2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Leveraging Cycle-Consistent Anchor Points for Self-Supervised RGB-D Registration

Siddharth Tourani*§, Jayaram Reddy†, Sarvesh Thakur†,
K Madhava Krishna†, Muhammad Haris Khan§, N Dinesh Reddy‡
* Computer Vision and Learning Lab, University of Heidelberg,
§ MBZUAI, † RRC, IIIT Hyderabad, ‡ Amazon

***Abstract*— With the rise in consumer depth cameras, a wealth of unlabeled RGB-D data has become available. This prompts the question of how to utilize this data for geometric reasoning of scenes. While many RGB-D registration methods rely on geometric and feature-based similarity, we take a different approach. We use cycle-consistent keypoints as salient points to enforce spatial coherence constraints during matching, improving correspondence accuracy. Additionally, we introduce a novel pose block that combines a GRU recurrent unit with transformation synchronization, blending historical and multi-view data. Our approach surpasses previous self-supervised registration methods on ScanNet and 3DMatch, even outperforming some older supervised methods. We also integrate our components into existing methods, showing their effectiveness.**

I. INTRODUCTION

RGB-D cameras are a rich source of information for scene understanding. They are especially useful for robotic tasks like: Simultaneous Localization and Mapping, drone navigation and object pose estimation. The growing use of such cameras has led to a substantial influx of RGB-D data lacking ground-truth pose information.

Typically, pose information for RGB-D data is subsequently derived using SfM pipelines [1], which can introduce noise and encounter optimization challenges, particularly in feature-scarce environments. The noisy pose information one may obtain from such methods can lead to catastrophic failures in downstream robotic tasks.

In this paper we investigate the use of salient portions of a scene to improve RGB-D registration. Existing self-supervised methods mostly leverage either feature similarity or geometric information available from the depth modality to perform this task. In our method, we exploit an under-exploited source of information, the salient portions of a scene. We assume the salient points of a scene are easily recognized in multiple views and leverage spatial relations with these points to constrain the difficult correspondence search process, yielding improved registration performance. We do so by incorporating a spatial coherence cost into the correspondence matching problem.

Assuming these salient points are accurately localized, we exploit the fact that for correct correspondences the relative distance of a 3D point to these salient points should be transformation invariant.

Our proposed method is trained on RGB-D video clips without any ground-truth. An assumption in our method is that the salient points we recognize are cycle-consistent, i.e.they are easily learned via a basic cycle-consistency loss and visible in every view. **We refer to these salient points as anchor points.**

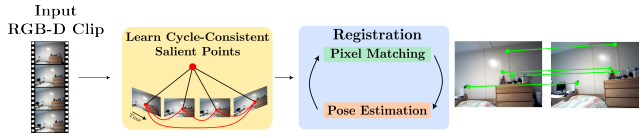


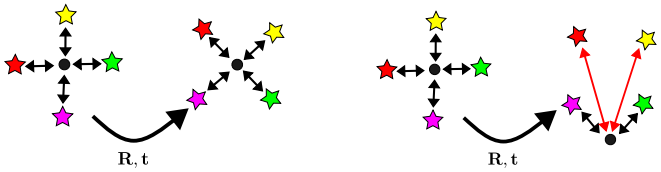
Fig. 1: We propose to initially learn cycle-consistent salient points from RGB-D clips to substantially improve registration accuracy by subsequently constraining the correspondence estimation process.

Our method consists of an initial anchor point matching stage, in we learn anchor points for each video clip during training. We then incorporate these anchor points in the correspondence matching problem via a spatial coherence cost. As our method takes as input multiple RGB-D frames, for pose estimation we combine a GRU pose optimizer similar to the one proposed in [2] with a pose synchronization module. The GRU unit incorporates past information via its hidden state, while pose synchronization leverages pose composition constraints across views yielding accurate pose estimates that are consistent across views.

Our contributions to the correspondence estimation and pose estimation modules lead to substantial improvements on ScanNet [3] and 3DMatch [4] datasets leading to new state-of-the-art for self-supervised RGB-D registration.

To summarize our contributions are as follows:

- We formulate a cycle-consistent keypoint matching module. The keypoints learned via this module impose spatial constraints on the correspondence estimation problem, improving registration.
- We propose a RANSAC-free approach for pose estimation, that combines historic information via a GRU unit with pose compositional consistency across views via transformation synchronization.
- We show through experiments and ablation studies the benefit of our proposed modules. Our method achieves a new state-of-the-art among self-supervised RGB-D registration methods and approaches performance close to strong supervised baselines in terms of correspondence accuracy.



(a) Spatially Coherent (b) Spatially Incoherent

Fig. 2: **Illustration of Spatial Coherence** The uniquely colored stars are anchor points, the black dot a correspondence and arrows indicate distance. (a) shows a spatially coherent correspondence. In the transformed view, the distances between the correspondence and the anchor point are roughly identical. (b) shows a spatially incoherent correspondence, with the red arrows indicating distances that violate spatial coherence.

