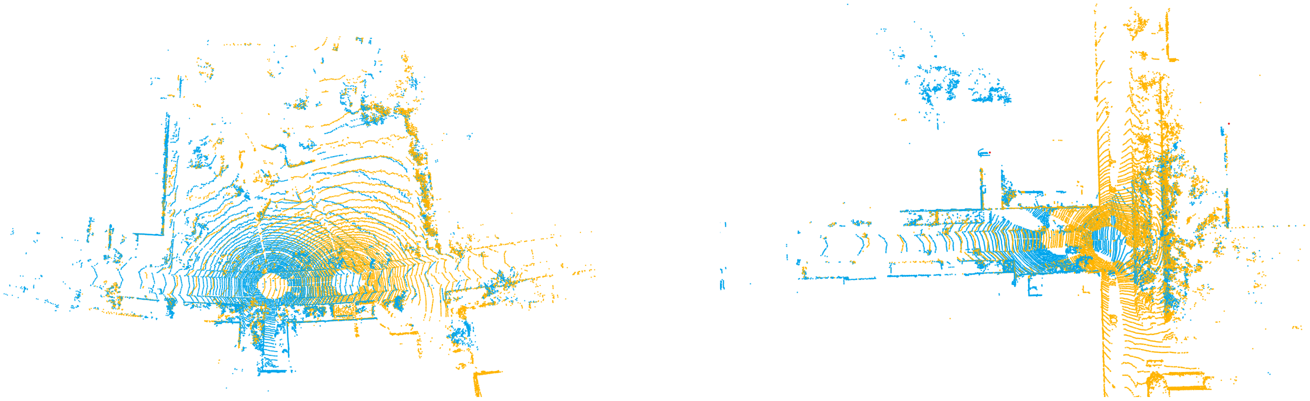

Figure 9. Registration results of EYOC on KITTI [16].

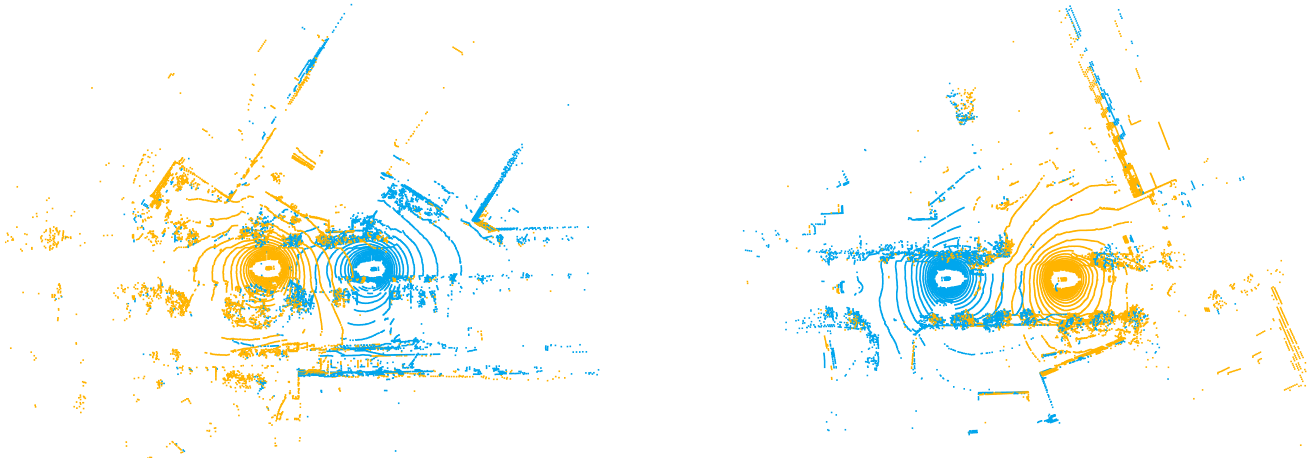

Figure 10. Registration results of EYOC on nuScenes [6].

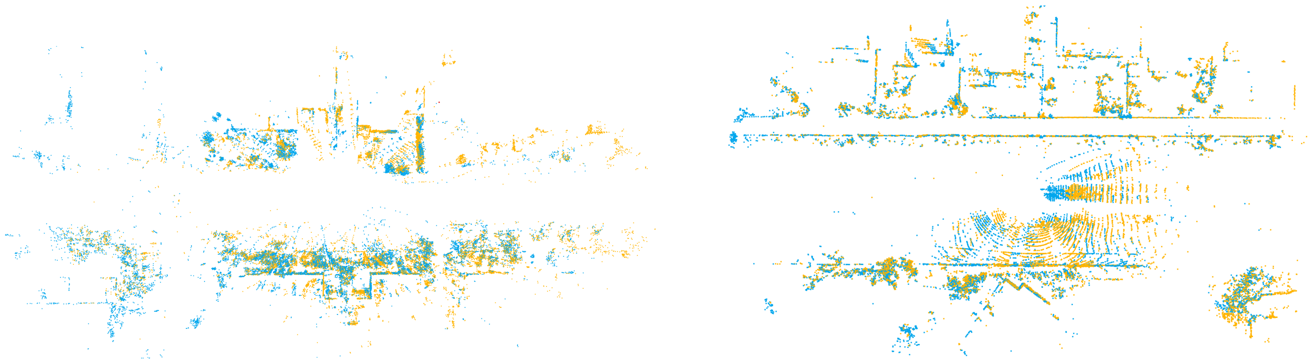

Figure 11. Registration results of EYOC on WOD [43], demonstrated using only the second return.