II. Related Works

We differentiate between point cloud registration and RGB-D registration as they use different input modalities.

a) ***Correspondence Estimation and Registration****:* Classical point cloud registration techniques heavily relied on manually engineered features [5] or assumed perfect correspondence [6], but their effectiveness was bounded by the expressivity of these features. In upgrading point cloud registration methods to the deep learning era, learned counterparts have been proposed for the different components of the registration pipeline. These include learned keypoint descriptors [7]–[9], correspondence estimation [10]–[15]. Amongst the supervised RGB-D registration approaches proposed in the literature are [16]–[21].

While cycle-consistency has been used previously for correspondence estimation [22], [23], these methods impose cycle-consistency on all pixels in a video clip. We instead use cycle-consistency to localize salient points of a scene, which we then use as inputs to our correspondence estimation problem. [24] uses within-frame spatial constraints similar to us, but they do so to prune outliers as opposed to learn anchor points.

b) ***Self-Supervised Registration****:* In addition to the supervised methods mentioned above, there have been self-supervised and unsupervised RGB-D registration methods proposed [25]–[27]. These serve as direct comparisons to our RGB-D registration methods. All of these methods estimate correspondences using a weighted version of Lowe's ratio test [28] for correspondence estimation and the Kabsch algorithm [29] for relative pose estimation between point cloud pairs. These methods are typically supervised by a single weighted *L2-registration loss*. In a different category, self-supervised point cloud registration methods, have been proposed as well [30]–[34].

III. Method

From each video clip, we learn a set of cycle-consistent anchor points $\mathcal{C}_{ij}$ by solving a multi-graph matching problem, where i, j refers to the frame indices. These points are used to enforce spatial constraints to pixel matching problem, which yields a set of soft-correspondences $\mathcal{M}_{ij}^{px}$. The soft correspondences are in-turn used for pose-estimation and refinement. The estimated pose information is fed into pixel matching block as geometric consistency cost. Pixel matching and pose information are iterated over. This we define as the inner iteration. As the estimated anchor points can be noisy, we periodically update the anchor points as well. This we define as the outer iteration. Our method is outlined in Figure 3 and we explain each module in detail below.

Notation $\mathbb{I}$ is the identity matrix. i, j throughout the paper are indices of the frames, while r, s, k, l refer are the matched patches or pixels and t denotes time.

A. Feature Extraction

We use a pre-trained ResNet-18 [35] to extract local coarse (at $1/4$ resolution) and fine-level features (at $1/2$ resolution), we downsample the fine-level features via average pooling to the coarse level resolution and concatenate them along the channel dimension yielding features that combine both high and low-frequency information.

B. Anchor Point Learning / Matching

We utilize the memory efficient matching strategy used in [36] and flatten the features $\mathcal{F}_i$ to 1-D vectors. We construct matching problem by computing the dot product of feature maps between frames, i.e.the score matrix of frames i, j is:

$$\mathcal{S}_{ij} = -\langle \mathcal{F}_i, \mathcal{F}_j \rangle \tag{1}$$

We do this for all pairs of frames. These score matrices are then optimized via *Sinkhorn normalization* [37] yielding soft matchings between pairs of frames. To allow for partial matching arising from different field of view, occlusions and missing depth we add slack row and column vectors to allow for non-assignment, as in [38].

To localize the keypoints in higher resolution, we up-sample the feature map and correspondences. We convert these soft pairwise matches $\mathcal{M}_{ij}$, into cycle consistent matches using the matrix factorization method proposed in [39]. We use these cycle-consistent matches as anchor points in our pixel matching module, to impose addition spatial constraints on the correspondence estimation process. They are denoted by $\mathcal{C}_{ij}$ for correspondences between frames i, j. The anchor point locations are stored as parameters and periodically optimized by minimizing their reprojection error.

C. Pixel Level Matching

Anchor Point Distance Encoding Relative distance encoding has been shown to encode $\mathbb{SE}(3)$ invariant information within a point cloud and effective in supervised registration [13]. Incorporating the same encoding into our method, we found that unlike in [13] it gave only marginal benefit. This difference is probably because the self-attention module in [13] benefits from supervision and access to ground-truth for determining relevant regions within a point cloud. Conversely, our self-supervised approach operates on pseudo ground-truth. Let $\mathbf{p}$ denote an exemplar 3D point.

Hence, our proposal is to capture the $\mathbb{SE}(3)$ invariant information by capitalizing on the stability of our anchor

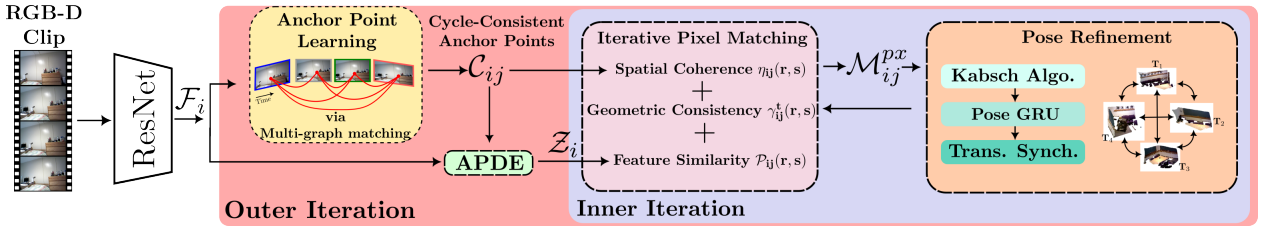


Fig. 3: **Overview of our Method** Features are extracted from an RGB-D clip a via ResNet backbone. They are then used to learn a set of anchor points that are cycle-consistent across frames from the clip. These cycle-consistent anchor points are input into the pixel matching module. The pixel matching module and pose refinement module iterate feeding into each other. This is termed the inner iteration. The outer iteration happens periodically to update the anchor point locations. ADPE stands for Anchor Point Distance Embedding.

points and solely measuring distances relative to these anchor points. We thus define an *anchor point distance embedding* $\mathbf{r_p}$ which we integrate into a self-attention module.

Given a feature matrix $\mathcal{F}_i \in \mathbb{R}^{N\times d}$ as input, the modified attention module outputs a feature matrix $\mathcal{Z}_i \in \mathbb{R}^{N\times d}$, where $\mathbf{z}_p$ is the row vector and is the weighted sum of all projected features for $\mathbf{p}$

$$\mathbf{z_p} = \sum_{q=1}^{N} a_{pq}(\mathbf{x_p}\mathbf{W^V}) \tag{2}$$

where a_{pq} is the normalized attention score computed by a row-wise softmax of e_{pq}.

$$e_{pq} = \frac{(\mathbf{x_p}\mathbf{W^Q}+\mathbf{r_p}\mathbf{W^R})(\mathbf{x_q}\mathbf{W^K}+\mathbf{r_q}\mathbf{W^R})}{\sqrt{d}} \tag{3}$$

Here d is the feature dimension. $\mathbf{W^Q}$, $\mathbf{W^K}$, $\mathbf{W^V}$ and $\mathbf{W^R}$ are projection matrices. The modified elements of the self-attention block are shown in red. We have modified the self attention equation based on the best practices espoused in [40], which explores the different ways to do this.

Let the average distance between $\mathbf{p}$ and the anchor points in its frame beρ_{pq}. $\mathbf{r_p}$ is computed by applying a sinusoidal function [41] on ρ_{pq}/σ_d, where σ_d is a hyper-parameter used to tune the sensitivity to distance variations. We upsampled the feature maps $\mathcal{F}_i$ to $1/2$ half resolution and pass them through the anchor point aware self attention block and get as output feature maps $\mathcal{Z}_i$. These are then input into the pixel matching module.

Spatial Coherence Cost We define spatial consistency as the consistency within frame of a point relative to the other points in the same frame. Intuitively speaking, it encodes transformation invariant information within frame which can be used as a cost to penalize spatially inconsistent correspondences across frames. Incorporating some form of spatial consistency has been shown to improve point cloud registration in [24], [42].

We instead compute a spatial consistency cost function by measuring distance to anchor points, as follows:

$$d_{rs} = \sum_{(k,l)\in\mathcal{C}_{i,j}} \texttt{abs}(\|\mathbf{x}_r - \mathbf{x}_k\| - \|\mathbf{x}_s - \mathbf{x}_l\|) \tag{4}$$

$$\eta_{ij}(r,s) = -\texttt{exp}(\frac{-d^2_{rs}}{\sigma^2_{rs}}) \tag{5}$$

Here, (k,l) are the anchor points in $\mathcal{C}_{ij}$ and abs is the absolute value. $\mathbf{x}_r$ and $\mathbf{x}_s$ are the 3D point corresponding to r, s and σ_{rs} is a learned hyper-parameter.

$\eta_{ij}(r,s)$ tends to 1 when the sum of distances between $\mathbf{x}_r$ and other matched points in frame i is close to sum of distances between $\mathbf{x}_s$ and other matched points in frame j and 0 when the difference is larger. Figure 2 gives an intuitive illustration of the spatial coherence cost.

Iterative Pixel-Level Matching The estimated anchor points allow us to incorporate spatial coherence costs and geometric costs into the matching process, providing additional sources of information to complement feature similarity. The pixel level matching block is used to get more and better localized correspondences .

To keep the pixel level matching tractable, we restrict the matching to $w \times w$ windows centered around each anchor point. These are also regions of higher confidence as they center on anchor points.

Let $\mathcal{Z}_i(r)$, $\mathcal{Z}_i(s)$ be the feature maps of the $w\times w$ windows at points r and s in frames i and j respectively. We initialize the cost matrix between matched points (r,s) of frames (i,j) as follows:

$$\mathcal{P}_{ij}(r,s) = -\langle \mathcal{Z}_i(r), \mathcal{Z}_j(s) \rangle \tag{6}$$

We iterate between the pixel-level matching block and pose update block matching as their outputs feed into each other leading to convergence to a fixed point, which hopefully fits the underlying data.

The pixel-level matching problem at inner iteration t is defined as

$$\mathcal{D}^t_{ij}(r,s) = -\mathcal{P}_{ij}(r,s) - \eta_{ij}(r,s) - \gamma^t_{ij}(r,s) \tag{7}$$

The geometric cost function, $\gamma^t_{ij}(r,s)$ is the Sampson error normalized to unit norm. It takes as input the relative transformations between frames i and j. For exact correspondences and relative pose the value is zero and increase with error. These are initially computed from the anchor points via [29] and subsequently refined by the pose estimation block.

We optimize the score matrices $\mathcal{D}^t_{ij}(r,s)$ via *Sinkhorn normalization*. $\mathcal{M}^{px}_{ij}$ denotes the soft correspondences i.e.for frames (i,j), output by the Sinkhorn algorithm.

D. Iterative Pose Refinement

We leverage both the historical information via a GRU unit and transformation consistency across views via transformation synchronization to improve pose estimation accuracy.

Given the soft correspondences $\mathcal{M}_{ij}^{px}$, transformation update $\delta\mathcal{T} \in \mathbb{SE}(3)$ is obtained by minimizing the following weighted mean-squared error:

$$\delta\mathcal{T}_{ij}^{*} = \arg\min_{\delta\mathcal{T}} \sum_{(r,s)\in\mathcal{M}_{ij}^{px}} w_{rs} \|\mathbf{x}_r - \mathcal{T}_{ij}(\mathbf{x}_s)\|; \tag{8}$$

This is done via a modified Kabsch [29] algorithm, commonly used in differentiable point cloud registration. $w_{rs} = \text{softmax}(\mathcal{D}_{ij}^{t}(r,s))$ weighs the L2-objective.

Inspired by [2], we add a GRU based pose update block to incorporate information from past time steps into the pose estimation process. Our GRU block predicts relative pose updates $\Delta\mathcal{T}_{ij}$ as 6D vectors (using the representation in [43]). We update the transformations via an $\mathbb{SE}(3)$ retraction.

$$\mathcal{T}_{ij}^{t+1} = \mathcal{T}_{ij}^{t} \cdot \text{Exp}(\Delta\mathcal{T}_{ij}) \tag{9}$$

Here Exp(.) is the $\mathbb{SE}(3)$ retraction.

After each update we additionally perform a single transformation synchronization iteration to average out the pose errors across frames. We use the power iteration algorithm used in [27] as its numerically stable. The transformation terms are weighed by the mean of the soft correspondences for each of the frame pairs.

Inner Iteration The inner iteration is used to iterate between the pixel-level matching and the pose update block to refine and increase the number of correspondences. Each inner iteration is performed 20 times.

Outer Iteration The outer iteration iterates over the entire pipeline, i.e.the anchor point learning, pixel matching and pose refinement. It is done 3 times per batch. The outer iteration lets for updating the anchor points and their locations. We do this by minimizing the reprojection error via [44]. Optimizing anchor point locations in-turn improves downstream task accuracy.

E. Supervision

We use the registration loss which minimizes the weighted residual error of the estimated correspondences using the estimated alignment. Given a pair of frames (i,j), we compute it as follows:

$$\mathcal{L}_{reg}^{3D}(i,j) = \sum_{(r,s)\in\mathcal{M}_{ij}^{px}} w_{rs} \|\mathbf{x}_s - \mathcal{T}_{ij}(\mathbf{x}_r)\|_2^2 \tag{10}$$

$w_{rs} = \text{softmax}(\mathcal{D}_{ij}^{t}(r,s))$ is a weighing term.

Cycle Consistency Loss We encourage cycle-consistency by combining orthogonality ($\mathcal{L}_{orth}$) and bijectivity ($\mathcal{L}_{bij}$)

$$\mathcal{L}_{orth}(i,j) = \|\mathcal{S}_{ij}\mathcal{S}_{ij}^{T} - \mathbb{I}\|_F + \|\mathcal{S}_{ij}^{T}\mathcal{S}_{ij} - \mathbb{I}\|_F \tag{11}$$

$$\mathcal{L}_{bij}(i,j) = \|\mathcal{S}_{ij}\mathcal{S}_{ji} - \mathbb{I}\|_F + \|\mathcal{S}_{ji}\mathcal{S} - \mathbb{I}\|_F \tag{12}$$

This formulation of cycle-consistency is simpler than the formulation used in [22] and is applied for anchor point learning only.

F. Test Time

At test time, we do not store or use anchor points. Our correspondence estimation module still consists of the two stages. At the initial anchor point learning stage, the matching problem is the same as at train time (see eq. (1)). Inspired by [45], for every anchor point match (r,s) between frames (i,j), we locate its position $(\hat{r},\hat{s})$ in the fine scale feature maps. We then crop two $w \times w$ local windows of the feature maps $\mathcal{Z}_i(\hat{r})$, $\mathcal{Z}_j(\hat{s})$ centered on $(\hat{r},\hat{s})$. We correlate the center feature of $\mathcal{Z}_i(\hat{r})$ with all the features of $\mathcal{Z}_j(\hat{s})$ and normalize, giving us a probability of matching each pixel in the neighborhood of $\hat{s}$ to the pixel $\hat{r}$. We take the expectation over this heat-map to get the final correspondence. For pose estimation, we use only the Kabsch algorithm (eq. (8) and discard the GRU and transformation synchronization.

IV. Results

We follow the evaluation protocol proposed in UR&R [25].

a) Datasets: We evaluate on 3DMatch [4] and the ScanNet [3] *v2* split. ScanNet provides RGB-D videos of 1513 scenes. 3DMatch provides 72 train sequences.

Method	Train Set	Sup.	Angular Error Accuracy ↑ 5°	Angular Error Accuracy ↑ 10°	Angular Error Error ↓ Mean	Angular Error Error ↓ Med.	Translation Error Accuracy ↑ 5	Translation Error Accuracy ↑ 10	Translation Error Error ↓ Mean	Translation Error Error ↓ Med.
3DMVR	Trained on 3DM	✓	81.1	89.3	9.4	1.8	54.5	76.2	18.4	4.5
DGR	Trained on 3DM	✓	87.7	93.2	6.0	1.2	69.0	83.1	11.7	2.9
Geom.Tr.	Trained on 3DM	✓	***98.3***	***99.6***	***1.1***	***0.6***	***91.8***	***96.5***	***2.4***	***0.8***
UR&R	Trained on 3DM	✗	87.6	93.1	4.3	1.0	69.2	84.0	9.5	2.8
BYOC [25]	Trained on 3DM	✗	66.5	85.2	7.4	3.3	30.7	57.6	16.0	8.2
LLT [46]	Trained on 3DM	✗	93.4	96.5	3.0	0.9	76.9	90.2	6.4	2.4
SyncM [27]	Trained on 3DM	✗	93.4	97.6	2.8	**0.7**	76.6	89.9	7.1	2.6
Ours	Trained on 3DM	✗	**95.6**	**98.1**	**2.4**	**0.7**	**81.5**	**92.3**	**3.7**	**1.9**
UR&R [25]	Trained on SN	✗	92.7	95.8	3.4	0.8	77.2	89.6	7.3	2.3
BYOC [26]	Trained on SN	✗	86.5	95.2	3.8	1.7	56.4	80.6	8.7	4.3
LLT [46]	Trained on SN	✗	95.5	97.6	2.5	0.8	80.4	92.2	5.5	2.2
SyncM [27]	Trained on SN	✗	95.4	97.5	2.4	0.7	81.3	**93.8**	5.4	1.9
Ours	Trained on SN	✗	**97.1**	**98.2**	**1.9**	**0.6**	**85.9**	93.6	**3.9**	**1.8**

TABLE I: **Registration Results On ScanNet** Best results are bold and italicized for supervised methods. Best results and next best are bold and underlined for un/self-supervised methods. The train set can be SN (ScanNet) or 3DM (3D Match). "Sup." indicates whether the method is supervised or not.

b) Training Details: Following [25], we generate sequences consisting of 6 images sampled 20 frames apart as in [25]. Images are reshaped to 256×256px in size. Our model is optimized with the Adam [47] optimizer using a learning rate of 5×10^{-4} and momentum parameters of $(0.9, 0.99)$. Hard matches are obtained by thresholding $\mathcal{M}_{ij}^{px}$ at test time. A few iterations of anchor point learning are done to learn some anchor points before moving to pixel matching and pose estimation.

A. Adding Spatial Coherence To Other Methods

To quantify the impact of spatial coherence, we integrate it into other methods. A non-trivial task. Instead of learning anchor points, we sample correspondences present in all input frames using ground-truth poses. These approximately

50 anchor points, similar in number to those learned by our pipeline, helps us introduce spatial coherence costs into other methods. Due to their ground-truth origin, these anchor points offer higher accuracy than those generated by our self-supervised pipeline, potentially providing greater benefits.

B. Registration Accuracy on ScanNet

We first evaluate our approach on RGB-D registration accuracy. The transformation is represented by a rotation matrix **R** and translation vector **t**. We use the commonly used angular error and translation error (see [25] for formulas) as evaluation metrics for registration accuracy.

Baselines We compare against the following un/self-supervised methods: `BYOC` [26][1] , `UR&R` [25], `LLT` [46] and `SyncMatch` [27]. We also show comparisons against older supervised methods `3DMVR` [16] and `DGR` [10] our method outperforms. Additionally, to put into context our method compared to the supervised state-of-the-art, we show results on the current state-of-the-art `Geom.Tr.` [13].

Table I shows results of registration on ScanNet and 3DMatch. `Ours` outperforms other un/self-supervised methods on both 3DMatch and ScanNet by wide margins in most metrics. We analyze them in detail in the ablation study.

Method	Inputs	3D Corres.			2D Corres.		
		1cm	5cm	10cm	1px	2px	5px
Supervised features with trained matching							
SG	I	8.7	62.4	78.7	2.5	9.0	36.9
SG+SC	I+D	16.8	78.8	87.7	5.9	19.7	58.9
LoFTr	I	16.0	72.2	84.6	5.6	18.5	55.5
LoFTr+SC	I+D	24.5	82.1	92.1	**9.5**	**27.2**	**63.1**
Unsupervised features with heuristic matching							
BYOC	D	13.1	55.1	65.4	4.6	15.3	43.9
UR&R	I+D	24.3	4.5	82.6	6.9	19.5	53.3
LLT	I+D	26.2	75.9	82.1	7.2	22.4	58.7
SyncM	I+D	13.1	55.1	65.4	4.6	15.3	43.9
SyncM+GART	I+D	26.8	76.5	84.4	7.5	23.5	59.7
Unsupervised features with trained matching							
Ours	I+D	**31.2**	**84.3**	**92.7**	9.1	25.3	61.2

TABLE II: **Correspondence Inlier % on ScanNet** For the inputs, I denotes image and D denotes depth. `LofTr` and `SG` are originally image based correspondence methods, so incorporating depth information involves some assumptions. Also, adding `SC` to various methods improves correspondence estimation, quantifying its efficacy. `SyncM+GART` is a SyncM variant that incorporates geometric constraints.

C. Correspondence Estimation

Evaluation Metrics We evaluate the estimated correspondences based on their 2D and 3D errors. We project the estimated correspondences into 3D for valid keypoints with depth using known depth and intrinsics. The ground truth transformations are used to align the keypoints and compute the 3D error and the 2D reprojection error. We extract 500 correspondences for all methods to allow for a meaningful comparison between precision values. [2]

[1]we modify the backbone from ResNet-5 to ResNet-18 to make the visual backbone similar to the other methods.

[2]If a method produces < 500 correspondences we use all of them.

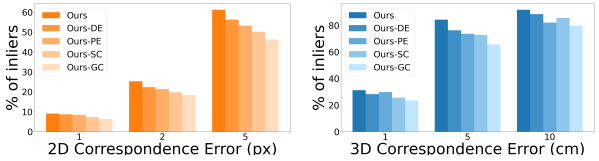
Baselines We compare against supervised and self-supervised counterparts: SuperGlue [38] `SG` attention-based matching algorithm built on top of SuperPoint, `LoFTr` [36] image based feature matching method, `UR&R`, `BYOC`, `SyncM` and `LLT`.

To isolate the impact of our spatial coherence term, we also incorporate it into the matching problems of `SG` and `LoFTr` to assess its effectiveness. This has the added effect of incorporating depth information into `SG` and `LofTr` which are RGB image feature matching methods. These modified methods are represented by `XXX + SC` where `XXX` is the method name. Refer to section IV-A to better understand how these methods make use of anchor points. For `LofTr`, we use Sinkhorn normalization instead of the Dual SoftMax to easily incorporate spatial coherence. We add the spatial coherence costs to the fine-matching stage similar to our method. Table II shows the results of correspondence estimation. While our method is still worse than strong supervised baselines in 2D correspondence, it is still relatively close to them and can be used as a suitable alternative to commonly used self-supervised correspondence estimation algorithms. Additionally, the effect of the spatial coherence cost (SC.) is strongly positive improving the percentage of correspondence inliers. Also note while these methods, assume different modalities (RGB or depth only). We do our best to modify them to use both when adding `SC`.

D. Importance of individual modules

1) Ablation Study: We ablate the following components of our pipeline: anchor point depth encoding (`DE`), spatial coherence costs (`SC`), geometric costs (`GC`), and the pose estimation block (`PE`). We measure the registration error and correspondence accuracy in 2D (*in pixels*) and 3D (*in cms*). Results are averaged over all test sequences in ScanNet.

Figure 4 and table IV shows the results of removing `DE`, `SC` `GC` and `PE` on correspondence error and rotation error respectively. Unsurprisingly `GC` which incorporates pixel specific information contributes the most to our method in correspondence estimation and registration accuracy. However, closely following it is `SC`, demonstrating our spatial coherence costs that make use of anchor points are also quite effective.



(a) 2D Error (in px) (b) 3D Error (in cm)

Fig. 4: **Correspondence Ablation** Ablation Study for Inlier % under various thresholds in 2D and 3D. While swapping or removing all four modules brings a decrease in inlier count, `GC` has the most significant effect followed by `SC`.

2) Adding Modules to Other Methods: We also validate the individual proposed modules by adding them to `SyncM`. We only do so to `SyncM.` as `UR&R` is quite close to it in

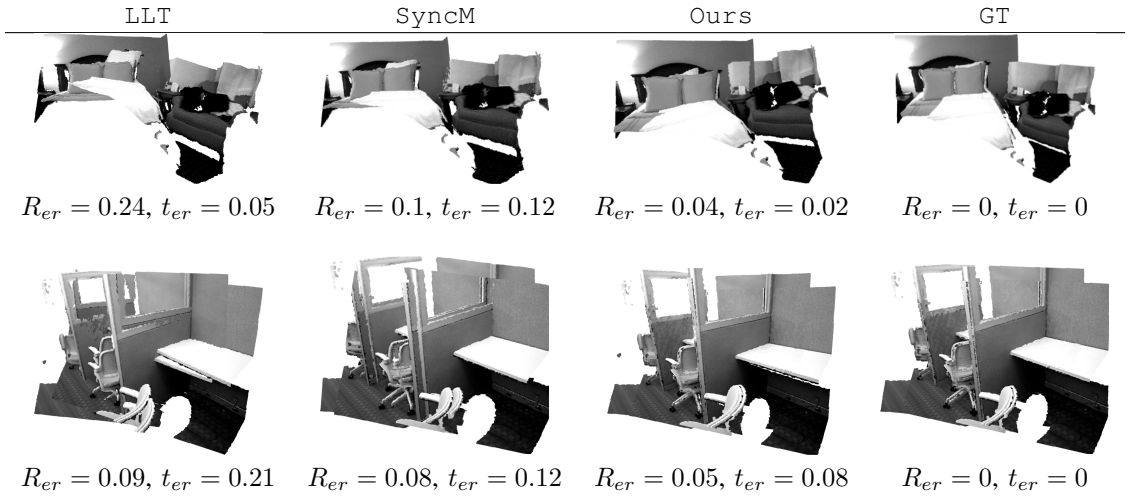
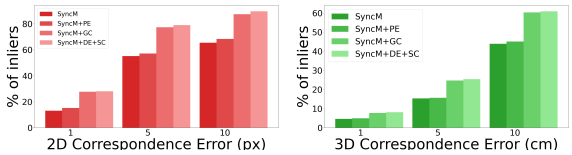


TABLE III: **Qualitative Results on 3DMatch** Each row shows the registration result on two RGB-D scans. Underneath each figure, we show the rotation and translation errors of the scans. We compare LLT, and SyncM to Ours. GT is the ground truth.



(a) 2D Error (in px) (b) 3D Error (in cm)

Fig. 5: **Addition Analysis on Correspondence Error** Adding SC+DE to SyncM has an even more significant impact on inlier % than even GC. All modules do improve the performance of SyncM.

Study	Method	Angular Error Accuracy ↑ 5°	10°	Angular Error ↓ Mean	Med.	Translation Error Accuracy ↑ 5	10	Translation Error ↓ Mean	Med.
Ablation	Ours-DE	96.7	98.1	2.0	0.8	81.6	92.5	6.9	2.1
	Ours-PE	96.4	97.8	2.2	0.7	83.4	91.6	6.7	2.2
	Ours-SC	95.3	97.7	2.4	0.7	82.1	90.7	5.1	2.3
	Ours-GC	94.2	97.9	2.4	0.7	80.4	90.4	5.7	2.4
	Ours	**97.1**	**98.2**	**1.9**	**0.6**	**85.9**	**93.6**	**3.9**	**1.9**
Addition	SyncM	93.4	97.6	2.8	0.7	76.6	89.9	7.1	2.6
	SyncM+PE	92.7	97.8	2.6	0.7	79.6	90.8	5.9	2.5
	SyncM+DE+SC	95.5	98.1	2.0	**0.6**	81.3	92.1	5.4	2.1
	SyncM+GC	95.7	97.9	2.0	**0.6**	81.6	91.8	5.2	2.2

TABLE IV: **Registration Ablation And Addition Analysis** Best results are bold. Removing GC has the greatest impact on registration error, followed by SC. In the addition analysis, adding DE+SC to SyncM has an even stronger impact than GC. The impact of PE is significant but not dominant.

methodology and results should be similar for both. While both LLT and BYOC process depth in a significantly different way making the incorporation of our modules in them difficult.

Even modifying SyncM to use our modules is non-trivial.

We create three modifications of SyncM: SyncM + PE by swapping its pose block with PE, SyncM + GC by adding GC to the matching problem and SyncM + DE + SC where we add DE to the features of SyncM and add SC to the matching problem. Note that the anchor points used in SyncM + DE + SC do not have the noise of our pipeline and are thus not optimized. Thus for SyncM + DE + SC, we only run the inner iteration.

Table IV shows the impact of adding the modules SyncM. As can be seen all modules are beneficial to the algorithm. Unsurprisingly SyncM + DE + SC outperforms even SyncM + GC. This is probably because of perfectly localized anchor points. Figure 5 mirrors a similar story for correspondence error with DE + SC outperforming GC.

We show additional ablation studies in the supplementary.

V. Conclusions

Leveraging easily identifiable salient portions within RGB-D scenes remains an underexplored resource for geometric reasoning. By using these salient points (anchor points), we constrained the correspondence matching problem, improving correspondence localization. Additionally, we introduced technical enhancements to the registration pipeline, effectively leveraging complementary data sources and enhancing registration accuracy. Consequently, our approach sets a new RGB-D registration state-of-the-art for both ScanNet and 3DMatch benchmarks.

References

[1] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[2] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," *Advances in neural information processing systems*, vol. 34, pp. 16 558–16 569, 2021.

[3] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in

Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 5828–5839.

[4] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1802–1811.

[5] F. Pomerleau, F. Colas, and R. Y. Siegwart, "A review of point cloud registration algorithms for mobile robotics," *Found. Trends Robotics*, vol. 4, pp. 1–104, 2015.

[6] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Other Conferences*, 1992.

[7] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 8958–8966.

[8] X. Bai, Z. Luo, L. Zhou, H. Fu, L. Quan, and C.-L. Tai, "D3feat: Joint learning of dense detection and description of 3d local features," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6358–6366, 2020.

[9] H. Deng, T. Birdal, and S. Ilic, "Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 602–618.

[10] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2514–2523.

[11] Z. J. Yew and G. H. Lee, "Rpm-net: Robust point matching using learned features," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11 821–11 830, 2020.

[12] W. Yuan, B. Eckart, K. Kim, V. Jampani, D. Fox, and J. Kautz, "Deepgmr: Learning latent gaussian mixture models for registration," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, Springer, 2020, pp. 733–750.

[13] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, and K. Xu, "Geometric transformer for fast and robust point cloud registration," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11 133–11 142, 2022.

[14] Y. Wang and J. M. Solomon, "Deep closest point: Learning representations for point cloud registration," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3522–3531, 2019.

[15] S. Huang, Z. Gojcic, M. (Usvyatsov, A. Wieser, and K. Schindler, "Predator: Registration of 3d point clouds with low overlap," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4265–4274, 2020.

[16] Z. Gojcic, C. Zhou, J. D. Wegner, L. J. Guibas, and T. Birdal, "Learning multiview 3d point cloud registration," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1759–1769.

[17] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, "The perfect match: 3d point cloud matching with smoothed densities," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5545–5554.

[18] A. Hertz, R. Hanocka, R. Giryes, and D. Cohen-Or, "Pointgmm: A neural gmm network for point clouds," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12 051–12 060, 2020.

[19] H. M. Le, T.-T. Do, T. Hoang, and N.-M. Cheung, "Sdrsac: Semidefinite-based randomized approach for robust point cloud registration without correspondences," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 124–133.

[20] W. Lu, G. Wan, Y. Zhou, X. Fu, P. Yuan, and S. Song, "Deepvcp: An end-to-end deep neural network for point cloud registration," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 12–21.

[21] X. Huang, G. Mei, and J. Zhang, "Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 366–11 374.

[22] A. Jabri, A. Owens, and A. Efros, "Space-time correspondence as a contrastive random walk," *Advances in neural information processing systems*, vol. 33, pp. 19 545–19 560, 2020.

[23] Z. Bian, A. Jabri, A. A. Efros, and A. Owens, "Learning pixel trajectories with multiscale contrastive random walks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6508–6519.

[24] X. Bai, Z. Luo, L. Zhou, *et al.*, "PointDSC: Robust Point Cloud Registration using Deep Spatial Consistency," *CVPR*, 2021.

[25] M. El Banani, L. Gao, and J. Johnson, "Unsupervised randr: Unsupervised pointcloud registration via differentiable rendering," in *CVPR*, 2021.

[26] M. El Banani and J. Johnson, "Bootstrap Your Own Correspondences," in *ICCV*, 2021.

[27] M. El Banani, I. Rocco, D. Novotny, *et al.*, "Self-supervised Correspondence Estimation via Multiview Registration," 2023.

[28] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[29] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A*, vol. 32, pp. 922–923, 1976.

[30] Y. Shen, L. Hui, H. Jiang, J. Xie, and J. Yang, "Reliable inlier evaluation for unsupervised point cloud

registration," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 2198–2206.

[31] G. Mei, H. Tang, X. Huang, *et al.*, "Unsupervised deep probabilistic approach for partial point cloud registration," *ArXiv*, vol. abs/2303.13290, 2023.

[32] X. Huang, S.-j. Li, Y. Zuo, Y. Fang, J. Zhang, and X. Zhao, "Unsupervised point cloud registration by learning unified gaussian mixture models," *IEEE Robotics and Automation Letters*, vol. 7, pp. 7028–7035, 2022.

[33] G. Mei, H. Tang, X. Huang, *et al.*, "Unsupervised deep probabilistic approach for partial point cloud registration," *arXiv preprint arXiv:2303.13290*, 2023.

[34] P. Kadam, M. Zhang, S. Liu, and C.-C. J. Kuo, "Unsupervised point cloud registration via salient points analysis (spa)," in *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, IEEE, 2020, pp. 5–8.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[36] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "Loftr: Detector-free local feature matching with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8922–8931.

[37] R. Sinkhorn and P. Knopp, "Concerning nonnegative matrices and doubly stochastic matrices," *Pacific Journal of Mathematics*, vol. 21, no. 2, pp. 343–348, 1967.

[38] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

[39] F. Bernard, J. Thunberg, J. Goncalves, and C. Theobalt, "Synchronisation of partial multi-matchings via non-negative factorisations," *Pattern Recognition*, vol. 92, pp. 146–155, 2019.

[40] K. Wu, H. Peng, M. Chen, J. Fu, and H. Chao, "Rethinking and improving relative position encoding for vision transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 033–10 041.

[41] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[42] Y. Zhang, J. Yu, X. Huang, W. Zhou, and J. Hou, "Pcr-cg: Point cloud registration via deep explicit color and geometry," in *European Conference on Computer Vision*, Springer, 2022, pp. 443–459.

[43] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5745–5753.

[44] L. Pineda, T. Fan, M. Monge, *et al.*, "Theseus: A library for differentiable nonlinear optimization," *Advances in Neural Information Processing Systems*, vol. 35, pp. 3801–3818, 2022.

[45] Q. Wang, X. Zhou, B. Hariharan, and N. Snavely, "Learning feature descriptors using camera pose supervision," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, Springer, 2020, pp. 757–774.

[46] Z. Wang, X. Huo, Z. Chen, J. Zhang, L. Sheng, and D. Xu, "Improving rgb-d point cloud registration by learning multi-scale local linear transformation," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, Springer, 2022, pp. 175–191.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015*, Y. Bengio and Y. LeCun, Eds., 2015.